IPP HURRAY!

www.hurray.isep.ipp.pt

Technical Report

An Explicit GTS Allocation Algorithm for IEEE 802.15.4

Jianxin Chen

Luis Lino Ferreira

Eduardo Tovar

HURRAY-TR-120901

Version:

Date: 09-05-2011

An Explicit GTS Allocation Algorithm for IEEE 802.15.4

Jianxin Chen, Luis Lino Ferreira, Eduardo Tovar

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:

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

Abstract

The IEEE 802.15.4 standard provides appealing features to simultaneously support real-time and non real- time traffic, but it is only capable of supporting real-time communications from at most seven devices. Additionally, it cannot guarantee delay bounds lower than the superframe duration. Motivated by this problem, in this paper we propose an Explicit Guaranteed time slot Sharing and Allocation scheme (EGSA) for beacon-enabled IEEE 802.15.4 networks. This scheme is capable of providing tighter delay bounds for real-time communications by splitting the Contention Free access Period (CFP) into smaller mini time slots and by means of a new guaranteed bandwidth allocation scheme for a set of devices with periodic messages. At the same the novel bandwidth allocation scheme can maximize the duration of the CFP for non real-time communications. Performance analysis results show that the EGSA scheme works efficiently and outperforms competitor schemes both in terms of guaranteed delay and bandwidth utilization.

An Explicit GTS Allocation Algorithm for IEEE 802.15.4

Jianxin Chen
Computer School, Nanjing University of Posts and Telecommunications,
No.66, Xinmofan Road,
Nanjing / China
jianxinchen06@gmail.com

Luis Lino Ferreira, Eduardo Tovar
CISTER Research Centre, ISEP
Polytechnic Institute of Porto
R. Dr. António Bernardino de Almeida, 431
4200-072 Porto / Portugal
{llf, emt}@isep.ipp.pt

Abstract

The IEEE 802.15.4 standard provides appealing features to simultaneously support real-time and non real-time traffic, but it is only capable of supporting real-time communications from at most seven devices. Additionally, it cannot guarantee delay bounds lower than the superframe duration. Motivated by this problem, in this paper we propose an Explicit Guaranteed time slot Sharing and Allocation scheme (EGSA) for beacon-enabled IEEE 802.15.4 networks. This scheme is capable of providing tighter delay bounds for real-time communications by splitting the Contention Free access Period (CFP) into smaller mini time slots and by means of a new guaranteed bandwidth allocation scheme for a set of devices with periodic messages. At the same the novel bandwidth allocation scheme can maximize the duration of the CFP for non real-time communications. Performance analysis results show that the EGSA scheme works efficiently and outperforms competitor schemes both in terms of guaranteed delay and bandwidth utilization.

1. Introduction

The IEEE 802.15.4 [1] is a *de facto* standard for low rate, low-power, short range Wireless Personal Area Networks (WPANs). It supports two medium access modes: contention access and contention-free access. This latter is possible in the beacon-enabled communication mode, where time is divided into a series of superframes.

In a superframe, the IEEE 802.15.4 allows non time-sensitive communications in a Contention Access Period (CAP) and time-sensitive communications in a Contention-Free access Period (CFP). However, during the CFP, the IEEE 802.15.4 defines that each superframe can provide at most seven Guaranteed Time Slots (GTSs), and that each GTS should be exclusively used by one device. The IEEE 802.15.4 divides the superframe duration into periods of equal duration called Time Slots (TS). During each of these periods, only one device is capable of transmitting.

Such factors limit the usage of the IEEE 802.15.4 in applications involving more than seven sensor nodes, each of which transmitting periodic messages requiring delay guarantees smaller than the duration of the superframe. An example of such an application could be human motion tracking [13], where 15 sensor nodes monitor the attitude of the various body parts (arms, legs, torso, etc.), with a data acquisition frequency of 60 Hz. That application obviously involves more than seven end devices and requires very small delays in the communications.

Currently, there are several solutions regarding the GTS allocation for time-sensitive applications [4, 8, 9]. But, to the best of our knowledge, none of those approaches can guarantee deadlines smaller than the superframe duration while supporting 15 end devices. In this paper, we propose a simple and effective scheme to overcome those limitations. The idea is to enlarge the bandwidth available on a superframe for time-sensitive applications, while maintaining, at least, the minimum CAP for non time-sensitive applications. To this purpose, the CFP is divided into mini time slots (mTS), smaller than a standard IEEE 802.15.4 TS, such that the devices can use them with more flexibility and efficiency. Our scheme also allows the use, by one node, of several mTS during a superframe.

We refer to this scheme as the Explicit GTS Sharing and Allocation scheme (EGSA), since the network scheduling is defined and configured on the sensor nodes prior to run-time. Detailed theoretical analysis and an EGSA application example prove that the ESGA scheme has the important advantage of satisfying more stringent delay constraints, while allowing more than seven devices to be simultaneously scheduled.

This paper is organized as follows. Section 2 provides an overview of the IEEE 802.15.4 standard, of the related work on its timing behavior and on the use of the CFP. In Section 3 we provide a motivation for the EGSA and a brief overview of its operation. Section 4 presents the EGSA scheme in detail, and Section 5 compares the performance of EGSA with other CFP allocation schemes. Finally, in Section 6, some conclusions are drawn.

2. Background

2.1. Overview of the IEEE 802.15.4 Standard

The IEEE 802.15.4 standard is a wireless communication protocol for personal area networks over relatively short distances. It defines the Physical layer and the Medium Access Control (MAC) layer protocols. In the beacon-enabled communication mode, time is divided into continuous superframes, as depicted in Figure 1.

A superframe is divided in two parts: the active and the inactive. The inactive part can be used to put sensor node components onto an energy saving mode, e.g. by turning off the radio circuitry. The active part is divided into 16 TSs, which are shared by the beacon frame, the Contention Access Period (CAP) and the Contention Free Period (CFP). During the CAP, a slotted Contention Sensing Multiple Access with Collision Avoidance (CSMA-CA) medium access scheme can be used for non time-sensitive applications.

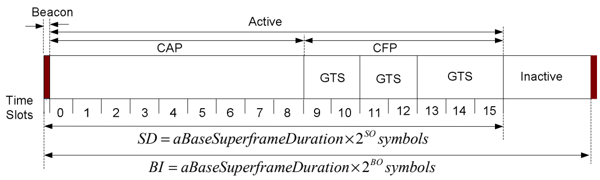

Figure 1 – IEEE 802.15.4 Superframe Structure

The first TS (time slot 0) starts after the beacon transmission. The CAP has a minimal duration of 440 symbols and each TS can only be used by one device. As previously mentioned, this latter aspect strongly limits its use in some applications.

During the CFP, up to seven GTSs are available to the devices with time-sensitive applications. The IEEE 802.15.4 standard defines that the Superframe Duration (*SD*) and Beacon Interval (*BI*) length, in symbols, are given by:

$$SD = aBaseSuperFrameDuration \times 2^{SO} \qquad (1)$$

$$BI = aBaseSuperFrameDuration \times 2^{BO} \qquad (2)$$

where, the parameters *SO* and *BO* are the Superframe Order and the Beacon Order, respectively. *BO* determines the superframe duration and *SD* determines the duration of the active part of the superframe. These parameters must comply to the following condition: 0 • *SO* • *BO* • 14. The setting of these parameters allows the system designer to define the active and inactive part of the superframe and the length of the TS, which can be computed as follows:

$$D_{TS} = \frac{960 \times 4 \times 2^{SO}}{16} = 240 \times 2^{SO} \qquad (3)$$

Note that in this paper, the notation D_x denotes a length in bits, and t_x denotes a duration in time.

The IEEE 802.15.4 defines the minimum length of a superframe, *aBaseSuperframeDuration*, to be equal to 960 symbols. On the 2.4 GHz frequency band, each symbol represents 4 bits, hereafter we assume the use of this band in all our formulations, since this is the most commonly used band. Therefore, the superframe duration varies between 1536 ms and 251.66 s, when *SO* and *BO* are equal to zero and to 14, respectively. Also note that the duration of a TS increases exponentially with *SO*, which can results in a waste of bandwidth if the message size is maintained.

In the beacon-enabled communication mode, the PAN coordinator sends a beacon frame at the beginning of each superframe, which is used to synchronize all devices in a WPAN. The beacon frame also carries the GTS allocation information.

2.2. Related Work

In the IEEE 802.15.4 standard, the GTS allocation is performed in a first-come-first-served fashion, which is far from being optimal when it is necessary to give more priority to certain messages. To better use the bandwidth for time-sensitive applications, several solutions have been proposed.

In [6], the authors proposed an Adaptive GTS Allocation (AGA) scheme. The AGA scheme prioritizes different devices according to their recent GTS usage.

In [8-10], the authors used network calculus to evaluate the delay bounds [8] and to prove its predictability. A power-efficient superframe parameter selection method was proposed in [10] to meet the delay requirements of real-time flows using the GTS allocation. Most importantly, in [9] the authors proposed an implicit GTS Allocation Mechanism (i-GAME), which accomplishes higher bandwidth utilization and allows the sharing of GTS.

To improve the bandwidth utilization, in [7] the authors, proposed a greedy GTS Allocation algorithm. Additionally, in [4], the same authors proposed a simple and efficient method in which the CFP is always divided into 16 mini time slots, independently of the CAP length and without changing the beacon frame structure. This mechanism increases the flexibly and adaptability of the GTS allocation for time-sensitive applications. In this paper we will refer to this scheme as the 16-mTS scheme.

To satisfy the delay constraints of industrial applications, in [5] the authors proposed to allocate the GTSs using the Earliest Due Date (EDD) scheduling algorithm. The authors in [11] also proposed an implementation of the Earliest Deadline First (EDF) scheduling algorithm and in [12] they improved it by distributing the GTSs of a transaction over multiple superframes. In [15] the authors proposed an offline scheduling algorithm for periodic messages in which the specific parameters, such as beacon order, superframe order and GTS allocation, were defined in way satisfying the delay constraints.

All of the previous related works considered the case where at most seven devices were able to use the available GTS bandwidth in one superframe. Nonetheless, in [2] and [14] the authors adapted the superframe structure in order to provide an allocation based on Time Division Multiple Access (TDMA) principles, in which the CAP does not exist and the data frame structure is simplified. However, such enhancement weakens the advantages of the IEEE 802.15.4 due to the absence of opportunities for non real-time traffic.

Therefore, in this paper, we propose the EGSA scheme. Our novel scheme has the following notorious features: (i) it is able to guarantee predictable communication delays; (ii) it does it so for more than seven devices; and (iii) at the same time it maximizes the available bandwidth for non time-sensitive applications. This is a set of features not simultaneously fulfilled by any other approach. Additionally, our proposal maintains compatibility with commercial-off-the-shelf (COTS) IEEE 802.15.4 devices for non time-sensitive traffic.

3. EGSA Overview

3.1. Motivation

We consider an IEEE 802.15.4 network with a star topology, operating on a beacon-enabled mode, where a coordinator node communicates with $(m + n)$ sensor nodes. The n sensor nodes execute a periodic sampling application which runs with a period of T_p. Each sample is encapsulated into a packet with a fixed length of B bits, which should be transmitted to the coordinator within a deadline (d). The other m sensor nodes perform non time-sensitive applications. The $m + n$ sensor nodes can use the CAP to transmit non time-sensitive data, and only n sensor nodes use the CFP to transmit time-sensitive data.

We focus on the GTS allocation in the CFP to satisfy the delay constraint of the n message flows, particularly when the required delay is less than the beacon interval ($d < t_{BI}$, where t_{BI} is the BI duration) and the number of nodes is higher than seven, since the IEEE 802.15.4 standard defines that at most seven GTSs are available in one superframe. It is important to note that we do not put any restriction on d in relation to T_p; d can be higher or lower than T_p.

If $n > 7$, a possible solution would be to use several superframes to provide enough GTSs for those n devices. However, in that case the delay would increase linearly with the number of superframes required for the transmission of the messages by all nodes. Such kind of solution has been proposed, for example, on the i-GAME approach [9], but if the delay constraint is $d < t_{BI}$, then such an approach would not generate a feasible schedule.

The i-GAME approach allows GTS sharing among multiple devices, which favours scenarios where the flows from those devices have different rates. When the flows have the same rates, the TSs contained in the shared GTS are allocated in a round-robin fashion. Consequently, this will result in a delay bound close to $(\lceil n/k \rceil \times t_{BI})$, where k is the number of TSs allocated by i-GAME during the CFP, which cannot be higher than 7.

Some previous research works about GTS allocation [3, 12] proposed a solution, where a device sends a GTS request to the coordinator when it has time-sensitive data to transmit; the coordinator accepts the request only if there is enough bandwidth available. But such GTS request/permission transaction is inefficient for systems with periodic message transmissions.

We rather propose an off-line bandwidth allocation algorithm, where nodes are configured with GTS allocations prior to run-time. Figure 2 represents a possible allocation scheme for 3 sensor nodes that transmit their information during the CFP, where the maximum delay between consecutive transmissions (d_g) from a sensor node is measured from the start of the transmission opportunity for sensor S1 until the start of the next transmission opportunity.

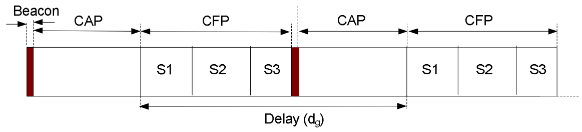

Figure 2 – Message transmission and delay constrains

To provide low delay guarantees we assume that SO is equal to BO, thus eliminating the inactive portion of the superframe. Let t_{CAP} denote the time required for the transmission of the beacon frame plus the minimum time specified for the CAP. Then, the delay (d_g) which can be guaranteed for n sensor nodes is given by:

$$d_g = t_{CAP} + n \times \left\lceil \frac{B + D_{IFS}}{D_{TS}} \right\rceil \times t_{TS} \qquad (4)$$

where D_{IFS} is the length, in bits, of the Inter-Frame Spacing (IFS) that is required after each transmission. The ceiling function results from the fact that each transmission occupies a multiple number of a TS. Since t_{TS} varies exponentially with SO, so does d_g. Additionally, this formulation is only valid for $n < 7$.

The IEEE 802.15.4 standard has supplementary inconveniences: since the duration of a TS increases exponentially with SO (recall eq. (3)) and a TS can only be used by one station, in systems where the message size does not change an additional waste of bandwidth will occur.

3.2. EGSA Fundamentals

Our GTS allocation algorithm aims at solving the case of scheduling a set of message flows from n sensor nodes, particularly in the following conditions:

1) the number of stations with real-time requirements is higher than seven ($n > 7$);
2) the deadline associated with each message stream is smaller than the beacon interval duration ($d < t_{BI}$).

Additionally, our approach has also the objectives of maintaining compatibility with standard IEEE 802.15.4 during the CAP and of achieving a more efficient use of the available bandwidth.

To fulfil the conditions and objectives we propose to change the IEEE 802.15.4 TS length during the CFP according to the message flow characteristics such that the length of each message exactly fits into the new time-slot length (D_{mTS}), hereafter referred as mini-Time-Slot (mTS).

With the objective of guaranteeing small delay bounds, we also allow, during one superframe, multiple transmissions from the same node; this leads to an organization of message transmissions in blocks, during which all nodes involved in the application are able to transmit. This solution also allows the allocation mechanism to better adapt to the use of different SO values. Figure 3 depicts a possible allocation scheme using

EGSA for a system with 15 sensor nodes. The CAP is still maintained and can be used by COTS IEEE 802.15.4 nodes for non real-time communications; its length computation is later on discussed and detailed in sections 4.2 and 4.4.3 of this paper.

As previously mentioned, the CFP is divided into several time blocks, during which all nodes can transmit messages in sequence, from sensor S1 until sensor S15. To each sensor node, a mTS of length D_{mTS} is granted for the transmission of its messages. Its size is also adjusted accordingly to the message length (see Section 4.1). The number of blocks (N) mainly depends on the periodicity of the message streams (details are provided in Section 4.4.1).

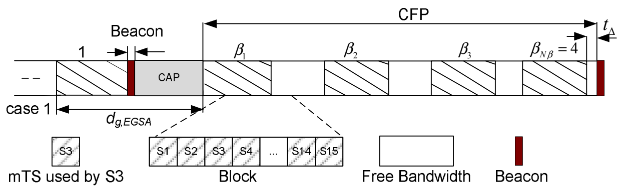

Figure 3 – EGSA allocation example

In all blocks the same sensor sequence is repeated. EGSA guarantees that the delay between blocks is never bigger than $d_{g,EGSA}$, and the message periodicity. Nevertheless, some jitter appears due to the CAP, and therefore variations on the interval between contiguous blocks, which have to be a multiples of a mTS, may result (see Section 4.4). Each block is referenced as $_x$, where x is the block index, assigned from the end of the CAP, as depicted in Figure 4. The maximum delay, using the EGSA scheme, between two consecutive transmissions ($d_{g,EGSA}$) of the same message flow, takes place when one transmission occurs on the block prior to the beacon and the next transmission occurs just after the CAP, between block $_1$ and block $_4$ in the example of Figure 3. This delay is given by:

$$d_g = t_{CAP} + t_\Delta + n \times t_{mTS} \tag{5}$$

where t_{mTS} is the duration of one mTS (see Section 4.1) and t represents the CFP time that cannot be used to convey any traffic (to be detailed in Section 4.4.1).

The data being transmitted during a superframe might, in some situations, exceed the duration of the CFP. To accommodate that, it is possible to increase the duration of a CFP by manipulating the SO parameter. However, increasing that value will also increase the CAP duration and consequently $d_{g,EGSA}$. Therefore, SO can only vary between its minimum value (SO_{min}) and its maximum value (SO_{max}) (Section 4.3).

4. EGSA Details

Since one of our objectives is to accomodate message streams with deadlines smaller than the superframe length, we assume that SO is equal to BO, thus eliminating the inactive portion of the superframe. The main advantage is a reduction on the jitter introduced on the block allocation. Nevertheless, the formulations in this paper can be easily adapted to cases where SO and BO are different. The remainder of this section addresses in detail the computations inherent to Algorithm 1.

4.1. Mini-Time-Slot Length Determination

The first step of the algorithm is to determine the duration of a mTS:

$$D_{mTS} = B + D_{IFS} \tag{6}$$

where B is the length in bits of a message and D_{IFS} is the length in bits required by a node after transmitting or receiving a message. It is important to note that D_{mTS} only depends on the message stream parameters and on D_{IFS}, both do not vary with any other parameter which can be changed by EGSA, like SO. Also note that when D_{mTS} is not a multiple of the CFP length, a small gap of length t exists which is smaller than a mTS and therefore cannot be used (see Figure 4).

After obtaining a value for the CFP length (eq. (1) and eq. (2)), the CFP is divided into N_{mTS} mini-time-slots, which can be calculated by:

$$N_{mTS} = \left\lfloor \frac{t_{CFP}}{t_{mTS}} \right\rfloor \tag{7}$$

```
Algorithm 1
System =(m, n, B, T_p, d)
Determine D_mTS;
Determine the set of possible SO values
   {SO_min; SO_max}
SO = SO_max;
Sh = False;
     // if the allocation succeeds or fails
While (SO ≥ SO_min) {
    Determine the CFP, CAP and mTS duration
    Determine the number of blocks
    Allocate the blocks;
    If (d ≥ d_g,EGSA){ // allocation finished
        Sh=True;
    } else {
        Obtain the starting mTS of blocks
    }
    if (Sh=True) break;  // success.
    SO = SO - 1;
}
if Sh = True {
     adjust the CAP length
}
```

4.2. CAP and Beacon Length

The strategy behind EGSA is, at first, to reserve the smallest possible CAP length, thus maximizing the length of the CFP, available for real-time traffic. But when Algorithm 1 finishes the user might be able to increase the CAP according to the rules defined in section 4.4.1.

The standard defines that the CAP should have a length of at least 440 symbols (220 bytes), and that it should be a multiple of a TS. The beacon frame has a minimum length of 13 bytes plus the physical layer overhead (totalizing 25 bytes) and its transmission occurs prior to the start of slot 0. Since we assume that the information related to the traffic allocation is transmitted to the nodes prior to run-time, then our scheme does not need to use the beacon frame to transmit traffic allocation data. Consequently, we define the minimum CAP length (D_{minCAP}) to be equal to 245 bytes, which can be converted to time length as follows:

$$t_{minCAP} = \left\lceil \frac{D_{minCAP}}{D_{TS}} \right\rceil \times t_{TS} = \left\lceil \frac{245}{30 \times 2^{SO}} \right\rceil \times t_{TS} \qquad (8)$$

4.3. Possible *SO* Values

Based on the system parameters *n, B* and *d*, we are now able to obtain the SO_{min} and SO_{max} values that satisfy the delay constrains defined for the traffic, i.e., d_g • *d*. To that purpose eq. (5) can be rewritten as follows:

$$d_g = \frac{\left\lceil \frac{245}{30 \times 2^{SO}} \right\rceil \times 30 \times 2^{SO} + n \times t_{mTS}}{R} \qquad (9)$$

where *R* is the bit rate in bits per second. Starting from $SO = 0$ until $SO = 14$ and testing eq. (8) for those values, it is possible to find the set of *SO* values that satisfy that equation.

Nonetheless, the bandwidth required by the message streams might not fit into the available CFP. Therefore, it is also necessary to evaluate following condition:

$$t_{CAP} + n \times t_{mTS} \leq \frac{SD \times 4 - D_{minCAP}}{R} \qquad (10)$$

Eq. (10) guarantees that at least one transmission from a message stream fits into the CFP. Note that the number 4 represents the conversion between symbols (in which SD is expressed) to bits.

4.4. Block Allocation

We had chosen to start our approach by setting $SO = SO_{max}$, since potentially in this way we maximise the bandwidth utilization during the CFP.

4.4.1. Computing the Number of Blocks

Transmission opportunities are organized into blocks, which contain *n* transmission opportunities, of length t_{mTS}. The number of blocks (*N*) can be defined as a function of t_{BI} and message stream periodicity (*T*) as follows:

$$N_\beta = \left\lfloor \frac{t_{BI}}{T} \right\rfloor \qquad (11)$$

Note, however, that if $N \times n \times t_{mTS} < t_{CFP}$ then the allocation of such traffic is not possible.

The allocation scheme leads to the presence of free gaps on the CFP. These gaps can be divided into two different types. One (*t*) is due to the division of the CFP into several mTS, which are not integer devisors of the CFP length. This gap is equal to: $t_{CFP} - \left\lfloor \frac{t_{CFP}}{t_{mTS}} \right\rfloor \times t_{mTS}$. The second gap type has a length that is multiple of a mTS, depending on the number of blocks. The total duration of this gap can be calculated as $t_{CFP} - t_\Delta - N_\beta \times n \times t_{mTS}$.

Hereafter, we assume initially that a gap *t* occurs at the end of the CFP and that it is not used by any node to transmit messages.

4.4.2. Block Allocation

We assume that the mTSs are indexed from 1 to N_{mTS}, being the first mTS the one that starts just after the CAP. The EGSA algorithm allocates blocks, placing the start of block $_N$ at the end of the CFP at position $N_{mTS} - n - 1$.

Block $_1$ must start on a mTS, which assures that the delay constrain (*d*) for the traffic is meet. Therefore, it is possible to define the start position for block 1 as follows:

$$S_1 = \begin{cases} \left\lfloor \frac{d - t_\Delta - t_{CAP}}{t_{mTS}} \right\rfloor, & S_1 < N_{mTS} - N_\beta \times n \\ N_{mTS} - N_\beta \times n, & S_1 \geq N_{mTS} - N_\beta \times n \end{cases} \qquad (12)$$

The second line of eq. (12) guarantees that the sampling frequency can be achieved since *N* depends on *T*. If S_1 is larger than zero, then there is a possible scheduling allocation for this system, otherwise the procedure must be restarted with $SO = SO - 1$.

The $n - 2$ blocks between $_1$ and $_N$ can now be allocated by guaranteeing that the time difference of adjacent blocks ($t_i - t_{i-1}$) does not exceed the traffic periodicity. To do so, we must define the size of the gaps between consecutive blocks. Between blocks $_1$ and $_N$, there are $N_{mTSfree}$ free mTS, where $N_{mTSfree}$ can be calculated as $N_{mTS} - N \times n - S_1 + 1$.

We are now able to obtain the start mTS of each block between block 2 and block $N - 1$ as follows:

$$S_{i+1} = S_i + n + \left\lfloor \frac{N_{mTS_{free}}}{N_\beta - 1} \right\rfloor + \delta_i, 2 \leq i \leq N_\beta - 1 \qquad (13)$$

Parameter $_i$ is required since in most situations it is not possible to evenly distribute the free mTS between the intervals. To calculate $_i$, let *r* be an integer number, so that $r = N_{mTSfree}$ *mod* $(N - 1)$, if $N_{mTSfree} < (N - 1)$ and $r = N_{mTSfree}$, if $N_{mTSfree}$ • $(N - 1)$, then:

$$\begin{cases} \delta_i = 1, & i \leq r \\ \delta_i = 0, & i > r \end{cases} \qquad (14)$$

The example allocation in Figure 4 shows a possible EGSA allocation scenario where the division of the CFP into mTS results in an unused small gap at the end of the CFP with duration of *t* .

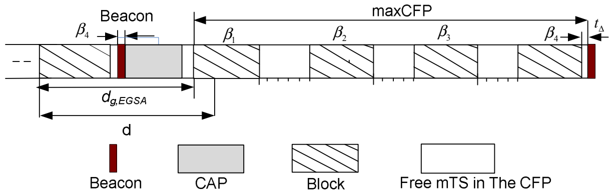

Figure 4 – EGSA final allocation

In order to guarantee the periodicity required by the message stream, four blocks must be allocated ($_1$, $_2$, $_3$, $_4$) on the CFP. The first block starts at mTS 4, which results from the application of eq. (12). By applying eq. (13) the start of the remaining blocks can be calculated. Since the number of unallocated mTS is not divisible by 3, the free space between blocks is different, being equal to 3 in the gap between $_1$ and $_2$ and equal to 4 on the other gaps.

Eq. (8) can now be rewritten taking into account that the maximum delay (d_g) occurs between transmissions in blocks $_N$ and $_1$.

$$d_g = t_{CAP} + t_{\Delta} + S_1 \times t_{mTS} \tag{15}$$

The time gap between consecutive blocks is always less than the period of the message stream, a consequence of the formulation to determine the number of blocks.

4.4.3. CAP length Final Adjustment

The application of the EGSA algorithm may result on a final allocation similar to Figure 4, in which we assumed a CAP length equal to the minimal CAP length. But, in many situations the start of block $_1$ might allow to increase the CAP duration, thus favouring non-real-time applications.

If required by the system designer the following procedure can be applied to readjust the CAP length. This procedure depends on the starting position for block $_1$, the duration of a mTS and on the length of TS.

The new duration of the CAP must be a multiple of a TS, it can be extended until t_{maxCAP}, which can be calculated by:

$$t_{maxCAP} = t_{minCAP} + \left\lfloor \frac{S_1 \times t_{mTS}}{t_{TS}} \right\rfloor \times t_{TS} \tag{16}$$

If it is not possible to extend the CAP, then the system designer can opt to make S_1 equal to zero, or to a value smaller than the one initially obtained. This kind of setting might be interesting in situations where the communication jitter is important to the system design.

If there is the need to set a new value for S_1 then the calculations, related to the start mTS of each block, described in Section 4.4.2, must be redone.

5. Performance Evaluation

In this section, we use an application example to evaluate the EGSA performance and compare it with i-GAME [9] and 16-mTS [3] schemes. The comparison focuses on the worst-case delays and on the bandwidth utilization.

5.1. Application Example

As an application example consider a motion tracking system, in which 15 wearable wireless sensors nodes are used to track the human motion [13]. Each sensor node contains a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer. Measurements from the 3-axis sensors are acquired periodically by a 10-bit Analog-to-Digital Converter (ADC) of the microcontroller and transmitted to a sink node, which is connected to a PC.

The reconstruction of the human motion is performed on a PC, which graphically displays on screen a reconstruction of the movement. For a smooth view of the human motion, a 60 Hz image update rate is required, which directly translates into a sampling rate of 60 Hz. Since this application has an estimated end-to-end delay of around 150 ms, we estimate that to fulfil this objective the system can tolerate a network delay of 30 ms.

Each message from a sensor node contains the data acquired from the three 3-axis sensors (90 bits) packed into 12 bytes (thus wasting only 6 bits). The IEEE 802.15.4 also requires a message to contain a Physical and MAC Layer headers, with a length of 6 bytes and 5 bytes, respectively. Therefore, each message transmitted by the EGSA scheme has a total length of 23 bytes. On the other schemes message size varies depends on SO, since several acquisition results are packed into the same frame. The radio transmission rate is 250 kbps in all schemes.

5.1.1. EGSA Allocation

In this scenario the application of the EGSA scheme results in a SO_{max} value of 4, corresponding to 16 TS, each with 15.23 ms. The EGSA assumes that the CAP length is equal to 1 TS and redefines the division of the CFP into 248 mTS, each with a length of 29 bytes (eq. (6)). These mTS are organized into 15 continuous blocks (eq. (11)), each one containing 15 mTSs.

The initial allocation of EGSA places all blocks adjacently, starting from the end of the CFP (line 6 of algorithm 1). This allocation occupies 208.8 ms, thus leaving 21.6 ms free on the CFP. But, the delay calculated by eq. (15) is equal to 36.96 ms, which is higher than the maximum allowed delay, consequently it is necessary to adjust the allocation.

By applying eq. (12) the starting position for block $_1$ is mTS 15 and for the last block ($_{15}$) is mTS 234. The remaining blocks allocation start mTS (S_2, S_3, …, S_{14}) are obtained by applying eq. (13) and eq. (14). Between blocks $_2$ to $_{10}$ there is a gap of 1 mTS, but between the remaining blocks there is no gap.

Figure 5 illustrates the final allocation resulting from the use of the EGSA scheme.

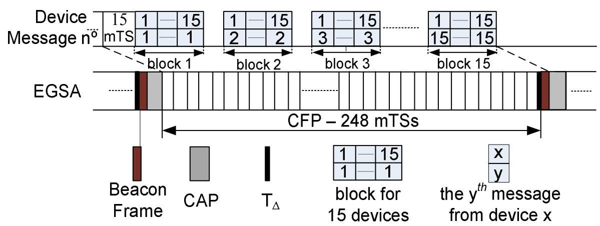

Figure 5 – Traffic allocation when using the EGSA

There are 15 mTS available to increase the CAP but it can only be increased by a multiple of a TS which would required 17 mTSs (eq. (16)), consequently the CAP length must be maintained equal to 1 TS.

5.1.2. i-GAME allocation

The i-GAME algorithm operates at TS scale. Consequently, it only allocates sensor data transmissions for each sensor node per TS. In the first TS, of the CFP, all 15 messages stored in station 1 are transmitted. Then, in the next TS, the messages from station 2 until station 15 (Figure 6).

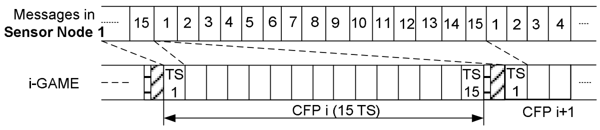

Figure 6 – Traffic allocation when using i-GAME

5.1.3. The 15 readings, from the three sensors, are compressed into a 186 bytes message, which already includes the protocol overhead. Each message from one sensor node is transmitted during the TS allocated to that node. Due to the compression into a single message a transmission from one sensor node only occupies 38.75 % of a TS duration.16-mTS Allocation

In this particular example, the 16-mTS is very similar to the i-GAME case. However, since 16-mTS scheme always divides the CFP into 16 mTS there is one unused mTS. The main advantage of this approach is that the unused mTS can be used to convey any other traffic, contrarily to i-GAME where the unused time cannot be reused by other sources. Nevertheless, in both cases the unused time in each TS or mTS can only be used to convey messages from the same sensor node.

Figure 7 shows how the traffic generated by one sensor node fits into the CFP when using the 16-mTS scheme. Note that we decided to localize the free mTS at the middle of the CFP in order to minimize the overall jitter.

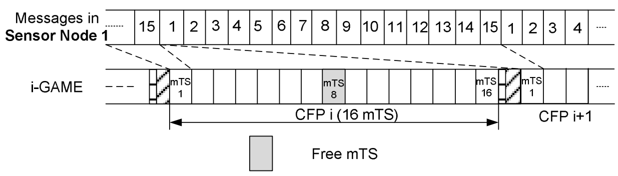

Figure 7 – Traffic allocation when using the 16 mTS

5.2. Bandwidth Utilization

The algorithms being compared have different performance in what concerns bandwidth utilization. The main advantage of EGSA relies on its capability of dividing the time into mTS more adjusted to the bandwidth required by the sensor nodes.

We defined two metrics to compare the performance of the three algorithms. One is the utilization of the CFP, which measures the capability of the protocol to accommodate other message flows:

$$U_{CFP} = \frac{t_{tx}}{t_{CFP}} \quad (17)$$

In eq. (17), t_{tx} is the time required for the transmission of all messages during the CFP using one of the protocols being compared and its associated overhead.

Another metric is the capability of the protocol to accommodate other message flows:

$$U_{slots}=\frac{N_{TSunu}\times t_{TS}}{t_{CFP}} \quad (18)$$

where N_{TSunu} is the number of unused TS or mTS (depending on the schema being used) during the CFP.

Figure 8 contains the plots for the bandwidth utilization as a function of *SO*. In i-GAME and 16-mTS the data is compacted into a single message per beacon interval. Consequently, the bandwidth utilization of these algorithms is lower, but the free space that remains in each time slot can only be used by the sensor node to which the time slot is allocated.

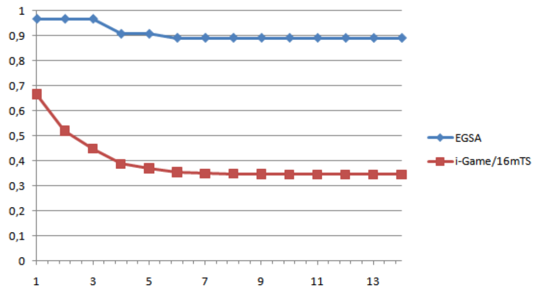

Figure 8 – Utilization of the CFP

Another aspect that can be evaluated is the percentage of free time-slots per beacon interval. Figure 9 shows that i-GAME uses all of the available time slots to schedule its transactions independently of the SO parameter. In 16-mTS just one TS is not used.

Although EGSA uses more bandwidth, the mTS length is smaller and used to its full extent, consequently more free mTS are available to convey traffic from any of the system nodes.

5.3. Delay bounds

In EGSA the delay bound is given by Eq. (5), and it manly depends on the CAP duration and on the length of a block of messages. In the other two schemes being compared the delay bounds formulations can be found in [9] and [3].

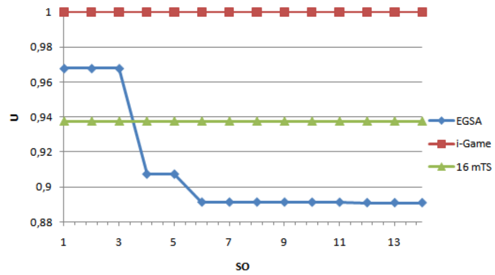

Figure 9 – Free TS per beacon interval

On the scenario being used for the comparison, with SO = 4, the delay bound of EGSA is 19.20 ms, which compares very favourably with 16-mTS (232.03 ms) and i-GAME (244.53 ms). Figure 10 shows a graphic comparing the delay bounds of EGSA with the other two approaches when *SO* varies. Note that the graphic uses a logarithmic scale on the vertical axis.

From Figure 10, it is easy to see that when *SO* < 4, the delay bounds provided by EGSA have small variations. This is due the fact that in EGSA, the delay bound depends mostly on the duration of the minimum CAP. While for i-GAME and 16-mTS schemes, the delay bounds increase exponentially with SO, since they mainly depend on the duration of a TS. When $SO = 1$, 16-mTS and i-Game provide a delay bound of 52.54 ms and 69.02 ms, respectively, which still cannot guarantee the delay of the human motion tracking application used in this evaluation. But in that case, the EGSA scheme can provide an even lower delay bound of 16.8 ms. When $SO > 4$, the EGSA cannot satisfy the delay constraint of the application; nevertheless it is important to note that the delay bound increases much more slowly than in the case the other two approaches.

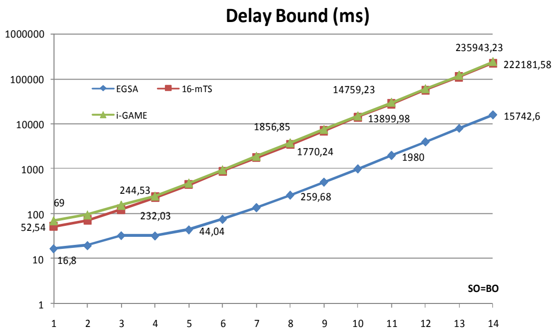

Figure 10 - Delay Bounds comparison

6. Conclusions

In this paper we presented the EGSA scheme, a GTS allocation scheme that enables an IEEE 802.15.4 network to guarantee the delay constraints in scenarios consisting of more than seven devices. In this paper we prove the main features of the novel allocation scheme: i) it enables a low latency communication system by using the maximized CFP in a beacon-enabled mode; ii) in the CFP, the time slots (in this paper redefined as mini time slots (mTSs)) are defined according to the application requirements, not only improving the bandwidth utilization, but also reducing the time exclusively used by one device. We compared the EGSA with other well known scheduling strategies for IEEE 802.15.4, showing its advantages.

Acknowledgments. This work is partially funded by the CMU-Portugal Program, funded by National Funds (PT), through the FCT - Portuguese Foundation for Science and Technology, under the Carnegie Mellon Portugal Program, with Grant ref. FCT-CMU-PT/0012/2006 and the CISTER Research Unit (608FCT) funded by FEDER funds through COMPETE (POFC - Operational Programme 'Thematic Factors of Competitiveness) and by National Funds (PT), through the FCT - Portuguese Foundation for Science and Technology.

References

[1] IEEE 802.15.4 standard: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), October 2003.

[2] F. Chen., "Improving IEEE 802.15.4 for low-latency energy efficient industrial applications", Proc. of Echtzeit 2008 - Fachtagung der GI-Fachgruppe Echtzeitsysteme, Boppard, Germany, Nov. 2008, pp. 61–70.

[3] L. Cheng., "IEEE 802.15.4 MAC protocol study and improvement", PhD thesis, Georgia State University, Dec. 2007.

[4] L. Cheng, A. G. Bourgeois, X. Zhang, "A new GTS allocation scheme for IEEE 802.15.4 networks with improved bandwidth utilization", in Int. Symposium on Communications and Information Technologies, Oct. 2007, pp. 1143–1148.

[5] M. Hameed, H. Trsek, O. Graeser, and J. Jasperneite, "Performance investigation and optimization of IEEE 802.15.4 for industrial wireless sensor networks", Proc. of the International Conference on Emerging Technologies and Factory Automation, Hamburg, Germany, September 2008, pp. 1016–1022.

[6] Y. Huang, A. Pang, T.Kuo, "AGA: Adaptive GTS allocation with low latency and fairness considerations for IEEE 802.15.4", Proc. of IEEE International Conference on Communications (ICC), volume 9, Istanbul, Turkey, Jun. 2006, pp. 3929–3934.

[7] L. Ko, Z. Chou, "A novel multi-beacon superframe structure with greedy GTS allocation for IEEE 802.15.4 wireless pans", in Proc. of IEEE Wireless Communication & Networking Conference (WCNC), Hong Kong, March 2007, pp. 2328–2333.

[8] A. Koubaa, M. Alves, E. Tovar, "GTS allocation analysis in IEEE 802.15.4 for real-time wireless sensor networks", In Proc. of the 14th International Workshop on Parallel and Distributed Real-Time Systems, Island of Rhodes, Greece, April 2006.

[9] A. Koubaa, M. Alves, E. Tovar, "i-game: An implicit GTS allocation mechanism in IEEE 802.15.4 for time sensitive wireless sensor networks", in Proc. of the 18th Euromicro Conference on Real-Time Systems, Dresden, Germany, July 2006, pp. 183–192.

[10] A. Koubaa, M. Alves, and E. Tovar, "Energy and delay trade-off of the GTS allocation mechanism in IEEE 802.15.4 for wireless sensor networks", International Journal of Communication Systems, 20(7):791– 808, July 2007.

[11] A. Mishra, C. Na, D. Rosenburgh, "On scheduling guaranteed time slots for time sensitive transactions in IEEE 802.15.4 networks". In Proc. of the IEEE Military Communications Conference, 2007 (MILCOM 2007), October 2007, pp. 1–7.

[12] C. Na, Y. Yang, A. Mishra, "An optimal GTS scheduling algorithm for time-sensitive transactions in IEEE 802.15.4 networks", International Journal of Communication Systems, 52(13):2543–2557, September 2008.

[13] A. Young, M. Ling, D. Arvind, "Orient-2: A wireless real-time posture tracking system using local orientation estimation", In Proc. of the 4th Workshop on Embedded Networked Sensors, June 2007, pp. 53–57.

[14] F. Chen, T. Talanis, R. German, F. Dressler, "Real-time Enabled IEEE 802.15.4 Sensor Networks in Industrial Automation", Proc. of the International Symposium on Industrial Embedded Systems 2009, July 2009, pp. 136–139.

[15] S. Yoo, P. Chong, D. Kim, *et al.*, "Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4", IEEE Transactions on Industrial Electronics, Vol. 57, No. 11, November, 2010, pp. 3868–3876.