IPP HURRAY!
www.hurray.isep.ipp.pt

Technical Report

Collision-Free Beacon Scheduling Mechanisms for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks

Anis Koubaa **Mário Alves**
Mellek Attia **Anneleen Van Nieuwenhuyse**

TR-061104
Version: 1.0
Date: Nov 2006

Collision-Free Beacon Scheduling Mechanisms for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks

Anis KOUBAA, Mario ALVES, Melek ATTIA, Anneleen VAN NIEUWENHUYSE

IPP-HURRAY!

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8340509

E-mail: {akoubaa@dei., mjf@}isep.ipp.pt

http://www.hurray.isep.ipp.pt

Abstract

The recently standardized IEEE 802.15.4/Zigbee protocol stack offers great potentials for ubiquitous and pervasive computing, namely for Wireless Sensor Networks (WSNs). However, there are still some open and ambiguous issues that turn its practical use a challenging task. One of those issues is how to build a synchronized multi-hop cluster-tree network, which is quite suitable for QoS support in WSNs. In fact, the current IEEE 802.15.4/Zigbee specifications restrict the synchronization in the beaconenabled mode (by the generation of periodic beacon frames) to star-based networks, while it supports multi-hop networking using the peer-to-peer mesh topology, but with no synchronization. Even though both specifications mention the possible use of cluster-tree topologies, which combine multi-hop and synchronization features, the description on how to effectively construct such a network topology is missing. This paper tackles this problem, unveils the ambiguities regarding the use of the cluster-tree topology and proposes two collision-free beacon frame scheduling schemes. We strongly believe that the results provided in this paper trigger a significant step towards the practical and efficient use of IEEE 802.15.4/Zigbee cluster-tree networks.

Collision-Free Beacon Scheduling Mechanisms for IEEE 802.15.4/Zigbee Cluster-Tree Wireless Sensor Networks

Anis KOUBAA[1], Mario ALVES[1], Melek ATTIA[1,2], Anneleen VAN NIEUWENHUYSE[1,3]

[1] IPP-HURRAY! Research Group, Polytechnic Institute of Porto,
Rua Antonio Bernardino de Almeida, 431, 4200-003 Porto, Portugal
[2] Sup'Com Tunis, Higher School of Telecommunication in Tunis
Route Raoued 2083 Cité El-Ghazala, Tunisia
[3] KaHo St Lieven, Electronics department, Ghent, Belgium

akoubaa@dei.isep.ipp.pt, mjf@isep.ipp.pt

Abstract. The recently standardized IEEE 802.15.4/Zigbee protocol stack offers great potentials for ubiquitous and pervasive computing, namely for Wireless Sensor Networks (WSNs). However, there are still some open and ambiguous issues that turn its practical use a challenging task. One of those issues is how to build a synchronized multi-hop cluster-tree network, which is quite suitable for QoS support in WSNs. In fact, the current IEEE 802.15.4/Zigbee specifications restrict the synchronization in the beacon-enabled mode (by the generation of periodic beacon frames) to star-based networks, while it supports multi-hop networking using the peer-to-peer mesh topology, but with no synchronization. Even though both specifications mention the possible use of cluster-tree topologies, which combine multi-hop and synchronization features, the description on how to effectively construct such a network topology is missing. This paper tackles this problem, unveils the ambiguities regarding the use of the cluster-tree topology and proposes two collision-free beacon frame scheduling schemes. We strongly believe that the results provided in this paper trigger a significant step towards the practical and efficient use of IEEE 802.15.4/Zigbee cluster-tree networks.

1. Introduction

The joint efforts of the IEEE 802.15.4 task group [1] and the Zigbee Alliance [2] have ended up with the specification of a standard protocol stack for Low-Rate Wireless Personal Area Networks (LR-WPANs), a promising technology for Wireless Sensor Networks (WSNs) [3-7]. In what follows, we denote by *Zigbee* the entire IEEE 802.15.4/Zigbee protocol stack.

Zigbee is gaining an exponentially increasing interest from industry and is considered as a universal solution for low-cost low-power wirelessly connected monitoring and control devices [5-7]. This interest is mainly driven by the potentially large number of emerging applications including home automation (as the current principal commercial target of the Zigbee Alliance), health care monitoring, industrial automation, environmental monitoring, surveillance, and so on. These applications have essentially been triggered by the wireless sensor network paradigm, which represents the new generation of network infrastructure for large-scale distributed embedded systems.

The reputation of Zigbee, even though not already widely commercially available, is closely related to the objectives for which it was designed [1, 2] and to its flexibility to fit different network and application requirements. While it was designed for low-cost wireless devices (such as wireless sensors), the most important technical features of Zigbee are to provide *low power consumption* and *real-time guarantees*. However, the benefit gained from these features typically depends on the configuration of the Medium Access Control (MAC) sub-layer, whether operating in beacon-enabled (with synchronization) or in non beacon-enabled (without synchronization) modes. In fact, in the beacon-enabled mode it is possible to achieve very low duty cycles (from 100% down to 0.006%), which is particularly interesting for WSN applications where energy constraint and network lifetime are main concerns. In addition, the beacon-enabled mode also offers real-time guarantees by means of the Guaranteed Time Slot (GTS) mechanism, an attractive feature for time-sensitive WSN applications. On the other side, the non beacon-enabled mode does not provide any of those features, but it has the advantage of lower complexity and more scalability as compared to the beacon-enabled mode, since the former does not require any synchronization. At a first glance, the non beacon-enabled mode may be an interesting solution for large-scale WSNs. Note that in the context of Zigbee, *synchronization* means that a central device called the *PAN Coordinator* (also referred to as *Zigbee Coordinator* - ZC) periodically transmits beacon frames to its neighbor nodes, which are then broadcast throughout the entire network via *Coordinator* nodes (or *ZigBee Routers* – ZRs). Summarizing; WSN applications with particular energy or/and delay requirements must be configured to operate in the synchronous beacon-enabled mode [8].

However, the beacon-enabled mode suffers from lacking scalability since, inherently to its operational behavior, it is limited to star-based networks. In fact, in a star-based network operating in beacon-enabled mode, beacon frames are periodically transmitted by a central node, for synchronizing the nodes in its vicinity. As a consequence, the network coverage is limited to the transmission range of the PAN Coordinator, which restricts the number of nodes in the network. This is particularly unsuitable for WSNs, which are commonly accepted to be large-scale and ad-hoc. Therefore, there is a paradox between supporting scalability at the cost of energy consumption and delay guarantees, and vice-versa. It would be more appropriate if both features (synchronization and scalability) could be simultaneously supported into the same network.

In that direction, the Zigbee standard also specifies the concept of cluster-tree topology. A cluster-tree network is formed by several coordinators (ZigBee Routers) that periodically send beacon frames to the nodes of their cluster, thus providing them synchronization services. From what we have understood from the standard specification and based on some interactions with some members of the Zigbee Alliance, the cluster-tree model (proposed in Section 5.2.1.2 in Reference [1]) is merely a suggestion from Motorola. In no other place in the IEEE 802.15.4/Zigbee standards there is clear description on how the cluster-tree model can be implemented. The available information regarding this topology gives a broad (rather confusing) overview on how the cluster-tree network should operate and some details on the tree routing algorithm that was proposed by Motorola [2]. However, the interaction between the MAC sub-layer and the routing layer that builds the cluster-tree network, such that the synchronization is maintained all over the network, is missing.

More specifically, the cluster-tree model proposes that the network contains more than one *coordinator* (also referred to as *Zigbee Router*), which generate periodic beacon frames to synchronize nodes in their neighborhood (cluster). In this case, if these periodic beacon frames are sent in a non organized fashion (with no particular schedule), they will collide either with each other or with data frames. It results that enabling the beacon mode in a cluster-tree Zigbee network is a challenging problem. In fact, in case of beacon frame collisions, nodes that wait the periodic beacon frames will loose synchronization with their coordinators, and consequently with the network, which will prevent them to communicate. As a consequence, beacon frame scheduling mechanisms must be defined to avoid beacon frame collisions in ZigBee cluster-tree networks. The problem that we tackle in this paper can be roughly formulated as follows:

Synchronization in a ZigBee cluster-tree network: Given an IEEE 802.15.4/Zigbee network with several coordinators generating periodic beacon frames and organized in a cluster-tree topology, how to schedule the generation time offsets of beacon frames issued from different coordinators to completely avoid beacon frame collisions with each other and with data frames.

The purpose of this paper is to overcome this problem by proposing collision-free beacon frame scheduling algorithms. To our best knowledge, the beacon frame scheduling problem has not been explicitly resolved by the Zigbee standardization groups or by previous research works. Only some basic approaches dealing with this problem were proposed for discussion by the Task Group 15.4b [9], which is a group aiming to improve some inconsistencies of the original specification. However, no algorithms for providing collision-free beacon frame generation have been proposed so far.

2. Related Work

Clustering and synchronization are common problems in WSNs that have been addressed in some research works (e.g. [10-12]). In Reference [10], the authors proposed LEACH, a clustering-based protocol using a randomized rotation and selection of cluster-heads to optimize energy consumption. After the random selection of cluster-heads, the other nodes decide to which cluster they belong, and inform the corresponding cluster-head (using CSMA/CA) of their decision. After the reception of all join requests, cluster-heads compute a TDMA (Time Division Multiple Access) schedule according to the number of nodes in their cluster. This schedule is broadcast back to the node in the cluster. Inter-cluster interference is mitigated using different CDMA (Code Division Multiple Access) codes in each cluster. This clustering and synchronization approach differs from the Zigbee approach in three aspects, which turns our problem quite different. First, concerning clustering in Zigbee networks, coordinators (or cluster-heads) are fixed (do not change during run-time). Second, the synchronization is not made using a TDMA schedule, but by means of periodic beacon frame transmissions, which has the advantage of higher flexibility (TDMA is not scalable and is vulnerable to dynamic network changes). Finally, Zigbee does not allow the use of CDMA to avoid inter-cluster interferences, which leads to having collisions between beacon and data frames issued in different clusters. In our case, a node that experiences a collision of a beacon frame will inevitably loose synchronization. Hence, there is a need to schedule different beacon frames from different coordinators to avoid beacon frame losses that lead to undesirable synchronization problems.

Being aware of this problem, the Task Group 15.4b [9] has been working on an improved version of the IEEE 802.15.4 standard and proposed for discussion some basic approaches for avoiding beacon frame collisions that may be adopted in the upcoming extension of the standard. A first approach, called the *Beacon-Only Period* approach, consists in having a time window in the beginning of each superframe reserved for beacon frame transmissions. The second approach, based on *time division*, proposed that beacon frames of a given cluster are sent during the inactivity periods of the other clusters. However, the algorithms showing how to schedule beacon frame transmission in a collision-free fashion are not presented. More specifically, the approaches proposed by the Task Group 15.4b show how to extend the standard to take beacon frame scheduling into account, but how to choose the time offsets of different beacon is not addressed, which triggered the motivation for this work. Surprisingly, the approaches discussed in the Task Group 15.4b were not (fully included) in the new version of the standard IEEE 802.15.4b [13] that will be free of charge by September 2006.

The main contributions of this paper are three-folded.

- First, we present and analyze the state-of-the art of the beacon frame collision problem (Section 4), and the different approaches proposed in Reference [9] to avoid it (Section 5).
- Second, we propose beacon frame scheduling mechanisms for both approaches proposed in Reference [9] (Section 6).
- Third, we present some implementation guidelines (Section 7).

3. Overview of the IEEE 802.15.4/Zigbee protocols

The IEEE 802.15.4 MAC protocol supports two operational modes that may be selected by the PAN Coordinator: (1) the *non beacon-enabled mode*, in which the MAC is simply ruled by non-slotted CSMA/CA, (2) the *beacon-enabled mode*, in which beacons are periodically sent by the PAN Coordinator to synchronize nodes that are associated with it, and to identify the PAN. In this paper, we focus on the beacon-enabled mode and analyze its deployment in cluster-tree networks.

In beacon-enabled mode, beacon frames are periodically sent by the *PAN Coordinator*, to identify its WPAN and synchronize nodes that are associated with it. Doing so, a superframe structure (see Fig. 1) is defined by (1) the *Beacon Interval* (*BI*), which defines the time between two consecutive beacon frames, (2) the *Superframe Duration* (*SD*), which defines the active portion in the *BI*, and is divided into 16 equally-sized time slots, during which frame transmissions are allowed. Optionally, an inactive period is defined if $BI > SD$. During the inactive period (if it exists), all nodes may enter in a sleep mode to save energy.

BI and *SD* are determined by two parameters, the *Beacon Order* (*BO*) and the *Superframe Order* (*SO*), respectively, as follows:

$$\left.\begin{aligned} BI &= aBaseSuperframeDuration \cdot 2^{BO} \\ SD &= aBaseSuperframeDuration \cdot 2^{SO} \end{aligned}\right\} for \;\; 0 \le SO \le BO \le 14 \qquad \textbf{(1)}$$

aBaseSuperframeDuration = 15.36 ms (assuming 250 kbps in the 2.4 GHz frequency band) denotes the minimum duration of the superframe, corresponding to $SO = 0$.

During the *SD*, nodes compete for medium access using slotted CSMA/CA in the *Contention Access Period* (CAP). For time-sensitive applications, IEEE 802.15.4 enables the definition of a *Contention-Free Period* (CFP) within the *SD*, by the allocation of *Guaranteed Time Slots* (GTS).

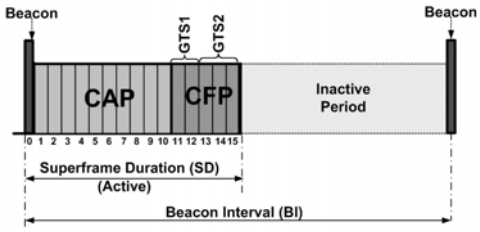


Fig. 1. Beacon Interval and Superframe Duration as defined by the IEEE 802.15.4 standard [1]

It can be easily observed in Fig. 1 that low duty cycles can be configured by setting small values of the superframe order (*SO*) as compared to beacon order (*BO*), resulting in greater sleep (inactive) periods.

The advantage of this synchronization with periodic beacon frame transmissions from the PAN Coordinator is that all nodes wake up and enter sleep mode at the same time. However, as discussed earlier, using this synchronization scheme in a cluster-tree network with multiple coordinators sending beacon frames, each with its own beacon interval, is a challenging problem due to beacon frame collisions.

4. Beacon Collision Problem in Cluster-Tree Zigbee WPANs

4.1 Network model

The beacon frame collision problem in cluster-tree Zigbee WPANs has been addressed as *Request for Comments* in the Task Group 15.4b [9]. In this section, we analyze the different types of beacon frame collision conflicts identified by the Task Group 15.4b in Reference [9].

In this paper, we consider the cluster-tree network model as presented in Fig. 2. The whole network is identified by the PAN Coordinator (or *Zigbee Coordinator – ZC*), which is unique. The PAN Coordinator may allow other special nodes, called *Zigbee Routers* (ZR) or *coordinators*, to send periodic beacon frames to synchronize the nodes in their vicinity. Throughout this paper, we interchangeably use *Zigbee Router* and *Coordinator* and both are denoted by *ZR*. Hence, each coordinator ZR_i acts as a cluster-head of the cluster *i* for all its child nodes (that are associated to the network through it), and as a consequence, will send periodic beacon frames to keep them synchronized. The cluster-tree is formed by several parent-to-child associations between Zigbee Routers until a certain depth. In Fig. 2, for instance, ZR2 is a parent coordinator of ZR5 and a child coordinator of the PAN Coordinator, considered as the root of the tree.

It is easy to notice that sending periodic beacon frames without special care on timing issues may result in beacon frame collisions in some nodes that are in the range of more than one coordinator. The Task Group 15.4b has identified two types of collisions: (1) direct beacon frame collisions and, (2) indirect beacon frame collisions, which are briefly explained next.

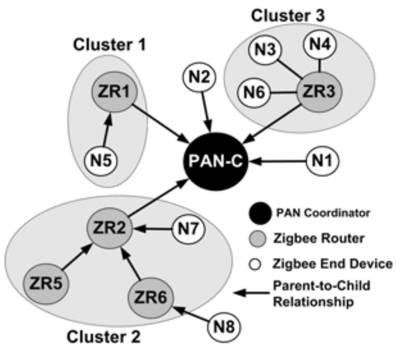
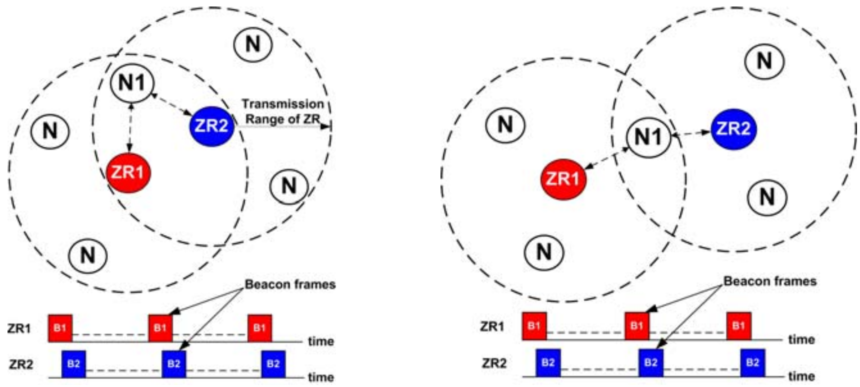


Fig. 2. The cluster-tree topology model

4.2 Direct beacon frame collisions

Direct beacon frame collisions occur when two ore more coordinators are in the transmission range of each other (*direct neighbors* or parent-to-child relation) and send their beacon frames at approximately the same time, as shown in Fig. 3.a. In that figure, assuming that node N1 is a child of ZR1, which sends its beacon frame at approximately the same time as ZR2, node N1 loses its synchronization with its parent ZR1 due to the beacon frame collisions.



a. Direct beacon frame collision ***b.*** Indirect beacon frame collision

Fig. 3. The beacon frame collision problem

4.3 Indirect beacon frame collisions

Indirect beacon frame collisions occur when two ore more coordinators cannot hear each other, but have overlapped transmission ranges (*indirect neighbors*) and send their beacon frames at approximately the same time, as shown in Fig. 3.b. In that figure, node N1, which is located in the overlapped region of the transmission ranges of ZR1 and ZR2, will not be able to correctly receive their beacon frames since they will be collided.

Note that a collision between data and beacon frames may happen when a coordinator sends its periodic beacon frame during the active period of an adjacent cluster. Hence, this problem must also be overcome.

5. Basic approaches of the Task Group 15.4b for Beacon Frame Collision Avoidance

Since no mechanism to avoid beacon frame collisions was considered in the current IEEE 802.15.4 standard, some proposals has been discussed in the Task Group 15.4b. These approaches were proposed as pattern ideas to trigger the design of the solutions of the beacon frame collision problem. In what follows, we outline these proposals.

5.1 Direct beacon frame collision avoidance

Two approaches were proposed to avoid direct beacon frame collision problem (Fig. 4).

5.1.1 The time division approach

In this approach, time is divided such that beacon frames and the superframe duration of a given coordinator are sent during the inactive period of its neighbor coordinators, as shown in Fig. 4.a. The idea is that each coordinator selects the starting time `Beacon_Tx_Offset` to transmit its beacon frame. The starting time must be different from the starting times of its neighbor coordinators and their parents. This approach requires that a coordinator wakes up both in its active period and in its parent's active period.

The limitations of this approach are: (1) it imposes low duty cycles; (2) direct communication between sibling coordinators (coordinators with the same parent) is not possible, since each cluster operates in a time window different from its adjacent clusters.

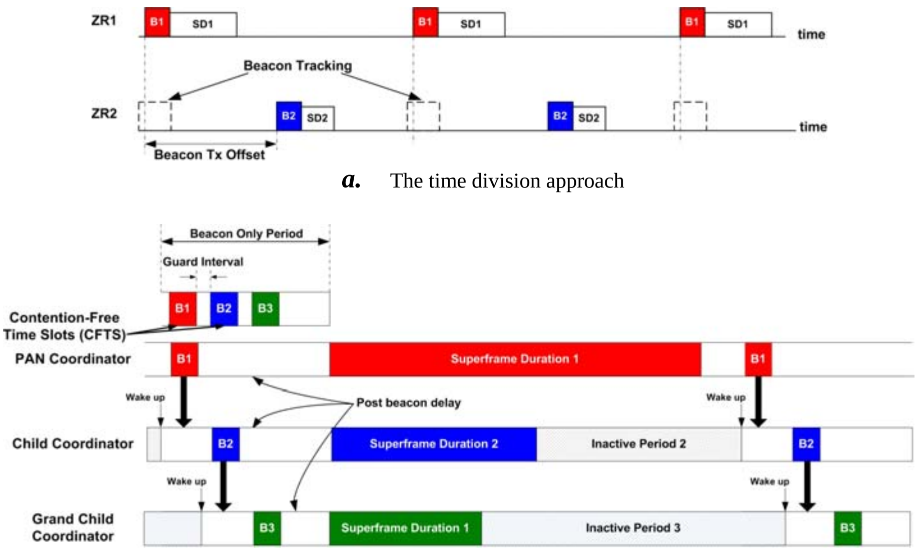
The density of devices that can be supported is inversely proportional to the ratio of the beacon order and superframe orders assuming that all *BO*s and *SO*s are equal for all clusters. This approach has been supported by the Zigbee specification [2].

Observe that `Beacon_Tx_Offset` must be chosen adequately, not only to avoid beacon frame collisions, but also to enable efficient utilization of inactive periods, thus maximizing the number of clusters in the same network. This problem is more challenging when the superframe orders and beacon orders are different from one cluster to another. This issue is addressed in Section 6.

5.1.2 The beacon-only period approach

In this approach, a time window, denoted as *Beacon-Only Period*, is considered at the beginning of each superframe for the transmission of beacon frames in a contention-free fashion (Fig. 4.b). Each coordinator chooses a sending time offset by selecting a *contention-free time slot* (CFTS) such that its beacon frame does not collide with beacon frames sent by its neighbors. The advantage of this approach as compared to the previous one is that the active periods of the different clusters start at the same time, thus direct communication between neighbor nodes is possible, and there is no constraint on the duty cycle.

The main complexity of this approach is the dimensioning of the duration of the beacon-only period for a given network topology. This duration depends on the number of nodes in the network, their parent-child relationship and also the scheduling mechanism used to allocate the CFTS to each coordinator. This issue is analyzed in Section 6.



a. The time division approach

b. The Beacon-Only Period approach

Fig. 4. Beacon frame collision avoidance approaches

5.2 Indirect beacon frame collision avoidance

The problem of indirect beacon frame collisions is more complex than the one of direct beacon frame collisions. There is a need to not only know the neighbor coordinators, but also all other coordinators that are two-hops away. Two alternatives were proposed by the Task Group 15.4b.

The reactive approach. In this approach, a coordinator does not carry any specific procedure to avoid indirect beacon frame collision during the association with its parent. Once a beacon frame conflict is detected by a given node, it initiates a recovery procedure to resolve the problem, which may take a long time. The interested reader can refer to [9] for more details, which will not be presented in this paper since they are out of scope.

The proactive approach. In this approach, coordinators try to avoid the indirect beacon frame conflict at the association phase by the collection of specific data about beacon frame transmission times of their neighbors. In this approach, each potential coordinator must have the ability to forward the beacon frame time offset of its parent to its neighbor coordinators. This approach is more complex than the reactive approach, but it completely avoids beacon frame collisions during network run-time.

5.3 discussions

We have presented the two approaches proposed by Task Group 15.4b to avoid direct and indirect beacon frame collisions. Note that these approaches do not include the algorithms to schedule beacon frames transmission. For the time division approach, the organization of the different superframe durations must be evaluated with care to maximize the number of clusters in the network. For the other approach, the beacon-only period must be efficiently dimensioned. To do so, in the next section we propose beacon frame scheduling mechanisms that solve the beacon frame collision problem and take into account the aforementioned requirements.

6. Beacon Frame Scheduling Mechanisms for the Time Division Approach

6.1 Problem formulation

Let us consider an IEEE 802.15.4/Zigbee network as presented in Fig. 2 with a set of N coordinators $\{ZR_i = (SD_i, BI_i)\}_{1 \le i \le N}$ that generate periodic beacon frames with a given superframe order SO_i and beacon order BO_i. SD_i and BI_i denote the superframe duration and the beacon interval of the i^{th} coordinator ZR_i, respectively. The problem is how to organize the beacon frames of the different coordinators to avoid collisions with other beacon and data frames, using the time division approach. The most intuitive idea is to organize beacon frame transmissions in a serial way such that no beacon frame will collide with another even if coordinators are in direct or indirect neighborhood (refer to Section 4). In addition, to avoid collisions with data frames, a beacon frame must not be sent during the superframe duration of another coordinator. Thus, the *beacon frame scheduling problem* comes back to a *superframe scheduling problem*, since each superframe duration starts with a beacon frame.

At a first glance, this problem can be considered as a non-preemptive scheduling of a set of periodic tasks, where the execution time of a task is equal to the superframe duration, and the period is equal to the beacon interval. However, the additional restriction in the superframe scheduling problem is that consecutive instances of SD must be separated by exactly one beacon interval BI.

In what follows, we tackle the superframe scheduling problem for two cases. First, we consider the case of equal superframe durations (beacon orders may be different). Second, we extend the results for the general case with different superframe durations.

6.2 Superframe Duration Scheduling (SDS) algorithm for the time division approach

In case of equal superframe durations, the superframe scheduling problem is somewhat similar to the *pinwheel scheduling problem* presented in [14]. The pinwheel problem consists in finding for a set of positive integer $A = (a_1, \ldots, a_n)$ a cyclic schedule of indices $j \in (1,2, . . . n)$ such that there is at least one index j within any interval of a_j slots. By analogy to our problem, given a set of beacon intervals $A = (BI_1, \ldots, BI_N)$, the problem is to find a cyclic schedule of superframe durations such that there is at least one SD_i in each BI_i. In addition to the pinwheel problem, the distance between two consecutive instances of SD_i must be equal to BI_i. In this paper, we propose a general result for the scheduling problem for different and equal superframe durations.

T1. Let $\mathbb{M} = \left\{A \mid A = \{a_1,\ldots,a_N\} \text{ where } i < j \Rightarrow a_i \text{ divides } a_j \text{ and } \sum_{i=1}^{N} 1/a_i \le 1\right\}$
For an instance $A \in \mathbb{M}$, if a cyclic schedule exists, then the least common multiple of all integers, $LCM(a_1, a_2 \ldots, a_n) = \max_{1\le i\le N}(a_i)$, is the minimum cycle length.

Proof.

The proof is made by contradiction. Assume that a cyclic schedule exists for an instance $A \in \mathbb{M}$ of the pinwheel problem. Since $\forall\, i < j \Rightarrow a_i$ divide a_j, the $\forall\, i < j$ it exists an integer k_{ij} such that $a_j = k_{ij} \cdot a_i$ (harmonic integers). Then, we have $LCM(a_1, a_2 \ldots, a_n) = \max_{1\le i\le N}(a_i)$.

Assume that the minimum cycle length is different from $LCM(a_1, a_2 \ldots, a_n)$. Then, since $LCM(a_1, a_2 \ldots, a_n)$ is not a cycle length, it exists a time slot n that contains a_i such that the $\left(n + LCM(a_1, a_2 \ldots, a_n)\right)^{th}$ time slot does not contains a_i. Since $LCM(a_1, a_2 \ldots, a_n)$ is a multiple of a_i, it directly implies that the set is not schedulable, which is absurd.

■

According to theorem **T1**, the superframe duration scheduling decision problem is PSPACE (by analogy to the pinwheel problem, which is also shown to be PSPACE). Thus, we propose the *Superframe Duration* Scheduling (SDS) algorithm which performs the schedulablity analysis of a set of superframes with different durations and beacon intervals, and provides a schedule if the set is schedulable. The algorithm also holds for equal superframe durations.

Let us consider a set of N coordinators $\{ZR_i = (SD_i, BI_i)\}_{1\le i\le N}$ with different superframe durations.

First, for being schedulable, it is necessary to satisfy that the sum of the duty cycles is lower than 1, which gives the following **necessary condition**.

$$\sum_{i=1}^{N} DC_i = \sum_{i=1}^{N} \frac{SD}{BI_i} \le 1 \qquad \textbf{(2)}$$

Based on theorem **T1**, it is sufficient to analyze the schedulablity of the superframe durations in a hyper-period equal to $\overline{BI}_{maj} = LCM\left(2^{BO_1}, 2^{BO_2} \ldots, 2^{BI_n}\right) = \max_{1\le i\le N}\left(2^{BO_i}\right)$. This hyper period is referred to as *major cycle.*

The idea of the SDS algorithm is the following.

- **Step 1.** We denote this set as $A = \{2^{BO_i}\}_{1\le i\le N}$. Let $\overline{BI}_{\min} = 2^{BO_{\min}}$ be the minimum beacon interval. Hereafter, we call it the *minor cycle*.
- **Step 2.** Organize the set $A = \{\overline{BI}_i = 2^{BO_i}\}_{1\le i\le N}$ in an *increasing* order such that if it exists i, j where $\overline{BI}_i = \overline{BI}_j$, then put $\overline{BI}_i, \overline{BI}_j$ in the set A in the decreasing order of their superframe durations. Hence, if $\overline{SD}_i \ge \overline{SD}_j$, then put $\overline{BI}_i$ before $\overline{BI}_j$ in the set A.
- **Step 3.** define a slotted time line of a length equal to the major cycle $\overline{BI}_{maj}$ and where the size of each slot is equal to the minimum superframe duration SD (time unit corresponding to $SO = 0$).

- **Step 4.** For each element i in A, schedule the superframe duration SD_i by searching the first available time slot in the slotted timeline, and write the index i in SD_i consecutive time slots.
- **Step 5.** Repeat "write the index i in SD_i consecutive time slots after each BI_i interval", until reaching the end of the major cycle.
- **Step 6.** Return "*not schedulable*" if a given superframe duration cannot find periodic free time slots in the major cycle.

To illustrate the SDS algorithm, let us consider the following example presented in Table 2.

Table 1. Example of PAN configuration

Coordinator	SD	BI
C1	4	16
C2	1	8
C3	2	16
C4	1	32
C5	4	32
C6	2	16

The SDS algorithm applied to the example in Table 2 is presented in Fig. 5.

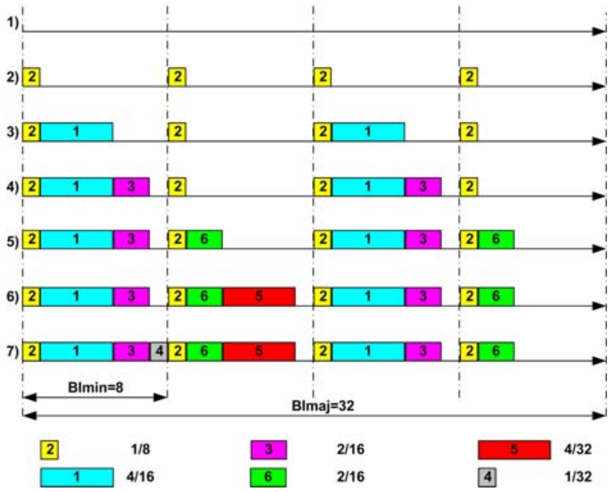


Fig. 5. Illustrative example of the SDS algorithm.

Observe that in this superframe duration set, the major cycle corresponds to $\overline{BI}_{maj} = 32$ and the minor cycle corresponds to $\overline{BI}_{\min} = 8$. Based on Step 2, the set of coordinators is arranged as follows (C2, C1, C3, C6, C4, C5) corresponding the set $A = \{8,16,16,16,32,32\}$. According to Step 3, we consider the slotted timeline of length 32 time slots (major cycle), where each time slot corresponds to a base superframe duration (i.e. SO = 0). Based on Step 4, for each element in A, we place the first instance of the superframe duration of the corresponding coordinator in the first available time slots such that the superframe duration can fit without overlapping with other superframe durations. For instance, the first instance of the superframe duration of Coordinator C2 is placed in the first time slot, and the subsequent instances are placed at a distance equal to a multiple of 8 time slots from the first instance. Then, the first instance of the superframe duration of C1 is placed just after the first superframe duration of C2 (time slot 2). The subsequent instances of C1 are placed at a distance equal to a multiple of 16 time slots from the first instance, and so on. Observe in line (7) of Fig. 6 that this set of coordinators is schedulable since all superframe durations are periodic and not overlapping in the major cycle.

6.3 Superframe duration scheduling with coordinator grouping

In this section, we extend the time division approach to optimize the superframe scheduling algorithm in large-scale networks. Observe that coordinators that are far enough such that their transmission ranges do not overlap can transmit their beacon frames simultaneously without facing the direct and indirect beacon frame collision problems.

To give an intuitive illustration of the approach, we propose the following example. The geographical distribution of this network is presented in Fig. 6 and the network parameters are presented in Table 2 (*SD* is considered as the time unit).

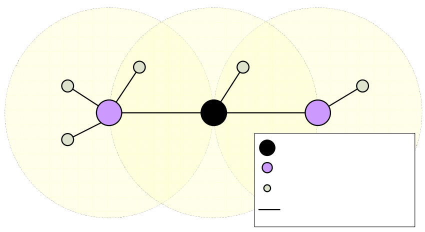


Fig. 6. The geographic distribution of the nodes in the network

Table 2: Example of PAN configuration

Coordinator	SO	BO	SD	BI
C0	0	1	1	2
C1	0	1	1	2
C2	0	1	1	2

Fig. 6 shows that beacon frames from C0, C1 and C2 could collide since C0 has overlapped transmission ranges with both C1 and C2. According to Eq. (2), note that it is not possible to schedule the Superframe of the three coordinators because the total Duty Cycle is greater than 1 (0.5+.05+0.5=1.5>1).

However, observe that coordinators C1 and C2 could send their beacon frames at the same time, since they are neither in direct nor in indirect neighborhood (no overlapping transmission ranges). Thus, it possible that C0 sends its beacon frame followed by coordinators C1 and C2, which may send their beacon frames simultaneously. In this case, no beacon frame collision will occur, and thus the coordinator set becomes schedulable, as presented in Fig. 7.

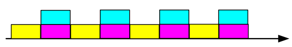


Fig. 7. Superframe duration scheduling with coordinator grouping

In what follows, we propose a general method to group nodes that can send their beacon frames simultaneously.

Let us consider that each coordinator has a circular transmission range of radius r. Two coordinators are not overlapping means that they are geographically separated by a distance at least equal to $2{\cdot}r$, thus can be allowed to send beacon frames at the same time. Hence, the problem can be considered as the vertex coloring problem of graph theory [15] where vertices are the coordinators and an edge is a link between two coordinators that are at least $2{\cdot}r$ away. The vertex coloring algorithm can be implemented in the PAN coordinator, which is assumed to know all coordinators that enter the network, performs group assignments and sends back the grouping result. After processing the vertex coloring algorithm, each coordinator with the same color belongs to the same group and all coordinators belonging to a group can send beacon frames simultaneously.

This grouping strategy has the advantage to find a schedule for a set of coordinators whose total duty cycle is greater than one (as presented in the previous example).

6.4 Implementation issues

From a practical point of view, the superframe duration scheduling algorithm (without coordinator grouping) can be easily implemented in the IEEE 802.15.4 with minor changes. When a new coordinator joins the network, it sends its superframe structure specification (*BO* and *SO*) hop-by-hop to the PAN coordinator. The PAN coordinator then analyzes the schedulability for the whole coordinator set, including the newly joined coordinator, by running the SDS algorithm presented in Section 6.2. If the SDS algorithm generates a valid schedule, the new coordinator is admitted to send beacon frames and the new schedule sequence is sent back to all nodes in the beacon frames. Then, each coordinator updates its new offset and sends its beacon frame with respect to this offset. Note that the offset of a given coordinator can be determined with reference to its parent's beacon transmission time. On the other hand, if the SDS algorithm returns a non schedulable result, the new coordinator will not be admitted to send beacon frames.

The implementation of SDS algorithm with coordinator grouping is more complex to implement and will not be addressed in this paper, for the sake of simplicity.

7. Beacon Frame Scheduling Mechanisms for the Beacon-Only Period Approach

7.1 Problem formulation

Let us consider an IEEE 802.15.4/Zigbee WPANs as presented in Fig. 2 with a set of N coordinators $\{ZR_i = (SD_i, BI_i)\}_{1 \le i \le N}$ that generate periodic beacon frames with a given superframe order SO_i and beacon order BO_i. SD_i and BI_i denote the superframe duration and the beacon interval of the i^{th} coordinator ZR_i, respectively. We assume that the superframe structure starts with a beacon-only period, as presented in Section 5.1.2.

The problem is how to perform the dimensioning of the beacon-only period and how to schedule beacon frame transmission in the beacon-only period.

7.2 The allocation of a contention-free time slot

In Reference [9], the 15.4b task group has suggested the following rules that a coordinator should satisfy to choose a Contention Free-Time Slot (CFTS).

Rule (1) The CFTS of a coordinator *Ci* must be different from the CFTS of its parent.

Rule (2) The CFTS of a coordinator *Ci* must be different from the CFTSs of the parent of its neighbors.

As far as we have understood in Reference [9], the word “neighbors” designates “neighbor coordinators”, excluding simple node neighbors. Hence, both rules can be summarized in one rule as follows:

Rule (12) **The CFTS used by one coordinator is unavailable to its children and their neighbor coordinators.**

Observe that Rules (1) and (2) do not completely eliminate the beacon frame collision problem since simple nodes do not have any means to inform the other coordinators on their parent’s CFTS. As a result, it possible that a beacon frame collision occurs at a simple node when its parent is using the same CFTS as another coordinator.

To avoid this problem, we modify Rule (2) to consider not only coordinator neighbors but also simple node neighbors, which gives:

Rule (2’) The CFTS of a coordinator *Ci* must be different from the CFTSs of the parent of all its neighbors (coordinators and simple nodes).

To completely avoid direct beacon collisions in non parent-to-child relationship situations, we propose the following new rule:

Rule (3) The CFTS of a coordinator *Ci* must be different from the CFTSs of its neighbor coordinators.

Observe that Rule (3) covers some cases that cannot be detected with Rule (2’), as explained in the example scenario presented in Fig. 8. In fact, imagine that node N31 in Fig. 8 does not exist; based on Rule (2’), coordinators C3 and C4 can use the same CFTS. Now, if node N31 joins the network while C3 and C4 already used the same CFTS, it will not be able to be associated neither to coordinator C3 nor to coordinator C4 due to direct beacon collisions. Hence, Rule (3) imposes that direct neighbor coordinators can never use the same CFTS, to avoid this situation.

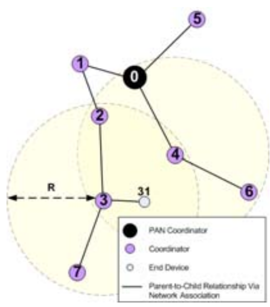


Fig. 8. Illustration of avoiding direct beacon collision with Rule (3)

Another important problem, which was not addressed in Reference [9], is hierarchical synchronization. In fact, note that Rules (1) and (2) allow two coordinators at different depths to have the same CFTS. To illustrate the problem, consider the example of CFTS allocation in Fig. 9, corresponding to the network in Fig. 8.

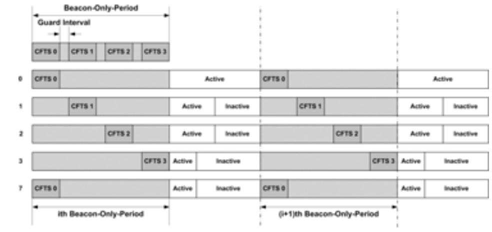


Fig. 9. Illustration of the hierarchical synchronization problem

Observe that coordinator C0 and C7 allocate the same CFTS (CFTS0) and C3, the parent of C7, allocates CFTS3. In this case, imagine that coordinator C3 fails for any reason and stop sending beacon frames, then according to this CFTS allocation scheme coordinator C7 sends its beacon frame before C3, thus children of C7 will be synchronized (while C7 is has not already be synchronized). Since C3 does not send its beacon frames due to a failure, then C7 will not be synchronized and will enter in an orphan state, while its children have already been synchronized.

From this scenario, it is important to impose that coordinators at different depths cannot share the same CFTS. In other words, the CFTS of a parent coordinator must be allocated before the CFTS of all subsequent children in the beacon-only-period. A coordinator should not send its beacon frame only after receiving the beacon frame of its parent to have a homogenously synchronized WPAN.

7.3 The dimensioning of the beacon-only period

The length of the beacon-only period must be evaluated dynamically (by the PAN coordinator) upon the join/leave operations of coordinators during the network run-time. A basic approach is to allocate a new CFTS for each coordinator joining the network. However, this approach is not efficient since the number of CFTS will linearly grows with the number of coordinators, which will significantly increase the duration beacon-only period.

A better alternative consists in using the same CFTS by coordinators that are at the same depth, but sufficiently far away to avoid direct and indirect beacon frame collisions. Hence, if the coordinators are location-aware, the PAN coordinator may assign the same CFTS for coordinators at the same depth and separated by a minimum threshold distance. The efficient dimensioning of the beacon-only period will be subject to a future work.

8. Concluding Remarks

This paper improves on the state-of-the art with the proposal of collision-free beacon frame scheduling mechanisms for IEEE 802.15.4/Zigbee cluster-tree networks. We have analyzed the problem of beacon frame collisions in IEEE 802.15.4/Zigbee WPANs, including direct and indirect beacon frame collisions, and presented the “draft” solutions proposed by the 15.4b task group and their limitations. The main contribution of this paper deals with the proposal of two collision-free beacon frame scheduling mechanisms. We proposed the superframe duration scheduling algorithm, which efficiently organizes the superframe durations of different coordinators in a non overlapping manner, based on their superframe orders and beacon orders. We have shown that this approach may be improved by using coordinator grouping, but induces increasing implementation complexity. The second proposal deals with the specification of the CFTS allocation mechanism in the beacon-only period approach and its dimensioning.

This work represents an important step in understanding the complexity of the deployment of cluster-tree topologies in IEEE 802.15.4/Zigbee WPANs and paves the way for their real deployment. We are currently working on the real implementation of these proposals in our experimental test bed, which already implements the basic features of the IEEE 802.15.4/Zigbee protocol stack build on top of MICAz motes [16] from Crossbow.

References

[1] IEEE-TG15.4, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE standard for Information Technology*, 2003.

[2] Zigbee-Alliance, "ZigBee specification," *http://www.zigbee.org/*, 2005.

[3] J. Zheng and J. L. Myung, "Will IEEE 802.15.4 Make Ubiquitous Networking a Reality?--A Discussion on a Potential Low Power, Low Bit Rate Standard," *IEEE Communications Magazine*, vol. 42, No. 6, pp. 140-146, , 2004.

[4] D. Geer, "Users Make a Beeline for ZigBee Technology," *IEEE Computer Society Press*, vol. 38, Issue 12, pp. 16-19, Dec., 2005.

[5] J. Adams, "Building low power into wireless sensor networks using ZigBee technology," *Industrial Embedded Systems Resource Guide, Networking: Technology*, pp. 26-30, 2005.

[6] T. Culter, "Deploying ZigBee in existing industrial automation networks," *Industrial Embedded System Resource Guide, Networking: Technology*, pp. 34-36, 2005.

[7] C. Herzog, "Creating value with ZigBee Networks," *Industrial Embedded System Resource Guide, Networking: Technology*, pp. 31-33, 2005.

[8] A. Koubâa, M. Alves, and E. Tovar, "GTS Allocation Analysis in IEEE 802.15.4 for Real-Time Wireless Sensor Networks," in *14th International Workshop on Parallel and Distributed Real-Time Systems (WPDRTS 2006)*. Rhodes Island (Greece): IEEE, 2006.

[9] T. G. 15.4b: http://grouper.ieee.org/groups/802/15/pub/TG4b.html.

[10] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocols for wireless microsensor networks," in Proceedings of the Hawaii International Conference on Systems Sciences, Hawai, 2000.

[11] G. Gupta and M. Younis, "Fault-Tolerant Clustering of Wireless Sensor Networks," in Proceedings of the IEEE Wireless Communication and Networks Conference (WCNC 2003), New Orleans (Louisiana), 2003.

[12] V. A. Kottapalli, A. S. Kiremidjiana, J. P. Lyncha, E. Carryerb, T. W. Kennyb, K. H. Law, and Y. Lei, "Two-Tiered Wireless Sensor Network Architecture for Structural Monitoring," in Proceedings of the 10th Annual International Symposium on Smart Structures and Materials, San Diego (USA), 2003.

[13] IEEE802.15.4b, "Draft Revision for IEEE Standard for Information technology— Telecommunications and information exchange between systems— Local and metropolitan area networks— Specific requirements— Part 15.4b: Wireless Medium Access (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)," *To be publically available in September 2006*, 2006.

[14] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, and D. Varvel, "The pinwheel: A Real-Time Scheduling Problem," in Proceedings of the 22nd Hawaii International Conference on System Science, Hawaii, 1989.

[15] R. Diestel, *Graph Theory*, Third Edition ed. New York: Springer, 2005.

[16] Crossbow, "MICAz datasheet," http://www.xbow.com, 2004.