A Simulation Model for the IEEE 802.15.4 Protocol: Delay/Throughput Evaluation of the GTS Mechanism

Petr Jurčík[1], Anis Koubâa[1,2], Mário Alves[1], Eduardo Tovar[1], Zdeněk Hanzálek[3]

[1]IPP-HURRAY! Research Group, Polytechnic Institute of Porto, Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto, PORTUGAL

[2]Al-Imam Muhammad Ibn Saud University, College of Computer Science and Information Systems, 11681 Riyadh, SAUDI ARABIA

[3]Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University, 121 35 Prague, CZECH REPUBLIC

{petr, mjf}@isep.ipp.pt, {akoubaa, emt}@dei.isep.ipp.pt, hanzalek@fel.cvut.cz

Abstract - **The IEEE 802.15.4 protocol has the ability to support time-sensitive Wireless Sensor Network (WSN) applications due to the Guaranteed Time Slot (GTS) Medium Access Control mechanism. Recently, several analytical and simulation models of the IEEE 802.15.4 protocol have been proposed. Nevertheless, currently available simulation models for this protocol are both inaccurate and incomplete, and in particular they do not support the GTS mechanism. In this paper, we propose an accurate OPNET simulation model, with focus on the implementation of the GTS mechanism. The motivation that has driven this work is the validation of the Network Calculus based analytical model of the GTS mechanism that has been previously proposed and to compare the performance evaluation of the protocol as given by the two alternative approaches. Therefore, in this paper we contribute an accurate OPNET model for the IEEE 802.15.4 protocol. Additionally, and probably more importantly, based on the simulation model we propose a novel methodology to tune the protocol parameters such that a better performance of the protocol can be guaranteed, both concerning maximizing the throughput of the allocated GTS as well as concerning minimizing frame delay.**

Keywords - IEEE 802.15.4; GTS; OPNET Modeler; simulation model; analytical model

I. INTRODUCTION

The IEEE 802.15.4 [1] protocol has recently been adopted as a communication standard for low data rate, low power consumption and low cost Wireless Personal Area Networks. This protocol is quite flexible for a wide range of applications if appropriate tuning of its parameters is carried out. Importantly, the protocol also provides real-time guarantees by using the Guaranteed Time Slot (GTS) mechanism [2, 3]. Indeed, the GTS mechanism is quite attractive for time-sensitive Wireless Sensor Network (WSN) applications, particularly when supported by cluster-tree network topologies [4], such as defined in the ZigBee standard [5].

This paper addresses the performance evaluation of the IEEE 802.15.4 GTS mechanism, and it is a research effort aiming at assessing the IEEE 802.15.4/ZigBee protocols as candidate technologies within the ART-WiSe framework [6], which targets the design of a two-tiered architecture for large-scale critical WSN applications.

For this performance evaluation, we propose a simulation model for the IEEE 802.15.4 GTS mechanism within the implementation of the protocol under the OPNET simulator. This simulation model has been recently made available publicly by us in open source [7]. We then use this model to carry out a set of experiments. The obtained results in terms of performance evaluation allow us to prove the correctness of the previously proposed analytical model [3] that used Network Calculus formalism [8]: results previously obtained through Network Calculus upper bound or overpass the results obtained through simulation. The tighter results obtained through simulation allow us to propose a novel methodology to tune the protocol parameters such that a better performance of the protocol can be guaranteed, both concerning maximizing the throughput of the allocated GTS as well as concerning minimizing frame delay.

The rest of the paper is organized as follows. We start by elaborating on the limitations of existing simulation models of the IEEE 802.15.4 protocol in Section II. Then, in Section III, we describe some of the most relevant aspects of the protocol with a special emphasis on the GTS mechanism. In Section IV we briefly describe the proposed simulation model (additional details are available in [19]). In Section V, we address the GTS performance evaluation and compare the simulation results against the results obtained from the analytical model proposed in [3]. Based on the simulation results, a methodology for setting up the relevant protocol parameters is proposed. Finally, conclusions are drawn in Section VI.

II. AVAILABLE SIMULATION MODELS/TOOLS FOR IEEE 802.15.4

We rely on the OPNET Modeler [9] for developing our IEEE 802.15.4 simulation model (available for downloading [7]). OPNET Modeler was chosen due to its accuracy and to its sophisticated graphical user interface. While Network Simulator 2 (ns-2) [11] has been used to evaluate WSNs, the accuracy of its simulation results are questionable since the Medium Access Control (MAC) protocols, packet formats, and energy models are very different from those used in real WSNs [10, 12]. This basically results from the facts that ns-2 was originally developed for IP-based networks and only after extended for wireless ad-hoc networks.

The National Institute of Standards and Technology (NIST) has developed an OPNET simulation model for the

This work was partially funded by FCT under the CISTER Research Unit (FCT UI 608), by the PLURALITY project (CONCREEQ/900/2001), by the ARTIST2 NoE and by the Czech Republic MPO project (61 03001).

IEEE 802.15.4, profiled for healthcare applications [13]. However while that model implements the slotted and the unslotted CSMA/CA MAC protocols it does not support the GTS mechanism.

In [14], the authors have presented a comprehensive simulation study of the slotted CSMA/CA MAC protocol deployed by the IEEE 802.15.4 protocol in beacon-enabled mode, using an OPNET simulation model, which we now extend to include the GTS mechanism.

The performance evaluation of the IEEE 802.15.4 protocol was recently evaluated in another research work [15]. That work, however, approaches the IEEE 802.15.4 simulation model through the ns-2 simulator. It provides analysis of various features of the protocol, including experiments investigating the different characteristics of the direct, indirect and GTS data transmissions. Those results can not however be compared to those we provide in this current paper.

III. Overview of the IEEE 802.15.4 GTS Mechanism

The IEEE 802.15.4 [1] standard specifies the physical layer and the MAC sub-layer for Low-Rate Wireless Personal Area Networks (LR-WPANs). In this paper, we consider the physical layer operating in the 2.4 GHz frequency band, with 250 kbps of bit rate (referred to as *physical data rate* hereafter), which is supported, as an example by the MICAz motes [16] from Crossbow Tech.

The MAC sub-layer supports the beacon-enabled or non beacon-enabled operational modes that may be selected by a central controller of the Personal Area Network (PAN), called *PAN coordinator* (PANC). The media access is contention based (slotted or unslotted CSMA/CA); however, using the beacon-enabled mode, Guaranteed Time Slots can be allocated by the PANC exclusively to devices willing to transmit time critical data or data requiring specific bandwidth reservation.

In *beacon-enabled mode*, beacon frames are periodically sent by the PANC every *Beacon Interval* (BI) to identify its PAN, to synchronize devices that are associated with it, and to describe the superframe structure (Fig. 1), comprising an active period and, optionally, an inactive period. The active period, corresponding to the *Superframe Duration* (SD), is divided into 16 equally sized time slots, during which data transmission is allowed. Each active period can be further divided into a *Contention Access Period* (CAP) and an optional *Contention Free Period* (CFP), composed of GTSs. Slotted CSMA/CA is used within the CAP.

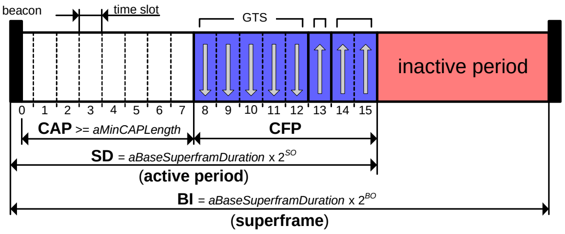


Figure 1. The IEEE 802.15.4 superframe structure

The structure of the superframe is defined by two parameters, the *Beacon Order* (BO) and the *Superframe Order* (SO), which determine the length of the superframe and its active period, respectively. The setting of *BO* and *SO* must satisfy the relationship $0 \leq SO \leq BO \leq 14$. The length of the superframe (BI) and the length of its active period (SD) are then defined as follows:

$$BI = aBaseSuperframeDuration \times 2^{BO}, \tag{1}$$

$$SD = aBaseSuperframeDuration \times 2^{SO}. \tag{2}$$

The *aBaseSuperframeDuration* constant denotes the minimum length of the superframe when *BO* is equal to 0. The standard fixes this duration to 960 symbols (one symbol corresponds to 4 bits, assuming the 2.4 GHz frequency band and 250 kbps of bit rate).

GTSs are always allocated by the PANC, either as a result of its own initiative or upon request from an *End Device* (or just *device*). Upon receiving a GTS allocation request, the PANC checks whether there are sufficient resources and, if possible, allocates the requested GTS. Each superframe supports up to 7 GTSs and each of those may contain one or more time slots. A GTS can only be used for messages from the device to the PANC (*transmit direction*) or from the PANC to the device (*receive direction*). Each device may request up to one GTS in the transmit direction and/or one GTS in the receive direction. The allocation of the GTS cannot reduce the length of the CAP to less than *aMinCAPLength* (440 symbols). Note that a device to which a GTS has been allocated can also transmit during the CAP. During the optional inactive period, each device may enter into a low-power mode to save energy.

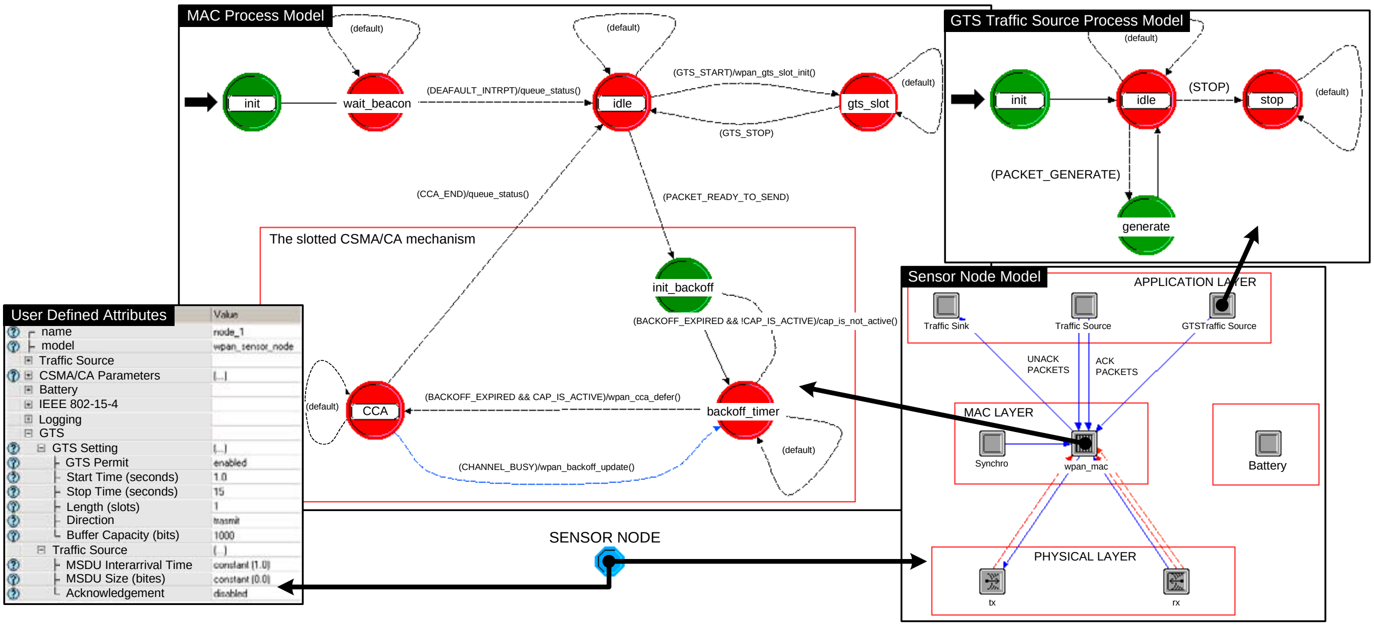
The *star* and *peer-to-peer* topologies are the two basic network topologies defined in the IEEE 802.15.4 standard. In the star topology, the communication is centralized and established between a PANC and its associated devices. The main advantage of this topology is its simplicity. The peer-to-peer topology has also a PANC; however, it differs from the star topology in that any device can communicate with any other device within its radio range. The cluster-tree topology [4] is a special case of a peer-to-peer topology with a distributed synchronization mechanism.

IV. The Simulation Model

A. The Simulation Model Structure

The OPNET Modeler is an industry leading discrete-event network modeling and simulation environment. Our simulation model builds on the wireless module, an add-on that extends the functionality of the OPNET Modeler with accurate modeling, simulation and analysis of wireless networks. Currently, our simulation model only supports the star topology, therefore enabling single-hop communications between End Devices and the PAN Coordinator.

The structure of the simulation model is presented in Fig. 2, in which the GTS-related part is given emphasis.



User Defined Attributes	Value
name	node_1
model	wpan_sensor_node
Traffic Source	
CSMA/CA Parameters	(...)
Battery	
IEEE 802-15-4	
Logging	
GTS	
GTS Setting	(...)
GTS Permit	enabled
Start Time (seconds)	1.0
Stop Time (seconds)	15
Length (slots)	1
Direction	trasmit
Buffer Capacity (bits)	1000
Traffic Source	(...)
MSDU Interarrival Time	constant (1.0)
MSDU Size (bites)	constant (0.0)
Acknowledgement	disabled

Figure 2. The simulation model of the IEEE 802.15.4 sensor node

The model is composed by the following four functional blocks:

1. The *physical layer* consists of IEEE 802.15.4 compliant radio transmitter (*tx*) and receiver (*rx*), operating at the 2.4 GHz frequency band and with 250 kbps of bit rate. The transmission power is set to 1 mW and the modulation technique is Quadrature Phase Shift Keying (QPSK).

2. The *MAC sub-layer* implements the slotted CSMA/CA and GTS mechanisms (e.g. GTS allocation, deallocation and reallocation). The GTS data traffic incoming from the application layer is stored in a buffer with a specified capacity and dispatched to the network when the corresponding GTS is active. This module also manages the generation of beacon frames, when a node acts as PANC.

3. The *application layer* consists of two data traffic generators - *Traffic Source* and *GTS Traffic Source* - and one sink. The *Traffic Source* generates unacknowledged and acknowledged data frames during the CAP, using slotted CSMA/CA (not used in this paper). The *GTS Traffic Source* can produce unacknowledged or acknowledged time critical data frames using the GTS mechanism. The *Traffic Sink* process module receives frames forwarded from lower layers and performs network statistics.

4. The *battery module* computes the consumed and remaining energy levels. The default values of the current draws are set to those of the MICAz mote [16].

We use and configure the default wireless modules of the OPNET library for emulating the physical characteristics of the radio channel such as background noise, propagation delay, radio interferences, received power and bit error rate.

B. *The User Defined Attributes*

This section depicts some important user-defined attributes of our GTS simulation model. The PANC may accept or reject the GTS allocation request from an End Device according to the value of the user-defined attribute *GTS Permit*. The device can specify the time when the GTS allocation and deallocation requests are sent to the PANC (*Start Time* and *Stop Time* attributes). The allocation request also includes the number of required time slots – *GTS Length* attribute – and direction (transmit or receive) – *GTS Direction* attribute.

When the requested GTS is assigned to a given device, its application layer starts to generate data blocks (hereafter called *frame payload*) that correspond to the MAC frame payload (i.e. MAC Service Data Unit (MSDU) [1]). On the other hand, the analytical model proposed in [3] considers data generated in a continuous bit stream. The size of the frame payload is specified by the probability distribution function of the *MSDU Size* attribute (see Fig. 3). The probability distribution function, specified in the *MSDU Interarrival Time* attribute, defines the inter-arrival time between two consecutive frame payloads. Then, the frame payload is wrapped in the MAC header and stored as a frame in the buffer with a given capacity (*Buffer Capacity* attribute).

The default size of the MAC header is 104 bits, since only 16-bit short addresses are used for communication (according to standard specification in [1]). The maximum allowed size of the overall frame (i.e. frame payload plus the MAC header) is equal to *aMaxPHYPacketSize* (1016) bits ([1]). The generated frames exceeding the buffer capacity are dropped. When the requested GTS is active, the frames are removed from the buffer, wrapped in the PHY headers, and dispatched to the network with an *outgoing data rate* equal to physical data rate *WPAN_DATA_RATE* (250 kbps).

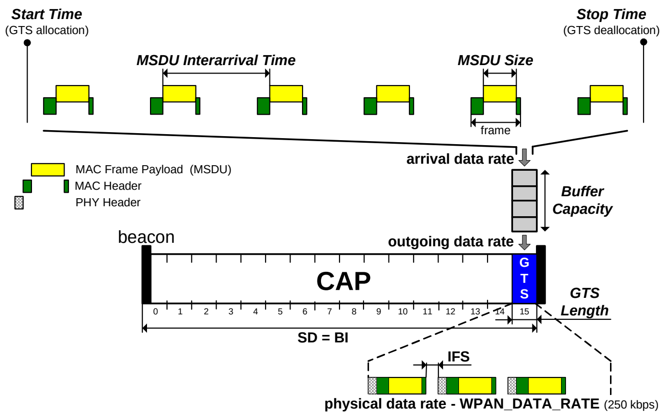


Figure 3. The behavior of the simulation model and its user defined attributes

C. The Simulation Setup

In our experiments, we consider a star-based IEEE 802.15.4 network with a PANC and one associated device within its radio coverage. This configuration is sufficient for the performance evaluation of the GTS mechanism, since there is no medium access contention. Thus, having additional devices would have no influence on the simulation results.

For the sake of simplicity, and without loss of generality, we assume the allocation of only one time slot GTS in each superframe and a 100% duty cycle ($SO = BO$). In what follows, the change of the superframe order means that the beacon order also changes while satisfying $SO = BO$.

The acknowledged and unacknowledged frames can be transmitted during the GTS. In this paper, we consider only unacknowledged transmissions for comparative purposes with the analytical results obtained in [3].

Consecutive frames are separated by *Inter-Frame Spacing* (IFS) periods. The IFS is equal to a *Short Inter-Frame Spacing* (SIFS) of 48 bits, for frame lengths smaller than *aMaxSIFSFrameSize* (i.e. 144 bits) [1]. Otherwise, the IFS is equal to a *Long Inter-Frame Spacing* (LIFS) of 160 bits, for frame lengths greater than *aMaxSIFSFrameSize* bits and smaller than *aMaxPHYPacketSize* (1016 bits) [1]. Note that a device that has allocated a GTS can only transmit a frame if the whole transmission (including the frame, the IFS and the acknowledgment - if requested) can be completed before the end of the GTS. Otherwise, it must wait until the next GTS.

The statistical data (e.g. average, maximum, minimum delays) are computed from a set of 1000 samples. Hence, the simulation time of one run is equal to the duration of 1000 superframe periods, and consequently the simulation time depends on the superframe order.

D. Simulation vs. Analytical Models

In Section V, we evaluate the performance of the GTS mechanism based in our IEEE 802.15.4 OPNET simulation model. The performance is compared to the analytical results of the GTS mechanism model derived in [3], which is based on the Network Calculus formalism. Network Calculus is a deterministic model for analyzing the performance guarantees in communication networks.

This analytical model relies on the (*b,r*) *model* as a linear arrival curve [8] for the GTS traffic generated by the sensor nodes. This means that each generated application data flow has a cumulative arrival function *R(t)* upper bounded by the linear arrival curve $\alpha(t) = b + r \times t$, where b denotes the maximum burst size and r denotes the average arrival rate. The analytical model is bit-oriented, which means that the application data are generated as a continuous bit stream with data rate r. On the other side, the simulation model has a more realistic frame-oriented basis, where the frame payload with a specified size is generated in a given time period (*MSDU Size* and *MSDU Interarrival Time* attributes - refer to Fig. 3). Consequently, the burst size b and arrival rate r, as defined in the (network calculus based) analytical model, should be implemented in the simulation model in the following way. A FIFO buffer with a specified capacity substitutes a data burst with a given size, and the *arrival data rate* is defined as follows:

$$r\,[\text{bps}] \equiv \frac{MSDU\ Size}{MSDU\ Interarrival\ Time}\,[\text{bps}] \qquad (3)$$

The smallest data unit in the analytical model is a bit while in the simulation model it is a frame with a bounded size. The data traffic of the OPNET simulation model depends on the Superframe Order (*SO*), capacity of the buffer (*Buffer Capacity*), time between two consecutive frames (*MSDU Interarrival Time*) and frame payload size (*MSDU Size*).

V. Simulation Results

In this section, we show how the superframe order, the arrival data rate, the buffer capacity and the size of the frame payload impact the data throughput of the allocated GTS and the delay of the transmitted GTS frames.

A. Impact of the Superframe Order on the GTS Throughput

Throughput as a function of the arrival data rate

The purpose of this section is to evaluate and compare the data throughput during one time slot GTS, for different values of the Superframe Order (*SO*) and for different arrival rates. For a given *SO*, the data throughput is related to the time effectively used for data transmission inside the GTS. Since the frames are transmitted without acknowledgments, the wasted bandwidth can only result from IFS or waiting for an empty buffer, as depicted in Fig. 4.

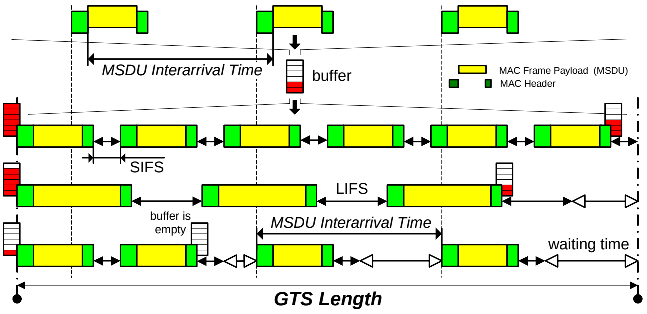


Figure 4. The utilization of the transmission time inside the GTS

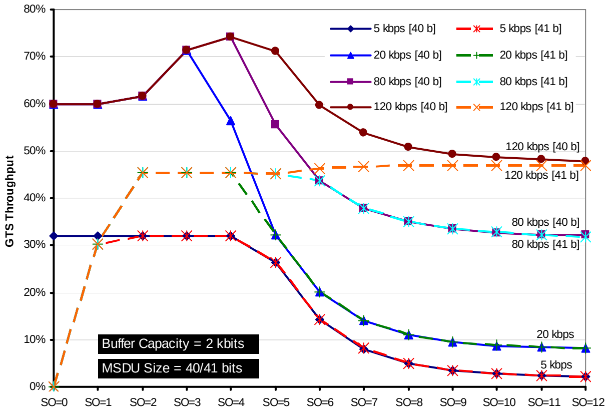


Figure 5. The throughput of the GTS as a function of the arrival data rate

The frames can be transmitted at the physical data rate (250 kbps) if the buffer does not become empty before the end of GTS (Fig. 4.a, b). Otherwise, if the buffer becomes empty, the frames are not stored in the buffer but they are directly dispatched to the network according to their arrival data rate (3), which is often lower than the physical data rate (Fig 4.c).

Fig. 5 plots the average data throughput of one allocated GTS for different superframe orders (with a duty cycle equal to 1) as a function of the arrival data rate, for two sizes of frame payload (40 and 41 bits). To identify the impact of the arrival data rate on the throughput, the buffer capacity is fixed to 2 kbits.

To show the impact of the IFS on the GTS throughput, the size of the frame payload is set to 40 and 41 bits. When the size of the frame payload is smaller or equal to 40 bits (frame size = 144 bits), the SIFS (48 bits) is used. Otherwise, if the frame payload size is greater or equal to 41 bits, then the LIFS (160 bits) is used. Note that, one additional bit in the frame payload causes 112 additional bits in the IFS. It can be easily observed in Fig. 5 that the impact of the IFS on the wasted bandwidth is more significant for low *SO* values.

When the size of the frame payload (*MSDU Size*) is fixed, the inter-arrival time (*MSDU Interarrival Time*) has to be changed according to (3) in order to reach the required arrival data rates. For instance, to achieve 5 kbps arrival data rate, the frame payload with 40 bits size has to be generated every 0.0288 s. We use the same settings as in the analytical model [3] and the *MSDU Size* and *MSDU Interarrival Time* attributes have been configured as constant values that correspond to the required data rates during each simulation run.

The behavior of the throughput for low *SO* values and the lowest arrival data rate (5 kbps) is quite different from the rest of the experiments. This occurs since the duration of one superframe for *SO = 0* is equal to 15.36 ms, but for 5 kbps arrival data rate the frame payload is generated every 28.8 ms. Thus, in every two superframes, one of them has no available frame in the buffer, and therefore the throughput is roughly the half of the ones resulting from other arrival data rates, where at least one frame is available in the buffer every superframe.

If the size of the frame payload is equal to 41 bits, it results that for *SO = 0* the throughput is zero for all arrival data rates, since the transmission of the frame (frame length plus LIFS) cannot be completed before the end of the GTS.

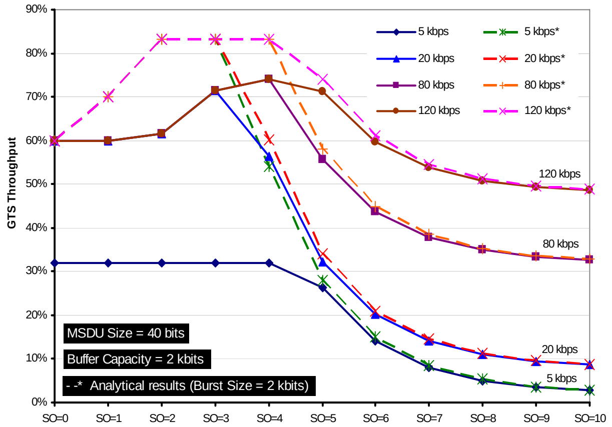


Figure 6. The throughput of the GTS as a function of the arrival data rate: simulation vs. analytical model. Analytical results drawn with dashed lines.

For low *SO* values, the throughput grows since the buffer does not become empty during a GTS duration (Fig. 4.a, b). On the other hand, the throughput for high *SO* values falls, since the buffer becomes empty before the end of the GTS (Fig. 4.c). For a large GTS, a significant amount of bandwidth is wasted when waiting for the incoming frame payload from the application layer. The throughput for high *SO* increases with the arrival data rate (i.e. lower *MSDU Interarrival Time*). It can be easily observed that the throughput performance for high *SO* values is identical and independent of the size of the frame payload.

Analytical results versus simulation results. In Fig. 6, the analytical and the simulation models have a very similar behavior in terms of the GTS throughput as a function of the arrival data rate. The throughput performance for high *SO* values has identical values and shape for both models. The simulation results are influenced by the frame-oriented approach of the simulation model, which is more significant for low *SO* values. The analytical model is bit-oriented, therefore it saturates available transmission bandwidth and therefore the throughput performance of the simulation model is upper-bounded by the maximum throughput of the analytical model (analytical results in dashed lines).

Throughput as a function of the buffer capacity

Fig. 7 plots the GTS throughput as a function of the buffer capacity. It can be observed that the throughput increases with the buffer capacity. The highest utilization of the GTS is achieved for *SO* between 2 to 5.

For the lowest *SO* values, the throughput depends neither on the arrival data rate nor on the buffer capacity, since the number of incoming frames during a superframe is low but still sufficient for saturating the GTS. For the higher *SO* values, the throughput does not depend on the buffer capacity and the throughput values grow with the arrival data rate. This occurs since the buffer becomes empty at the beginning of a large GTS and then, the generated frames are directly forwarded to the network with the rate equal to the arrival data rate.

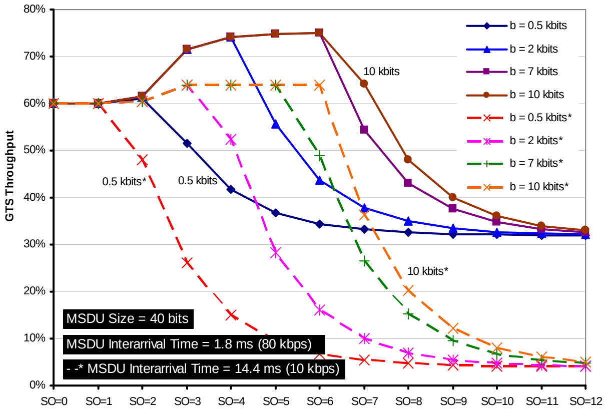


Figure 7. GTS throughput as a function of the buffer capacity.

Analytical results versus simulation results. In Fig. 8, the behaviors of the analytical and simulation models are very similar in terms of the GTS throughput as a function of the burst size/buffer capacity. The analytical results are obtained for the arrival rate equal to 5 kbps. We cannot use the same arrival rate for the simulation, because the lowest data rate 5 kbps has a specific behavior in case of the simulation model (Fig. 5). According the Fig. 5 we select an arrival data rate of 10 kbps as the closest one. The throughput performance for high *SO* values has identical values for both models, but for low *SO* values the simulation results are influenced by the frame vs. bit-oriented approach of the simulation and analytical model, as reported for the results given in Fig. 6. The analytical results upper bound the simulation results.

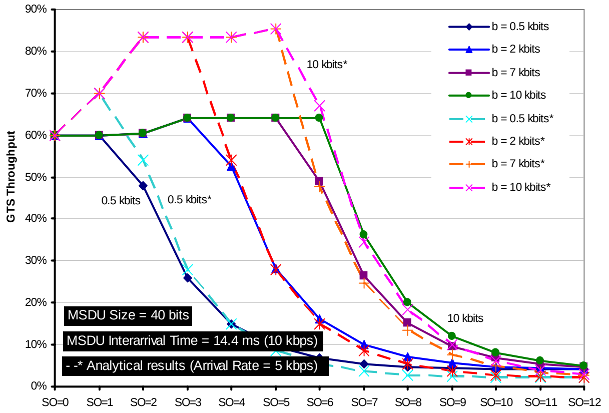


Figure 8. The throughput of the GTS as a function of the buffer capacity/burst size: simulation vs. analytical model. Analytical results in dashed lines.

The first conclusion concerning the GTS throughput for low arrival data rates and low buffer capacities is that high *SO* values are not suitable for ensuring efficient usage of the GTS in terms of data throughput. The maximum utilization of the allocated GTS is achieved with low superframe orders (3-4). The wasted GTS bandwidth increases with *SO*. To avoid this underutilization of the shared wireless medium, the i-GAME mechanism presented in [18] can be used.

B. Impact of the Superframe Order on the Delay Bound

In time sensitive applications, it is necessary to determine the frame delay bounds. In what follows, we present the impact of the *SO* values on the delay bound of the GTS frames for 100% duty cycle. We also determine the most suitable *SO* values for providing the lowest delay bound. Note that the delay is defined as the time duration between the instant when the frame is generated at the application layer and the instant when the frame is transmitted to the network. We consider two initial states for the buffer: empty or full.

Fig. 9 presents the frame delay bound as a function of the arrival data rate (i.e. frame payload inter-arrival time), for a frame size and a buffer capacity equal to 40 bits and 4 kbits, respectively. Observe that the delay bound depends neither on the arrival data rate nor on the initial size of the buffer for higher values of arrival data rate (20-120 kbps). In this case the behavior is almost identical to each other and the lowest delay bound is achieved for *SO* values equal to 2-3. This occurs since for low *SO* values ($SO < 5$) the maximum delay is achieved for full buffer. For increased values of the arrival data rate, only the time for filling the buffer grows. This explains also the identical behavior for initially full or empty buffer. For *SO* values higher or equal to 5, all frames stored in the buffer (with capacity equal to 4 kbits) can be transmitted during one GTS and the delay bound grows with *SO*. The value of this breakpoint depends on the buffer capacity (Fig. 10).

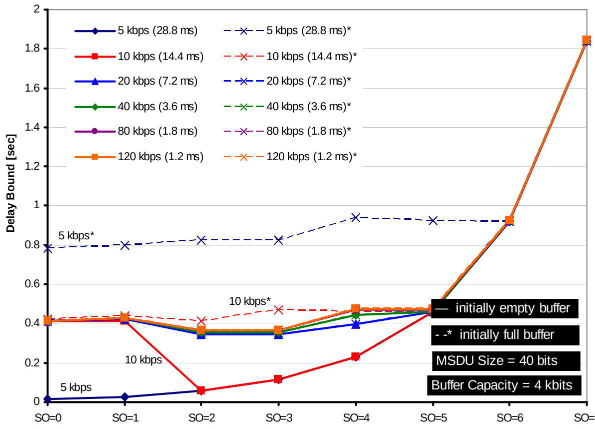


Figure 9. The frame delay bound as a function of the arrival data rate.

The delay bound behavior for the lowest arrival data rate is a monotonic function with the minimum for $SO = 0$. The arrival data rate is too slow and the buffer becomes always empty during one GTS for all *SO* values. Thus, the value of delay bound grows with *SO* and does not depend on the buffer capacity. When the buffer is initially full, the maximum delay is achieved at the beginning, and then the buffer becomes gradually empty. For $SO \geq 6$, the buffer is filled up during one superframe duration and the behaviors of initially full and empty buffers are met.

The specific delay bound behavior, for an arrival data rate equal to 10 kbps, is explained in more detail, with the support of the results shown in Fig. 10.

Fig. 10 shows the frame delay bound as a function of the buffer capacity, for a frame payload size and an inter-arrival time equal to 40 bits and 14.4 ms (i.e. 10 kbps), respectively.

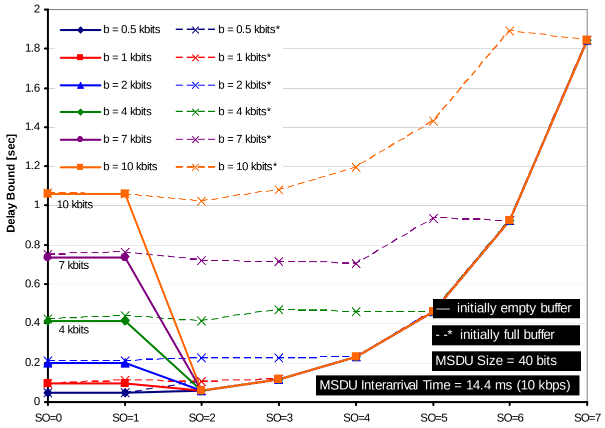


Figure 10. The frame delay bound as a function of the buffer capacity.

For the low *SO* values (0 and 1), the number of generated frames during one superframe is higher than the maximal number of potentially transmitted frames during the GTS so that the number of stored frames in the buffer grows. The maximum delay of the frame is reached when the buffer is full. Therefore, the frame delay bound depends only on the buffer capacity and grows with it. For increasing *SO* values, only the time when the buffer will be full grows. The delay bound values are roughly constant since when the superframe duration is doubled, (i.e. *SO* value is incremented by one) the GTS duration has to be doubled too. In what follows, the number of generated and transmitted frames is also doubled, thus their ratio stays constant.

When the buffer is initially empty and *SO* values are higher than 2, the frame delay bound depends only on the *SO* values instead of the buffer capacity and it is roughly equal to the superframe duration minus one GTS duration. This occurs since the number of generated frames is lower than the maximum number of potentially transmitted frames so that no frame is stored in the buffer between two consecutive superframes. When the buffer is initially full, the frame delay bound still depends on the buffer capacity until the value of *SO* causes that the full buffer becomes empty during one GTS. Afterwards, the delay bound depends only on the value of *SO*. The maximum delay is reached at the beginning, before the buffer becomes empty.

In this special case, for the lowest buffer capacity (0.5 kbits), the delay bound function is monotonic and grows with *SO* values, which makes superframe order zero the most suitable for providing the lowest delay bound. For higher buffer capacities, the most suitable value of *SO* in terms of the lowest delay bound is definitely 2 and does not depend on the buffer capacity, when the buffer is initially empty.

For the next experiments, we only consider initially empty buffer. The average and maximum delays (i.e. delay bound) as a function of the buffer capacity are compared in Fig. 11. The maximum delay is achieved at the beginning of each GTS for the first frame removed from the buffer. The following frames removed from the buffer during the GTS have lower delays than the first one, since the incoming data rate is often lower than the outgoing data rate. For low *SO* values, the number of transmitted frames during one GTS is also low and the average delay is then close to the maximum delay. For high *SO* values, the difference in delay between the first and last frames removed from the buffer during one GTS is greater, and the average delay is then further from the maximum delay.

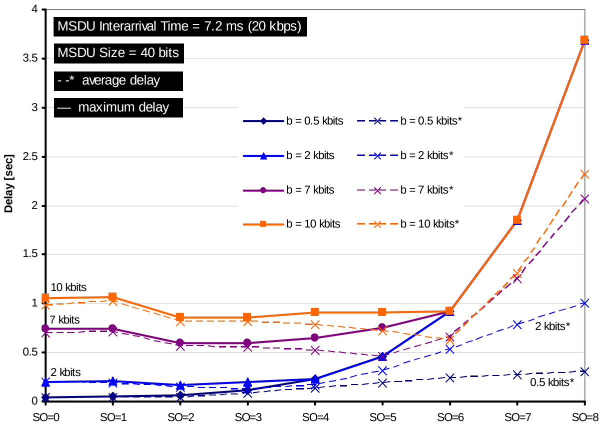


Figure 11. Average vs. maximum delay as a function of the buffer capacity.

Analytical results versus simulation results. The simulation and analytical results of the delay bound as a function of the buffer capacity or burst size are compared in Fig. 12. The analytical results are obtained for the arrival data rate equal to 5 kbps. We cannot use the same arrival data rate for the simulation model, because the delay bound for an arrival data rate equal to 5 kbps does not depend on the buffer capacity (Fig. 9). According to the results shown in Fig. 9, we select an arrival data rate equal to 20 kbps as the closest one. Hence, we cannot compare the values, but only the behavior of the models in terms of delay bound. This behavior is roughly similar for both models, and the lowest delay bound is achieved for $SO = 2$ for the case of higher buffer capacity (2-10 kbits), or for $SO = 0$ in case of lower buffer capacity (0.5 and 1 kbits). The difference between frame and bit-oriented approaches of the simulation and analytical models can be observed for the higher *SO* values. In the case of the analytical model, the delay bound curves converge slowly into a single one.

C. Setting SO for Time-Sensitive Applications

In summary, WSN applications with low data rates and low buffer capacities achieve the lowest delay bound for $SO = 0$. However, for higher buffer capacities (more than 1 kbits) and higher arrival data rates (more than 10 kbps) the most suitable value of *SO* for providing real-time guarantees is 2. The simulation and analytical results are roughly identical in terms of the delay bound, and the simulation results are upper-bounded by the analytical results.

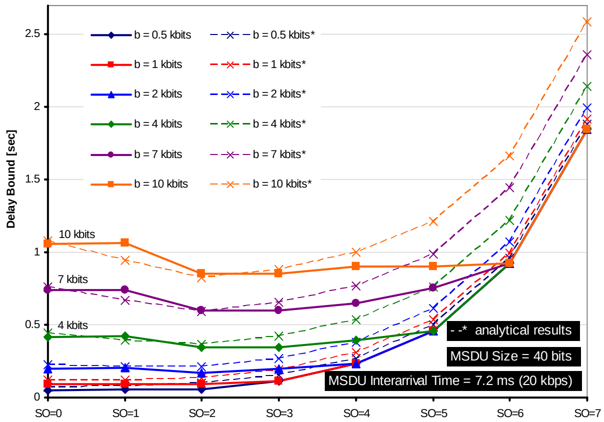


Figure 12. The delay bound of the GTS frame as a function of the buffer capacity/burst size: simulation vs. analytical model.

VI. Conclusions

In this paper, we briefly describe an OPNET simulation model of the IEEE 802.15.4 Guaranteed Time Slots (GTS) mechanism that we have added into a previously existing model of the protocol. This extended OPNET simulation model is made available publicly in open source [7].

We particularly focus on the performance evaluation of the GTS mechanism, comparing the obtained simulation results with the ones that were previously obtained [3] using an analytical model based on Network Calculus. The behaviors of both models are roughly identical in terms of the GTS data throughput and the delay bound; the analytical results upper bound the simulation results. Discrepancies (most significant for low superframe orders) are mainly due to the impact of the bit-oriented and frame-oriented approaches used by the analytical and simulation models, respectively.

We have also proposed a methodology to tune the protocol parameters for obtaining maximum data throughput and minimum frame delay. For applications with low data arrival rates and low buffer capacities, the maximum utilization of the allocated GTS is achieved for low superframe orders (3-4). However, the superframe order equal to 2 is the most suitable value for providing real-time guarantees in time-sensitive WSNs, since it grants the minimum delay bound for the GTS frames. High superframe orders are not suitable for ensuring efficient usage of the GTS neither in terms of data throughput nor delay bound.

Reference [19] is an extended version of this paper providing more detailed simulation results and a more detailed description of the OPNET simulation model.

References

[1] IEEE 802.15.4 Standard-2003, “Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR WPANs)”, IEEE SA Standards Board, 2003.

[2] A. Koubaa, M. Alves, and E. Tovar, “Time-Sensitive IEEE 802.15.4 Protocol,” chapter of the book “Sensor Networks and Configurations: Fundamentals, Techniques, Platforms, and Experiments,” Springer-Verlag, Germany, Jan. 2007.

[3] A. Koubaa, M. Alves, and E. Tovar, “GTS Allocation Analysis in IEEE 802.15.4 for Real Time Wireless Sensor Networks,” Workshop on Parallel and Distributed Real Time Systems (WPDRTS’06), Apr. 2006.

[4] A. Koubaa, M. Alves, and E. Tovar, “Modeling and Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks,” Real-Time Systems Symposium (RTSS’06), Brazil, Dec. 2006.

[5] Zigbee-Alliance, “ZigBee Specification,” http://www.zigbee.org/

[6] The ART-WiSe framework, http://www.hurray.isep.ipp.pt/art-wise/

[7] IEEE 802.15.4 OPNET Simulation Model, http://www.open-zb.net/

[8] J-Y. Leboudec, and P. Thiran, “A Theory of Deterministic Queuing Systems for the Internet,” LNCS, Vol. 2050, May 2004.

[9] Opnet Tech. Inc., Opnet Modeler - ver. 11.5A, http://www.opnet.com/

[10] D. Curren, “A Survey of Simulation in Sensor Networks,” project report (CS580), University of Binghamton, 2005.

[11] The Network Simulator NS-2, http://www.isi.edu/nsnam/ns/

[12] A. Koubaa, M. Alves, B. Nefzi, and Y. Q. Song, “Improving the IEEE 802.15.4 Slotted CSMA/CA MAC for Time-Critical Events in Wireless Sensor Networks,” Workshop on Real Time Networks RTN’06, Jul. 2006.

[13] S. Khazzam, “IEEE 802.15.4 MAC Protocol Model (used in ZigBee low-rate WPAN),” OPNET contributed model, Nov 2005.

[14] A. Koubaa, M. Alves, and E.Tovar, “A Comprehensive Simulation Study of Slotted CSMA/CA for IEEE 802.15.4 Wireless Sensor Networks,” Workshop on Factory Communication Systems (WFCS’06), Torino (Italy), Jun. 2006.

[15] J. Zheng and M. J. Lee, “A Comprehensive Performance Study of IEEE 802.15.4,” Sensor Network Operations, IEEE Press, Wiley InterScience, Chapter 4, pp. 218-237, 2006.

[16] MICAz Datasheet, http://www.xbow.com/

[17] A. Koubaa, M. Alves, and E. Tovar, “IEEE 802.15.4 for Wireless Sensor Networks: A Technical Overview,” IPP-HURRAY Technical Report (TR-050702), Jul. 2005.

[18] A. Koubaa, M. Alves, and E. Tovar, “i-GAME: An Implicit GTS Allocation Mechanism in IEEE 802.15.4,” Euromicro Conference on Real-Time Systems (ECRTS’06), Jul. 2006.

[19] P. Jurcik, A. Koubaa, M. Alves, E. Tovar and Z. Hanzalek, “A Simulation Model for the IEEE 802.15.4 protocol: Delay/Throughput Evaluation of the GTS Mechanism,” IPP-HURRAY Technical Report (TR 070508), May 2007.