



HAL
open science

TBoPS: a Tree based distributed Beacon only Period Scheduling mechanism for IEEE 802.15.4

Bilel Nefzi, Dawood Khan, Ye-Qiong Song

► **To cite this version:**

Bilel Nefzi, Dawood Khan, Ye-Qiong Song. TBoPS: a Tree based distributed Beacon only Period Scheduling mechanism for IEEE 802.15.4. IEEE workshop WiSARN, in conjunction with IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS2012), IEEE, May 2012, Hangzhou, China. pp.341-346. hal-00745229

HAL Id: hal-00745229

<https://inria.hal.science/hal-00745229>

Submitted on 25 Oct 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TBoPS: a Tree based distributed Beacon only Period Scheduling mechanism for IEEE 802.15.4

Bilel Nefzi

LORIA - University of Lorraine
Campus Scientifique - BP 239 - 54506
Vandoeuvre-Les-Nancy Cedex, France
Email: Bilel.Nefzi@loria.fr

Dawood Khan

INRIA - 615 rue du jardin botanique
54600 Villers-Les-Nancy, France
Email: dawood.khan@inria.fr

Ye-Qiong Song

LORIA - University of Lorraine
Campus Scientifique - BP 239 - 54506
Vandoeuvre-Les-Nancy Cedex, France
Email: Ye-Qiong.Song@loria.fr

Abstract—IEEE 802.15.4 standard specifies a beacon-enabled mode which provides a synchronization environment using beacon transmissions. However, this mode is designed for single hop networks and its use in multi-hop networks is not straightforward. The main challenges of using beacon-enabled mode in multi-hop networks are how to efficiently schedule beacon transmissions to avoid direct and indirect beacon collisions and how to make a schedule tolerant to the clock drifts due to the low cost components. In this paper, we present TBoPS, a novel technique for scheduling beacons in the cluster tree topology. TBoPS uses a dedicated period called beacon only period (BOP) to schedule beacons at the beginning of IEEE 802.15.4 superframe. The advantages of TBoPS is that every beacon-enabled node selects a beacon schedule distributively during association. We analysed the robustness of TBoPS to clock drifts. We also show through simulations that all nodes in the network are synchronized and follow the same superframe structure.

I. INTRODUCTION

Cyber-physical systems (CPS) may monitor and interact with the physical world thanks to the recent advances on wireless sensor and actuator networks (WSAN). When interacting with the real world, real-time properties must be ensured. Providing common time awareness across these WSAN is critical in CPS since the intelligent entity is led to make decisions based on coherent temporal information. Let us take the example of a simple multi-hop WSAN-based CPS for intrusion detection and tracking¹. The detection of an intruder, possibly by multiple sensors, will be reported to the sink, which will trigger the execution of the intruder position estimation (e.g. based on kalman estimation), and then sending of a command to guide mobile robots (mobile sensors) to cover the estimated possible target (intruder) position, which is often an ellipse area whose size depends on the round trip time (RTT) between the intruder detection and the command reception by the mobile sensors. It is obvious that long RTT results in larger area to cover, so needs more mobile sensors to move and to cover the estimated region, and thus consumes more energy. Moreover, the coordinated moving of the mobile sensors needs a common sense of time among them. Otherwise it is impossible to correlate data and make good decision.

¹The intrusion detection and tracking is the main scenario that we are dealing with within the Quasimodo project: <http://quasimodo.loria.fr/>

IEEE 802.15.4 [1] standard is widely used in Wireless Sensor Networks (WSNs) because it meets many of the requirements of these networks. The MAC layer provides two access modes; the beacon-enabled mode and the non beacon-enabled mode. The beacon-enabled mode provides a synchronization environment using beacon transmissions. Therefore, the MAC sublayer uses a superframe structure, defined as the time between two successive beacon transmissions. The superframe can be divided into an active period, during which nodes can transmit data, and a sleep period, during which nodes go to sleep in order to preserve energy. This mode works perfectly in the case of star networks. However, for large scale multi-hop networks, many beacon-enabled nodes (FFD) should transmit beacons in order to synchronize the network.

The main challenge is how to schedule beacon transmissions of these nodes to avoid collisions. In addition, only a subset of beacon enabled nodes should transmit beacon to synchronize the entire network. The existing approaches suppose that these nodes are already known and well distributed in the network. This hardens the deployment process. Our approach efficiently builds the cluster tree to optimize the synchronization process. An FFD becomes a beacon broadcaster only if it has children. If the association rules minimize the number of cluster-heads (*i.e.* FFDs which have children), the beacon scheduling process is eased.

In this paper, we present a Tree based distributed Beacon only Period Scheduling mechanism (TBoPS) for IEEE 802.15.4 multi-hop wireless sensor networks. TBoPS adopts a beacon only approach in which a period of time, called BOP, is reserved for beacon transmissions at the beginning of each superframe. This BOP is followed by the working period (WP) which is used for data transmission. TBoPS procedure is activated for a particular associated FFD if another node tries to associate to it. The advantage of the approach in dense networks is that it minimizes the number of beacon broadcasters while maintaining the superframe structure for all nodes in the network.

The rest of paper is organized as follows. In the next section, we briefly present the beacon-enabled mode IEEE 802.15.4. Then we describe the related work in section III. TBoPS details are given in section IV. Section V gives the clock-drift analysis which derives the maximum superframe duration. Section VI

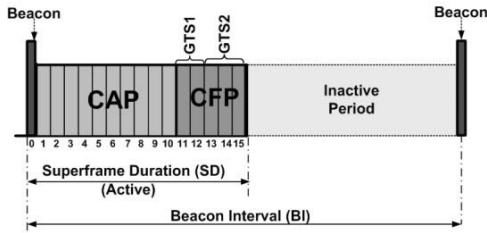


Fig. 1: Superframe

analyzes the cluster-tree formation performance. Section VII concludes the paper.

II. IEEE 802.15.4

In beacon-enabled mode, beacon frames are periodically sent by a central device, referred to as PAN coordinator, to identify its PAN and synchronize nodes that are associated with it. The PAN coordinator defines a superframe structure characterized by a Beacon Interval (BI) specifying the time between two consecutive beacons, and a Superframe Duration (SD) corresponding to the active period, defined as:

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

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

$$\forall 0 \leq SO \leq BO \leq 14 \quad (3)$$

BO and SO are called Beacon Order and Superframe Order, respectively. The Beacon Interval may optionally include an inactive period (for $SO < BO$), in which all nodes may enter into a sleep mode, thus saving energy.

By default, nodes compete for medium access using slotted CSMA/CA during the Contention Access Period (CAP). The IEEE 802.15.4 protocol also provides a Contention-Free Period (CFP) within the superframe, in which a node may request the PAN coordinator to allocate Guaranteed Time Slots (GTS). In this paper, we consider the physical layer operating in the 2.4 GHz frequency band and with a 250 kbps data rate.

It can be easily observed in (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 coordinator is that all nodes wake up and enter sleep mode at the same time. However, with multiple coordinators sending beacon frames, each with its own beacon interval, sending beacon is a challenging problem due to frame collisions. Frame collisions may be direct, which occur when two or more neighbour coordinators transmit their beacons at the same time. They may be indirect, which occur when two or more two-hop neighbour coordinators transmit their beacons at the same time. In the latter case, common neighbour nodes between these coordinators are affected.

III. RELATED WORK

Two approaches are used for beacon scheduling in IEEE 802.15.4; the Time Division (TD) approach and the Beacon Only Period (BOP) approach.

In the TD approach, the beacons of a coordinator are always scheduled in the sleep period of its neighbor coordinators and two-hop coordinators (the ones which send beacons). This approach is the one suggested by IEEE 802.15.4 standard because it does not modify the structure of the superframe. This solution has two major drawbacks. The first one is that it imposes the use of a very low duty cycle since the sleep period have to be long enough to include the active periods of all coordinators. The second drawback is that coordinators are not able to communicate directly with each other. A consequence is that broadcasting is not possible any more. Koubaa *et al.* [2] proposed a centralized beacon scheduling mechanism which uses the TD approach. MeshMAC [5] implements a distributed mechanism for the TD approach. In MeshMAC the node needs first to be associated as an end device. Then it finds a suitable schedule by collecting neighbours and neighbours' neighbours beacon schedule.

In the BOP approach, a portion of the time, called BOP, is reserved for beacon scheduling transmission at the beginning of the superframe. During this period, a coordinator picks a Contention-Free Time Slot and transmits its beacon. We selected the BOP approach for two reasons. First, it ensures that the active periods of all nodes in the network are synchronized thus easing message broadcasting and ensuring fast message delivery (if generated during the active period). Second, it allows coordinators to use different SD values. This is an interesting feature for data gathering scenarios. The coordinators located at higher depths collect fewer packets. Thus the active portion of the superframe have to be smaller than those of coordinators located at lower depths (near the PAN coordinator). One of the drawbacks of this solution is that it alters the standard. Additionally, the use of the CFP, as defined by the standard, is not possible any more and the coordinators have to use a higher layer protocol to establish these kind of CFPs. Lu *et al.* [3] proposed a beacon scheduling mechanism which uses the BOP approach.

In all of these approaches, the coordinators are already known before network start. This increases the deployment complexity, especially in large networks, since every beacon broadcaster has to be deployed in a precise location to ensure global network synchronization.

IV. TBOPS IN DETAILS

A. General description

We suppose that the PAN coordinator node is already defined. Once the PAN coordinator node starts, it starts immediately using the superframe structure by periodically broadcasting the beacons. The PAN coordinator defines the superframe length (BI), the maximum active period length (SD) and the maximum number of slots d_{MAX} in the BOP. These values are embedded in the beacon payload. The BOP

is used to transmit and receive beacons and the active period is used for regular message transmission and reception. The BOP is divided into d_{MAX} frame slots, numbered from 1 to d_{MAX} . The PAN coordinator always uses the first slot to transmit beacons. Other coordinators choose a frame slot (from 1 to d_{MAX}) for beacon schedules whenever they become beacon broadcasters. End devices always join the network as a leaf node.

The next sections detail the association and TBoPS procedures

B. Association

Figure 2 shows the finite state machine of the association procedure. It is executed by all nodes in the network. The root node is considered associated directly. Any other node starts the procedure by listening to the medium. It keeps listening until it receives either a beacon or a Hello message. In the first case, it synchronizes itself and follows the superframe structure. Moreover, it can determine when the WP begins. If it receives a hello message, it means that the network is already in the WP. In both cases, the node knows now when it should transmit frames. In the next state, the node sends a *HelloRequest* message and then starts the collection of *HelloResponse* messages for a certain period. *HelloResponse* procedure is illustrated in Algorithm 2. The node determines the RSSI of the received *HelloResponse* message and records all the information in the neighbour table. At the expiration of the collection period, the node selects the best father and sends an association request to it. If the list is empty, the node goes to the listen to medium state. If the association is granted by the selected father, the node is now associated. If the father is already a beacon broadcaster, it can start tracking its father's beacon. If the node is the first child of the father, it must wait until the father finds a beacon schedule. During this time, the node cannot accept association requests. If the father fails to find a beacon schedule, the node must dissociate from it and selects another parent. If the association is not granted or no response is received, the node shall select the next best father in the list and repeat the procedure.

A node A selects a father according to the following rules.

- $\{Neigh\}$ is the list of neighbour coordinators.
- A select a subset $\{EligibleNeigh\}$ from $\{Neigh\}$ which has $RSSI > Thresh$ where $Thresh$ is a constant defined to ensure a good quality link.
- A select from $\{EligibleNeigh\}$ the father which has the highest number of children. In case of multiple choices, it chooses the father with the minimum depth.

The association rules minimize the number of cluster-heads in the network.

C. TBoPS procedure

Algorithm 1 describes the TBoPS procedure. It is executed by a coordinator which becomes a father. The PAN coordinator is considered associated directly and starts transmitting beacons upon start. The procedure starts by sending a *HelloRequest* to message and then starts the collection of

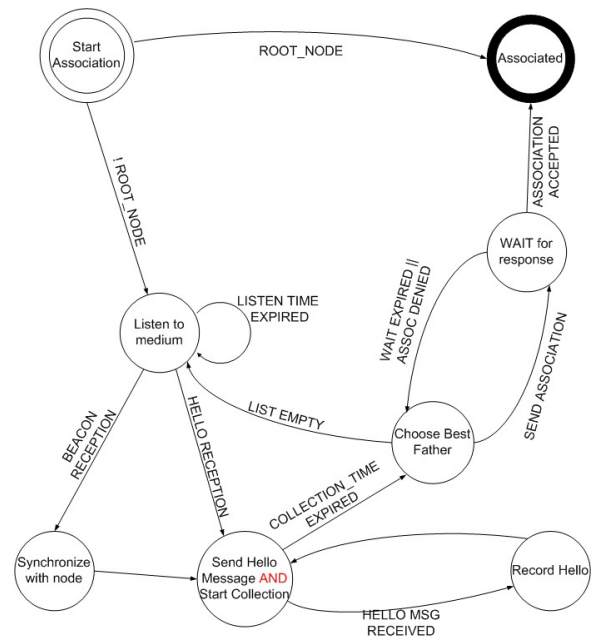


Fig. 2: Finite State Machine of the association mechanism.

HelloResponse messages for a certain period of time. Although the node may already have many of the information gathered previously, this step ensures that it has the most up to date information. The coordinator obtains neighbour coordinators, neighbours' neighbour coordinators and fathers' neighbour end devices beacon schedules. After the expiration of the collection period, the node executes *SelectBeaconSlot()* procedure to select a slot (illustrated in Algorithm 3). If a beacon slot is found, the node advertises it as being temporarily reserved. Then it waits for a period of time. If during this time, a neighbour node informs it, through a *ConflictAdvertisement*, that this schedule is already taken, the coordinator repeat this step. If the advertisement time expires, the beacon slot is definitely assigned to this coordinator. When beacon slot selection fails, the coordinator must inform the new son so that it can deassociate and find a new father.

V. CLOCK-DRIFT ANALYSIS

Let $t_i \in \mathbb{R}^+$ be the release time of a beacon and t_j such that $t_j > t_i$ be the next release time of a beacon by a cluster head. Let Δ be the beacon inter-arrival period for some cluster head, therefore, $\Delta = t_j - t_i$. In synchronized cluster-tree, beacons are the references for the system wide scheduling. Therefore, a stable beacon is at very base of system performance. The clock-drift in beacon generation may cause a synchronization errors if son nodes miss multiple beacons from their father. The clock drift of WSN nodes is independent from each other, the value of which is dependent of the quality of the clock oscillator used on the WSN board. Therefore, we need to provide a bound such that if multiple beacons are missed leaf nodes remain synchronized despite clock-drift for cluster heads (CH) in the scheduling path.

```

Step 1
Send Hello Request message
repeat
  Collect Hello responses and beacon slot
  advertisements
until (Collection time expires)

if No collected Hello responses then
  Go to Step 1
end

Step 2
if SelectaBeaconSlot() != -1 then
  Advertise selected slot as being temporarily reserved
  repeat
    if ConflictAdvertisement received then
      Go to Step 2
    end
  Advertise selected slot as being definitively reserved
  until (Advertisement time expired)
else
  Inform the new son that beacon selection failed
end

End of TBoPS procedure
Algorithm 1: TBoPS procedure

```

```

HelloResponse()
/* This is in addition to
information described in the
association procedure of the
standard*/
if I am a coordinator then
  Send
  - my beacon schedule
  - Neighbour beacon schedules
  - SD and BI
  - My address, depth and Number of children
  - The RSSI of the received HelloRequest
else
  - Send my father's schedule
end
end
Algorithm 2: HelloResponse procedure

```

Therefore, let Π be set of clocks representing beacon generation periods in a scheduling path such that $\Pi = \{\pi_1, \pi_2, \dots, \pi_n\}$ be a set of clocks representing a scheduling path having n nodes with π_1 being the clock of the root node. The periodic generation clock π_i for cluster head C_i is given as $\pi_i(t) = (1 \pm \delta_i) * t + \phi_i$. Where t is the time, $\phi \in \mathbb{R}^+$ is the phase-offset, and $\delta \in \mathbb{R}$ is the clock-drift.

If $\pi(t)$ is the clock time at t and $\pi(t + \Delta t)$ is the clock

```

SelectaBeaconSlot()
A={1.. $d_{max}$ } Remove from A beacon frame slots
already chosen/advertised by
- Neighbour coordinators
- Neighbours of neighbour coordinators (including
father)
- Fathers of neighbour end devices
if A is empty then
  return -1
else
  slot=select random from A
  return slot
end
end

```

Algorithm 3: Beacon slot selection procedure

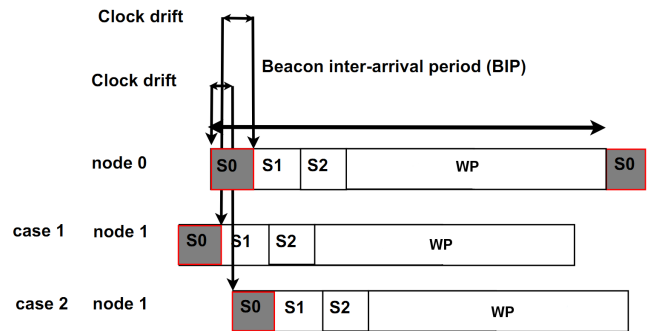


Fig. 3: Cases of clock drift for the receiving nodes.

time at some time $t + \Delta t$ then the drift can be given as:

$$\delta(t) = \frac{\pi(t + \Delta) - \pi(t)}{\Delta} - 1 \quad (4)$$

The value of $\pm\delta$ is a clock drift rate often expressed as a ratio in parts per million (ppm) or microsecond/second, this value can be found from a data sheet of a WSN board. Therefore, the change in the clock after a time interval Δt can be given as:

$$\Phi(\Delta t) = |\Delta t * \delta| \quad (5)$$

The clocks, therefore, can be represented as $\pi_i(t) = (1 \pm \delta_i)t$ (i.e., with zero offset).

Example 1: The crystal oscillators commonly found in WSN nodes usually have a drift rating of the order of value between 10^4 to 10^6 that is, two similar but uncalibrated oscillators will drift apart between 1 and 100 microseconds every second.

For the correct beacon synchronization condition when a father cluster head (FCH) is sending its beacon, all its child nodes heads (CN) must be in receiving mode. However, at the moment when a FCH starts sending its beacon in its allocated slot the CN may not be in their corresponding receive slot due to phase errors from clock drift, as illustrated in figure 3. In figure 3, if node 0 sends beacon during its the entire allocated slot, then node 1 may not be able to receive this

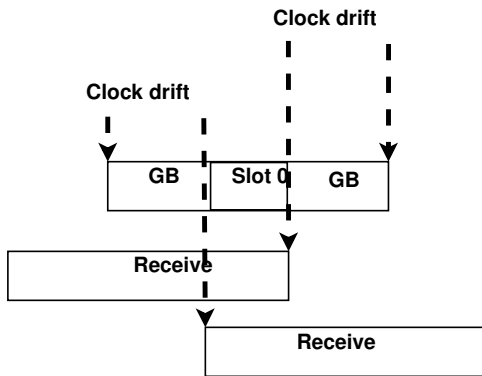


Fig. 4: Bound with drift and guard-bands.

beacon successfully, due to un-aligned slots. Taking the clock of node 0 as a reference, the clock of node 1 may drift in either direction as shown by case 1 and case 2 (i.e., backward or forward). In such a situation, the nodes may have a different view of the current slot number within the superframe. In order to cope with this problem we use guard-bands (GB), such that we have a bound for drift with respect to GB in an cluster-tree. This will ensure that a beacon are not transmitted during the entire sending slot but only in the middle, as illustrated in figure 4. Therefore, the FCH will not transmit at the beginning or the end of its slot (for duration defined by GB), in order to accommodate for the clock drift of receiving CN. If SL is the slot length in seconds and BTT is the beacon transmission time in seconds and GB is the guard-band in seconds, then the maximum phase error due to the clock drift between FCH and CN at any time t should be bound as:

$$\max\{\Phi\} \leq SL - BTT - GB \quad (6)$$

The cluster head C_i transmits beacons every $T(C_i)$ and its CN should be able to receive this beacon in its appropriate beacon slot. During this beacon period of $T(C_i)$ for C_i the phase error of CN can be found using equation (5) as:

$$\Phi(T(C_i)) = T(C_i) * |\delta| \quad (7)$$

Where δ is the clock drift of the CN. Thus, from equations (6 & 7) we have the bound for the waiting time of C_i such that its CN are synchronized as:

$$T(C_i) \leq \frac{SL - BTT - GB}{|\delta|} \quad (8)$$

This ensures that the slots remain synchronized between FCH and CN. Therefore, the superframe duration shall be less than this period.

VI. CLUSTER-TREE FORMATION PERFORMANCE

The simulation is setup in Matlab and consists of a scenario represented by a square area with each side equal to 100 meters. The WSN nodes are assumed to be uniformly distributed in this area. We assume unit disc model for the node communication ranges, where each node's communication range is assumed to be uniformly distributed within a range

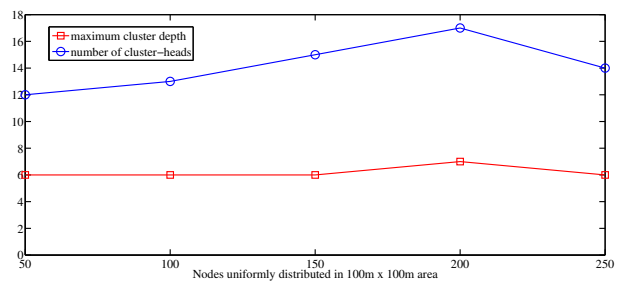


Fig. 5: Simulation results of cluster tree formation

of 25 to 30 meters (to simulate the different battery power levels of the nodes). A more realistic model like the one presented by Zuniga et al. [8] could be also used. We leave this as a future work. The simulations are carried out for various node densities in the square area. The simulations have the flexibility to fix the root node (by us) or choose one randomly and thereof begin the cluster tree formation based on the rules discussed in earlier sections. The objective of the simulation is to study the effectiveness of the proposed cluster-tree formation algorithm in terms of cluster-tree depth and the number of cluster-heads.

The simulation results are depicted in Figure 5. Figure 5 shows the results which have been averaged over six different simulations for each node density. We see that there is very little variation in the cluster-head count and the cluster-tree depth over different node densities. This result implies that the number of nodes in the clusters is increasing while the depth and cluster-head count remains some what constant, that is the density of nodes in the clusters is increasing. Which was our objective, to being with, when we formed the rules for tree formation in section IV-B.

VII. CONCLUSION

In this paper we proposed a new beacon scheduling approach based on collision free beacon frame of each coordinator (cluster head) in the network organized in the cluster tree topology. An algorithm is provided to construct an optimized cluster-tree. We used a dedicated period (beacon only period) to schedule beacon at the beginning of IEEE 802.15.4 superframe. We derived the maximum superframe length given clock drift and guard interval. We simulated effects of association rules in cluster head selections. Our algorithm keeps the number of cluster heads almost constant although the number of nodes increases in the network. As a future work, we will implement this mechanism and measure its performance in a real world scenario.

VIII. ACKNOWLEDGEMENT

This work was partially supported by Quasimodo project under No. ANR 2010 INTB 0206 01 in France and No. NSFC 61061130563 in China

REFERENCES

- [1] IEEE-TG15.4. IEEE-TG15.4 (2006). part 15.4: Wireless Medium Access control (MAC) and Physical Layer (PHY) specifications for low-rate Wireless Personal Area Networks (LR-WPANs), September 2006.

- [2] A. Koubaa, A. Cunha, and M. Alves. A time division beacon scheduling mechanism for ieee 802.15.4/zigbee cluster-tree wireless sensor networks. In *Real-Time Systems, 2007. ECRTS '07. 19th Euromicro Conference on*, pages 125–135, july 2007.
- [3] Juan Lu, Adrien Van den Bossche, and Eric Campo. A New Beacon Scheduling Mechanism for Mesh Wireless Personal Area Networks based on IEEE 802.15.4 (short paper). In *Emerging Technologies and Factory Automation (ETFA), Toulouse, 05/09/2011-09/09/2011*, page 4, <http://www.computer.org/portal/web/cscps>, septembre 2011. IEEE Computer Society - Conference Publishing Services.
- [4] Miklós Maróti, Branislav Kusy, Gyula Simon, and Ákos Lédeczi. The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems, SenSys '04*, pages 39–49, New York, NY, USA, 2004. ACM.
- [5] Panneer Muthukumaran, Rodolfo Paz, Rostislav Spinar, and Dirk Pesch. Meshmac: Enabling mesh networking over ieee 802.15.4 through distributed beacon scheduling. In Jun Zheng, Shiwen Mao, Scott F. Midkiff, Hua Zhu, Ozgur Akan, Paolo Bellavista, Jiannong Cao, Falko Dressler, Domenico Ferrari, Mario Gerla, Hisashi Kobayashi, Sergio Palazzo, Sartaj Sahni, Xuemin (Sherman) Shen, Mircea Stan, Jia Xiaohua, Albert Zomaya, and Geoffrey Coulson, editors, *Ad Hoc Networks*, volume 28 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 561–575. Springer Berlin Heidelberg, 2010. 10.1007/978-3-642-11723-7_38.
- [6] M.B. Uddin and C. Castelluccia. Toward clock skew based wireless sensor node services. In *Wireless Internet Conference (WICON), 2010 The 5th Annual ICST*, pages 1–9, march 2010.
- [7] Fang-Jing Wu, Yu-Fen Kao, and Yu-Chee Tseng. From wireless sensor networks towards cyber physical systems. *Pervasive and Mobile Computing*, 7(4):397–413, 2011.
- [8] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 517–526, oct. 2004.