Simulation study of energy efficient scheduling for IEEE 802.15.4/ZigBee cluster-tree Wireless Sensor Networks with time-bounded data flows

Petr Jurčík
Department of Control Engineering, Faculty of Electrical Engineering
Czech Technical University in Prague, Czech Republic
CISTER/ISEP, Polytechnic Institute of Porto, Portugal
jurcikp@fel.cvut.cz

Zdeněk Hanzálek
Department of Control Engineering, Faculty of Electrical Engineering
Czech Technical University in Prague, Czech Republic
hanzalek@fel.cvut.cz

Abstract

The simulation analysis is important approach to developing and evaluating the systems in terms of development time and cost. This paper demonstrates the application of Time Division Cluster Scheduling (TDCS) tool for the configuration of IEEE 802.15.4/ZigBee beacon-enabled cluster-tree WSNs using the simulation analysis, as an illustrative example that confirms the practical applicability of the tool. The simulation study analyses how the number of retransmissions impacts the reliability of data transmission, the energy consumption of the nodes and the end-to-end communication delay, based on the simulation model that was implemented in the Opnet Modeler. The configuration parameters of the network are obtained directly from the TDCS tool. The simulation results show that the number of retransmissions impacts the reliability, the energy consumption and the end-to-end delay, in a way that improving the one may degrade the others.

1. Introduction

Wireless Sensor Networks (WSNs) emerge as enabling infrastructures for industrial monitoring and control systems [22]. Timeliness and energy efficiency are important requirements to be fulfilled in these systems because the transmission of real-time messages must respect given deadlines and the wireless nodes are usually energy-constrained. The interdependence of the reliability of data transmission, the energy consumption of the nodes and the end-to-end communication delay introduces additional complexity to the network design.

In this paper, we assume a static deployment of wireless nodes organized in the cluster-tree topology, where each node knows its parent router and child nodes (e.g using the ZigBee address assignment mechanism [24]). The network carries time-bounded data flows given by the parameters (such as sink node, source nodes, required period, end-to-end deadline) that must be known in network design/redesign time. We rely on cluster-tree topology because it supports predictable and energy efficient behavior, which is suited for time-sensitive applications using battery-powered nodes. On the other side, the cluster-tree topology requires a precise cluster scheduling to avoid inter-cluster collisions. Thus, the key problem is to find a periodic schedule, which specifies when the clusters are active while avoiding possible inter-cluster collisions, minimizing the energy consumption of the nodes and meeting all data flows' parameters. In [7], we have proposed a Time Division Cluster Scheduling (TDCS) tool solving this problem, which is NP-hard from the time complexity point of view. The TDCS tool enables system designers to easily configure all the required parameters of the IEEE 802.15.4/ZigBee beacon-enabled cluster-tree WSNs in the network design/redesign time.

This paper demonstrates the application of proposed TDCS tool using the simulation study, which analyzes the interdependence of reliability, energy consumption and timeliness in IEEE 802.15.4/ZigBee beacon-enabled cluster-tree WSNs. We show how to apply the TDCS tool to configure a given simulation scenario, as an illustrative example that confirms the practical applicability of this tool. The paper also contributes with the extended simulation model of IEEE 802.15.4/ZigBee protocols that was implemented in the Opnet Modeler simulator [19] and is used to carry out a set of experiments.

This work was partially funded by the Ministry of Education of the Czech Republic under the projects ME10039, P103/10/0850 and Research Program MSM6840770038, by the Portuguese Science Foundation under the CISTER Research Unit (FCT UI 608), by the ArtistDesign and CONET European Networks of Excellence and by the EMMON European project.

The rest of the paper is organized as follows. The system model assumed within the simulation study is presented in Section 2. In Section 3, some of the most relevant aspects of the IEEE 802.15.4/ZigBee protocols are addressed. Sections 4 and 5 provide brief description of the proposed TDCS tool and IEEE 802.15.4/ZigBee Opnet simulation model, respectively. Section 6 describes the simulation scenario and presents the simulation results. Finally, conclusions are drawn in Section 7.

1.1. Related work

Recently, several analytical and simulation models of the IEEE 802.15.4 protocol have been proposed. Nevertheless, currently available simulation models [20] for this protocol are both inaccurate and incomplete, and in particular they do not support the Guaranteed Time Slot (GTS) mechanism, which is required for time-sensitive WSN applications.

Opnet Modeler, ns-2 and OMNeT++ are widely used and popular network simulators, which include a simulation model of the IEEE 802.15.4 protocol. The 802.15.4/ZigBee simulation model in Opnet model library [19] supports only non beacon-enabled mode, therefore, the cluster-tree topology and GTS mechanism cannot be simulated. In addition, the source codes of the network and application layers are not available. The National Institute of Standards and Technology (NIST) has developed own Opnet simulation model for the IEEE 802.15.4 protocol [14]. However, while that model implements the slotted and the unslotted CSMA/CA MAC protocols it does not support the GTS mechanism as well. It also uses its own radio channel model rather than the accurate Opnet wireless library. The Network Simulator 2 (ns-2) [6] is an object-oriented discrete event simulator including a simulation model of the IEEE 802.15.4 protocol. 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 [16]. This basically results from the facts that ns-2 was originally developed for IP-based networks and further extended for wireless networks. Moreover, the GTS mechanism was not implemented in the ns-2 model. OMNeT++ (Objective Modular Network Testbed in C++) [21] is another discrete event network simulator supporting unslotted IEEE 802.15.4 CSMA/CA MAC protocol only. Finally, note that while ns-2 and OMNeT++ are open-source projects, the Opnet Modeler is commercial project providing a free of charge university program for academic research projects.

There have also been several research works on the performance evaluation of the IEEE 802.15.4 protocol using simulation model. Zheng et al. [23] have evaluated various features of the 802.15.4 protocol (e.g. direct, indirect and GTS data transmissions), and investigated the collision behavior of IEEE 802.15.4. In addition, the simulation experiments compare the performance of 802.15.4 and 802.11 protocols. The authors have developed own ns-2 simulation model of 802.15.4 protocol, which additionally implements beacon-enabled mode and GTS mechanism. Since the network layer has not been implemented, a star topology is only supported. Based on this implementation, Chen et al. [2] have developed own simulation model of IEEE 802.15.4 protocol in OMNeT++. Contrary to the standard OMNeT++ model, their simulation model implements a battery module, beacon-enabled mode and GTS mechanism, and supports only star topology. Using this simulation model, the IEEE 802.15.4 star network has been evaluated in terms of energy consumption and end-to-end communication performance in [3]. Hurtado-Lopez et al. [8] have extended the above mentioned IEEE 802.15.4 model in OMNeT++ to support cluster-tree topology.

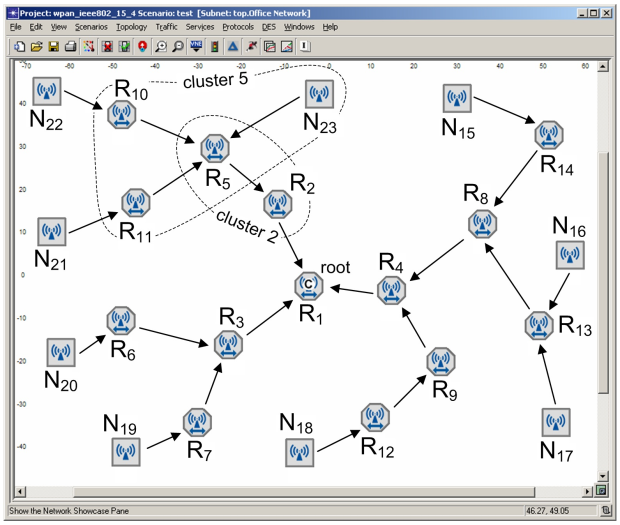


Figure 1. The simulation scenario in Opnet Modeler (parent-child relationships).

In this work, the proposed TDCS tool is evaluated using own IEEE 802.15.4/ZigBee simulation model (Section 5) that was implemented in Opnet Modeler simulator [19]. The performance analyses of the slotted CSMA/CA mechanism and GTS mechanism of IEEE 802.15.4 protocol in beacon-enabled mode have been presented in [15] and [13], respectively, using the former version of IEEE 802.15.4 simulation model, which we now extended to include the ZigBee network layer enabling a simulation study of the cluster-tree WSNs.

2. System model

We consider a static deployment of wireless nodes organized in a cluster-tree topology. The hierarchy of the cluster-tree topology is defined by the parent-child relationships forming a directed tree, called in-tree [5], in the sense that each solid arrow in Fig. 1 leaves the *child* node and enters the *parent* node. Note that the in-tree has the following property: one node, called *root*, has no parent and any other node has exactly one parent.

The routers and end-nodes are two types of wireless nodes in cluster-tree WSNs. The nodes that can participate in multi-hop routing are referred to as *routers* (R_i). The nodes that do not allow association of other nodes and do not participate in routing are referred to as *end-nodes* (N_i). In the cluster-tree topology, the nodes are organized in logical groups, called *clusters*. Each router forms a cluster and is referred to as its *cluster-head* (e.g. router R_5 is the cluster-head of cluster 5). All of its child nodes (e.g. end-node N_{23} and routers R_{10} and R_{11} are child nodes of router R_5) are associated to the cluster, and the cluster-head handles all their transmissions.

In cluster-tree topology, the multi-hop communication is deterministic because each node only interacts with its pre-defined parent router and child nodes. Messages are forwarded from cluster to cluster until reaching a sink. The time behavior of each cluster is periodic and the period of each cluster is divided into two portions. *Active portion*, when the cluster-head enables the data transmissions inside its cluster, and subsequent *inactive portion*. Each router (except the root) belongs to two clusters, once as a child node and once as a cluster-head. For example in Fig. 1, router R_5 acts as a cluster-head in cluster 5 and as a child node in cluster 2. Thus, each router must be awake whenever one of these two clusters is active, otherwise it may enter a low power mode to save energy.

The traffic is organized in the multi-source mono-sink flows (see user-defined parameters of the flows from the simulation scenario summarized in Table 1), which must be known in network design time. Each flow has one or more sources and exactly one sink. In this paper, we assume that both routers and end-nodes can have sensing or/and actuating capabilities, hence, they can be sources or/and sinks of flows. A node regularly measures a sensed value (e.g. temperature, pressure) with the required period, called the *req_period*, and reports the acquired sensory data of a given size, called the *sample_size*, to a sink. Note that req_period defines the minimal inter-arrival time between two consecutive measurements, and a particular inter-arrival time has to be equal to or greater than req_period.

End-to-end (e2e) delay d_{ij}, given as a time between the instant when a source i sends the message and the instant when the sink j receives this message, is bounded by $e2e\_deadline_{ij}$ such that $d_{ij} \leq e2e\_deadline_{ij}$. Note that this parameter is set for each source of a particular data flow, and all of them must be met.

A *collision domain* of a cluster is a set of clusters, which compete for the same radio channel and, therefore, their active portions must be non-overlapping, i.e. only one cluster from a collision domain can be active at a given time instant. It is easy to see that in a network with multiple collision domains, the clusters from different non-overlapping collision domains may be active at the same time (i.e. some clusters' active portions can run simultaneously).

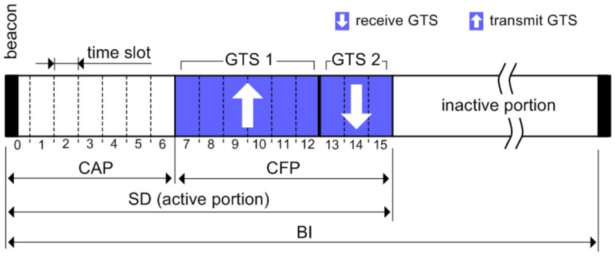


Figure 2. Superframe structure.

3. Overview of IEEE 802.15.4/ZIGBEE

The IEEE 802.15.4/ZigBee [9, 24] stand as the leading communication technologies for low cost, low power and low data rate WSNs. The IEEE 802.15.4 [9] standard specifies the physical and data link layers, while the network and application layers are defined by the ZigBee specification [24]. The Medium Access Control (MAC) layer supports the beacon-enabled or non beacon-enabled modes that may be selected by a central controller of the WSN, called the *PAN coordinator*. In this paper, we only consider the beacon-enabled mode, since it supports cluster-tree topology and enables the energy conservation using low duty-cycles. In addition, the beacon-enabled mode also offers some real-time guarantees by means of the *Guaranteed Time Slot* (GTS) mechanism, which ensures the collision-free and predictable access to the shared wireless medium. Thus, when the timeliness and energy efficiency are the main concerns, the beacon-enabled mode should be employed.

While IEEE 802.15.4 in the beacon-enabled mode supports only the star-based topology, the ZigBee specification has proposed its extension to the cluster-tree topology. In the particular case of ZigBee cluster-tree WSNs, a PAN coordinator is identified as the root of the tree and forms the initial cluster. The other routers join the cluster-tree in turn by establishing themselves as cluster-heads, starting to generate the beacon frames for their own clusters. Beacon frames are periodically sent at *Beacon Interval* (BI) to synchronize the child nodes that are associated with a given cluster-head and to define a superframe structure (Fig. 2).

Each cluster's period, corresponding to BI, is divided into an active and an inactive portions. During the inactive portion, each associated node may enter a low power mode to save energy. The active portion, corresponding to *Superframe Duration* (SD), is divided into 16 equally-sized time slots, during which the data transmission is allowed. These time slots are further grouped into a *Contention Access Period* (CAP) using slotted CSMA/CA for the best-effort data delivery, and an optional *Contention Free Period* (CFP) supporting the time-bounded data delivery. Within the CFP, the cluster-head can allocate Guaranteed Time Slots (GTSs) to its child nodes. The CFP supports up to 7 GTSs and each GTS may contain one or more time slots. Each child node may request up to one GTS in the *transmit direction*, i.e. from the child node to the parent router, and/or one GTS in the *receive direction*,

flow ID	sources	sink	$e2e\_deadline$ [sec]	req_period [sec]	$sample\_size$ [bit]
1	$\{N_{19}, N_{21}, N_{23}\}$	N_{15}	2.6	2.1	64
2	$\{N_{17}, N_{18}\}$	N_{20}	0.8	1.4	32
3	$\{R_{12}, N_{16}, N_{20}\}$	N_{22}	3.4	1	48

Table 1. The user-defined parameters of the data flows from the simulation scenario.

i.e. from the parent router to the child node. Note that a node to which a GTS has been allocated can still transmit the best-effort data within the CAP.

Durations of the cluster's period (BI) and the cluster's active portion (SD) are defined by two parameters, the *Beacon Order* (BO) and the *Superframe Order* (SO) as follows:

$$\begin{aligned} BI = aBaseSuperframeDuration \cdot 2^{BO} \\ SD = aBaseSuperframeDuration \cdot 2^{SO} \end{aligned} \quad (1)$$

where $0 \leq SO \leq BO \leq 14$ and *aBaseSuperframeDuration* = 15.36 ms (assuming the 2.4 GHz frequency band and 250 kbps of data rate) and denotes the minimum duration of active portion when $SO = 0$. Note that the ratio of the active portion (SD) to the whole period (BI) is called the *duty-cycle*.

Remind that due to the cluster-tree topology, each router (except the root) belongs to two clusters, once as a child node and once as a cluster-head. Hence, router r has to maintain the timing between the active portion (SD) of its parent's cluster (in which a beacon and the data frames from the parent router are received, and the data frames to the parent router are sent) and its own active portion (in which a beacon and the data frames are sent to the associated child nodes, and the data frames from child nodes are received). Router r acts as a child node in the former active portion, while in the latter active portion it acts as a cluster-head. The relative timing of these active portions is defined by the *StartTime* parameter [9].

4. Scheduling tool

This section provides a brief explanation of the energy efficient Time Division Cluster Scheduling (TDCS) mechanism that we have implemented in Matlab [10] using the GLPK solver (GNU Linear Programming Kit by A. Makhorin). For additional details please refer to [7].

In cluster-tree WSNs, the flows traverse different clusters on their routing paths from the source nodes to the sink nodes. The clusters may have collisions when they are in the neighborhood. Thus, to avoid inter-cluster collisions (beacon/data frames transmitted from nodes in different clusters), it is mandatory to schedule the clusters active portions (SDs) in an ordered sequence, that we call the *Time Division Cluster Schedule* (TDCS). The fact that a cluster is active only once during its period and the flows may have opposite directions leads to cyclic behavior of periodic schedule. Hence, the TDCS is characterized not only by the moments when the clusters become active within the period, but due to the cyclic nature of the problem it is also characterized by the index of the period for each flow in a given cluster.

A number of TDCSs can be found for a cluster-tree WSN, but we are interested in a feasible TDCS ensuring that each data flow ”deterministically” meets its e2e deadlines. The key idea is to formulate the problem of finding a feasible TDCS as a cyclic extension of the Resource Constrained Project Scheduling with Temporal Constraints (RCPS/TC) problem [18] so that the users are not restricted to a particular implementation but they can make a similar extension to any of the algorithms solving this problem. The performance evaluation showed that the problems with dozens of tasks can be solved using an Integer Linear Programming (ILP) algorithm.

Since wireless nodes are usually battery-powered, the objective is to minimize the energy consumption of the nodes by maximizing the TDCS period, corresponding to BI, while avoiding possible inter-cluster collisions (i.e. resource requirements) and meeting all data flows' end-to-end deadlines (i.e. temporal requirements). Note that to minimize the energy consumption of nodes, the lowest duty-cycles must be chosen (IEEE 802.15.4 supports duty-cycles under 1%). All clusters have equal BI, defined by BO, but various SD, defined by SO, (i.e. various duty-cycle) to ensure efficient bandwidth utilization. The BI should be set as long as possible to minimize clusters' duty-cycle and, consequently, to minimize the energy consumption of the nodes. As a result, the cluster's inactive portion is extended, and the nodes may stay in the low power mode longer to save energy. On the other hand, low duty-cycles enlarge the end-to-end delays. Hence, energy consumption is in contrast to the fast response of a WSN, therefore we are interested in finding the TDCS minimizing the duty-cycles while respecting all of the required data flows' e2e deadlines.

Hence, the TDCS algorithm is called iteratively starting from the minimum BI up to the maximum BI. The maximum BI, given by BO_{max} in Eq. (1), is equal to or shorter than the shortest *req_period* among all of the data flows. The minimum BI, given by BO_{min}, is equal to or longer than the duration of all clusters' SDs when assuming that non-interfering clusters overlap. If a feasible TDCS is found for a given BI, BO is increased by 1 and the TDCS algorithm is called once again with new BI. This procedure is repeated until $BO = BO_{max}$ or a feasible TDCS is not found. Then, the last feasible TDCS meets all the resource and temporal requirements while minimizing the energy consumption of the nodes.

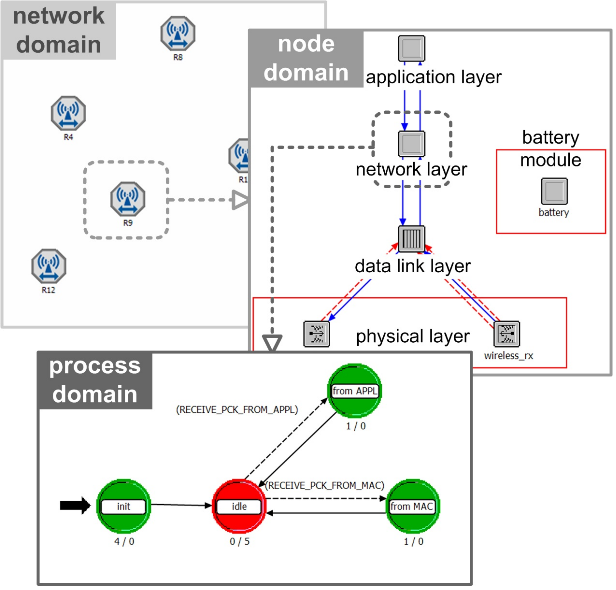


Figure 3. The structure of the IEEE 802.15.4/ZigBee simulation model.

Using our TDCS scheduling tool, we are able to configure the parameters of each cluster, such as BO, SO and $StartTime$, in IEEE 802.15.4/ZigBee beacon-enabled cluster-tree WSNs. Furthermore, for every cluster's superframe, the configuration parameters [9] of each allocated GTS such as *GTS device*, *GTS direction*, *GTS length* and *GTS starting slot* can be obtained as well.

5. Simulation model

This section presents the structure of the IEEE 802.15.4/ZigBee simulation model [12] that we have implemented in the Opnet Modeler simulator. For additional details please refer to [11].

The Opnet Modeler [19] is an industry leading discrete event network modeling and simulation environment. Opnet Modeler was chosen due to its accuracy and to its sophisticated graphical user interface. The development environment consists of three hierarchical modeling domains (Fig. 3). Network domain describes network topology in terms of nodes and links. Internal architecture of a node is described in the node domain. Within the process domain, the behavior of a node is defined using state transition diagrams. Operations performed in each state or transition are described in embedded C/C++ code blocks. The IEEE 802.15.4/ZigBee 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.

In accordance to the ZigBee [24] specification, there are three types of nodes implemented in the simulation model, namely a PAN coordinator, a router and an end device. All types of nodes have the same internal structure (node domain) but they differ in the available attributes.

The structure of the IEEE 802.15.4/ZigBee simulation model is presented in Fig. 3. The model implements the *physical layer* of the IEEE 802.15.4 [9] standard running at 2.4 GHz Frequency band with 250 kbps data rate. Default settings are used for the physical characteristics of the radio channel such as background noise and interference, propagation delay, antenna gain, and bit error rate.

The *data link layer* supports the beacon-enabled mode (non beacon-enabled mode is not supported yet) and implements two medium access control protocols according to the IEEE 802.15.4 standard, namely the contention-based slotted CSMA/CA and contention-free GTS. Data frame incoming from the network layer is wrapped in MAC header and MAC footer and stored to two separate FIFO buffers, namely a buffer for best-effort data frames and another buffer for real-time data frames. The frames are dispatched to the network when the corresponding CAP or CFP is active. On the other hand, the frame incoming from the physical layer is unwrapped and passed to the network layer for further processing. The data link layer also generates required commands (e.g. GTS allocation, deallocation and reallocation commands) and beacon frames when a node acts as PAN coordinator or router.

The *network layer* implements hierarchical routing protocol according to the ZigBee [24] specification. The frames are routed upward or downward along the cluster-tree topology according to the destination address by exploiting the hierarchical addressing scheme provided by ZigBee [24]. This addressing scheme is based on the symmetric hierarchical addressing tree.

The *application layer* can generate unacknowledged and/or acknowledged best-effort and/or real-time data frames transmitted during CAP or CFP, respectively. There is also a *battery module* that estimates the energy consumption using the formula $U \cdot I \cdot t$ based on the execution time (t), the voltage (U), and current draw (I). The default values of current draws are set to those of the TelosB [4] mote specification.

In [13], the GTS mechanism of this simulation model has been validated using the analytical model [17] based on the Network Calculus methodology.

6. Simulation study

It is unrealistic to support hard real-time communications in a WSN due to communication errors resulting from the unreliable and time-varying characteristics of wireless channels [1]. To increase the reliability of data transmission, the acknowledgment and retransmission mechanisms can be employed. Both mechanisms are natively supported by the IEEE 802.15.4 standard [9]. Note that the maximum number of retransmissions must be bounded, otherwise, the analysis will not be possible. Given a channel error rate, this simulation study shows how the maximum number of retransmissions (parameter $macMaxFrameRetries$ [9]) impacts the reliability of data transmission, the energy consumption of

cluster	BO	SO	StartTime	GTS device	GTS length	GTS direction	GTS starting slot
cluster 1	6	2	0.0	R_2	1	transmit	7
				R_3	1	transmit	8
				R_4	2	transmit	9
				R_2	2	receive	11
				R_3	1	receive	13
				R_4	2	receive	14
cluster 2	6	1	0.75168	R_5	2	transmit	11
				R_5	3	receive	13
cluster 3	6	0	0.65952	R_6	2	transmit	8
				R_7	2	transmit	10
				R_6	4	receive	12

Table 2. The configuration parameters of clusters 1, 2 and 3 obtained by the TDCS tool.

the nodes and the end-to-end communication delay, in a way that improving the one may degrade the others. The configuration parameters of each cluster are obtained directly from the TDCS tool [10]. Table 2 presents a part of the TDCS tool's output, namely the configuration parameters of clusters 1,2 and 3 from the simulation scenario (Fig. 1) assuming unacknowledged transmission (i.e. $macMaxFrameRetries = 0$).

6.1. Simulation scenario

The simulation scenario (illustrated in Fig. 1) consists of 14 clusters and 23 TelosB motes forming a cluster-tree WSN. The TelosB [4], which simulation model has been presented in Section 5, is a battery-powered wireless module widely used in WSNs. We consider a set of three time-bounded data flows with user-defined parameters summarized in Table 1.

New TDCS and configuration parameters of clusters, which ensure that each data flow meets its e2e deadline while minimizing the energy consumption of the nodes, are generated for each number of retransmissions from scratch. Without loss of generality, the non-overlapping TDCSs are assumed (i.e. a single collision domain), because the simulation model does not support the definition of the multiple collision domains. The simulation time of one run is equal to 20 minutes involving generation of 1707 frames in case of flow 1, 1706 frames in case of flow 2 and 3585 frames in case of flow 3.

In fact, to engineer applications with certain guarantees, we must have a certain confidence on the communication channel, and this can be done by empirically analyzing the channel error rate prior to a given deployment. For the sake of simplicity, the homogeneous channel error rate (a ratio of a number of dropped frames to a number of dispatched frames) equal to 20% is assumed. That means when a node receives a frame, the dropping probability is genereted as an uniformly distributed random number on the interval 0 to 100. If the dropping probability is less than 20, the received frame is dropped by a given node.

6.2. Simulation results

Figure 4 shows the reliability of data transmission as a function of the maximum number of retransmissions (parameter $macMaxFrameRetries$). For each flow, the reliability of data transmission is calculated as the ratio of the number of dispatched frames by all sources to the number of received frames by the sink. The average ratio of all flows is then plotted in the chart (see Fig. 4).

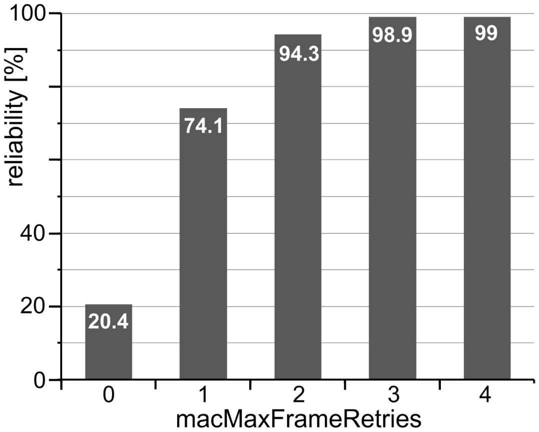


Figure 4. Reliability of data transmission.

Figure 5 shows the sum of energy consumption of all nodes within the simulation run as a function of the maximum number of retransmissions. As expected, the reliability and energy consumption grow with the number of retransmissions. It can be observed that the reliability of acknowledged transmission with the maximum of one retransmission ($macMaxFrameRetries = 1$) increases 3.6 times against the reliability of unacknowledged transmission ($macMaxFrameRetries = 0$). On the other side, the energy consumption increases only 1.52 times. Hence, a fair trade-off between reliability and energy efficiency must be found for a given application specific requirements.

The maximum end-to-end delays (d_{ij}) for each flow and each number of retransmissions are presented in

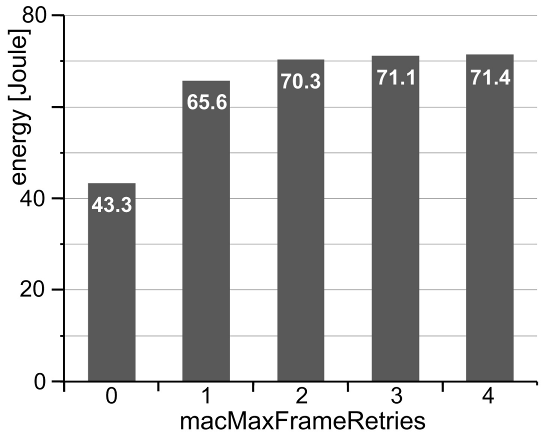
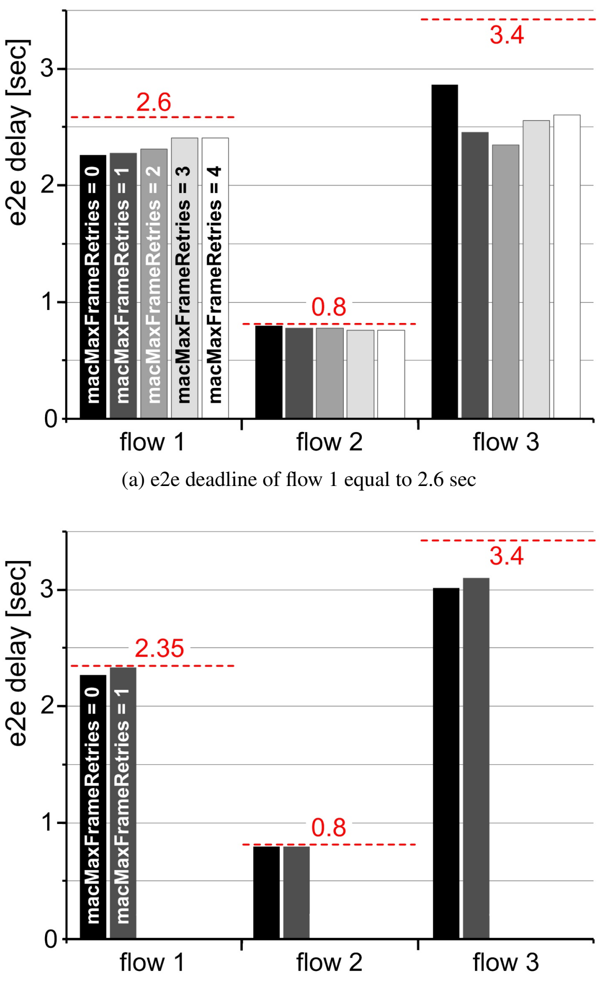


Figure 5. Sum of energy consumption of all nodes in the network.

Fig. 6. The dashed line at each column depicts the required end-to-end deadline ($e2e\_deadline$) for a given flow. A first observation confirms that all TDCSs are feasible, because the maximum end-to-end delays are shorter than end-to-end deadlines. However, the maximum e2e delays cannot be compared among each other, because new TDCS is generated from scratch for each number of retransmissions to meet required e2e deadlines. Note that the objective of TDCS tool is to meet the e2e deadlines but not to minimize the e2e delays. Hence, to study how the number of retransmissions impacts the e2e delay, we reduced e2e deadline of flow 1 to 2.35 seconds (the other parameters are kept the same). In this case, a feasible TDCS can be only found for $macMaxFrameRetries$ in the range of 0 to 1, as depicted in Fig. 6b. For $macMaxFrameRetries = 2$ and more, no feasible TDCS exists, because the maximum e2e delay of a flow is always greater than its e2e deadline. Compared with the former case (Fig. 6a) where a feasible TDCS can be found for $macMaxFrameRetries$ in the range of 0 to 4, we can implicitly deduce that e2e delay grows with the number of retransmissions. In the example of Fig. 6, it can be observed that the small reduction in e2e deadline (2.6 sec → 2.35 sec) causes that an acknowledged transmission with the maximum of one retransmission is only feasible, which results in a significant degradation of the reliability of data transmission (99% → 74.1%).



(a) e2e deadline of flow 1 equal to 2.6 sec

(b) e2e deadline of flow 1 reduced to 2.35 sec

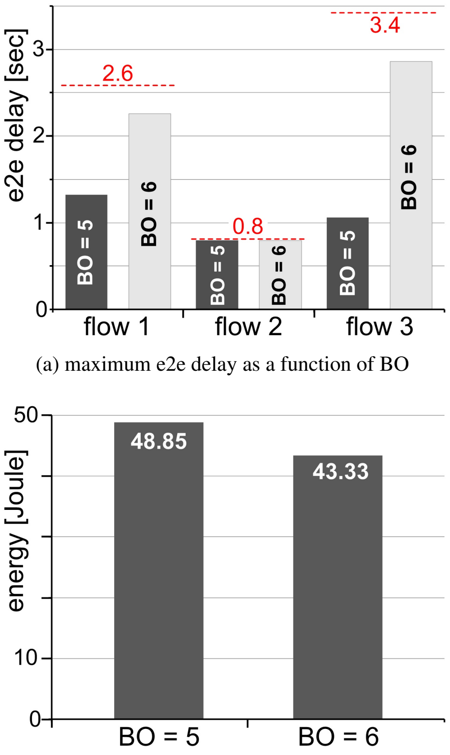
Figure 6. Maximum e2e delay as a function of the maximum number of retransmissions.

Finally, this section demonstrates how the length of the TDCS period, given by the Beacon Order (BO), impacts the energy consumption of the nodes. In case of unacknowledged transmission ($macMaxFrameRetries = 0$), there exists two feasible TDCSs. A shorter TDCS with the period given by $BO = 5$, and a longer TDCS with the period given by $BO = 6$. Figure 7a confirms that both TDCSs are feasible, because the maximum end-to-end delays are shorter than end-to-end deadlines (dashed line) in both cases. However, Figure 7b shows that the network nodes consume more energy when the shorter TDCS ($BO = 5$) is applied. Hence, according to our required objectives, the TDCS tool returns the longer TDCS that meets all e2e deadlines while minimizing the energy consumption (i.e. maximizing the lifetime of the nodes).

7. Conclusions

IEEE 802.15.4 standard [9] and Zigbee specification [24] admit the formation of the cluster-tree network but none of them imposes any algorithm or methodology to create and organize it. Using the proposed TDCS tool [7, 10] and Opnet simulation model [12], the paper demonstrates that system designers are able to easily configure and organize the IEEE 802.15.4/ZigBee beacon-enabled cluster-tree WSNs, and find the trade-off between reliability, energy consumption and timeliness for a given application specific requirements prior to the network deployment.

The interdependence of reliability, energy consumption and timeliness introduces additional complexity to the network design. Hence, the simulation results show



(a) maximum e2e delay as a function of BO

(b) energy consumption as a function of BO

Figure 7. The QoS metrics of two feasible TDCSs assuming unacknowledged transmission.

that providing higher reliability while increasing the number of retransmissions requires greater amount of bandwidth that, consequently, enlarges the clusters' active portions. On the other side, longer active portions imply higher duty-cycle and thus higher energy consumption of the nodes. In addition, longer clusters' active portions may increase the TDCS period which leads to longer end-to-end delays.

References

[1] H. Bai and M. Atiquzzaman. Error modeling schemes for fading channels in wireless communications: A survey. *IEEE Communications Surveys and Tutorials*, 5(2), 2003.

[2] F. Chen and F. Dressler. A Simulation Model of IEEE 802.15.4 in OMNeT++. In *Proc. of the 6th GI/ITG KuVS Fachgesprach Drahtlose Sensornetze (FGSN)*, pages 35–38, July 2007.

[3] F. Chen, N. Wang, R. German, and F. Dressler. Performance Evaluation of IEEE 802.15.4 LR-WPAN for Industrial Applications. In *Proc. of the 5th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS)*, pages 89–96, Jan. 2008.

[4] Crossbow Technology, Inc. TelosB Mote Datasheet, 2009. (http://www.xbow.com).

[5] R. Diestel. *Graph Theory*. Springer-Verlag, 2005.

[6] K. Fall and K. Varadhan. The Network Simulator 2 (ns-2), 2009. (http://www.isi.edu/nsnam/ns).

[7] Z. Hanzálek and P. Jurčík. Energy Efficient Scheduling for Cluster-Tree Wireless Sensor Networks With Time-Bounded Data Flows: Application to IEEE 802.15.4/ZigBee. *IEEE Transaction on Industrial Informatics*, aritcle in press. doi: 10.1109/TII.2010.2050144.

[8] J. Hurtado-Lopez, E. Casilari, and A. Ariza-Quintana. Enabling IEEE 802.15.4 cluster-tree topologies in OMNeT++. In *Proc. of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools)*, Mar. 2009.

[9] IEEE SA Standards Board. Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs). Number 802.15.4, 2006.

[10] P. Jurčík and Z. Hanzálek. TDCS Matlab tool, 2009. (http://dce.felk.cvut.cz/hanzalek/ETFA10).

[11] P. Jurčík and A. Koubâa. The IEEE 802.15.4 OPNET Simulation Model: Reference Guide v2.0. Technical Report HURRAY-TR-070509, CISTER-ISEP research unit, Porto, Portugal, May 2007. (http://www.open-zb.net).

[12] P. Jurčík and A. Koubâa. IEEE 802.15.4/ZigBee Opnet Simulation Model 3.0, 2009. (http://www.open-zb.net).

[13] P. Jurčík, A. Koubâa, M.Alves, E. Tovar, and Z. Hanzálek. A Simulation Model for the IEEE 802.15.4 protocol: Delay/Throughput Evaluation of the GTS Mechanism. In *Proc. of the 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages 109–116, Oct. 2007.

[14] S. Khazzam. IEEE 802.15.4 MAC Protocol Model, 2005. (Opnet Modeler contributed model).

[15] A. Koubâa, M. Alves, and E.Tovar. A Comprehensive Simulation Study of Slotted CSMA/CA for IEEE 802.15.4 Wireless Sensor Networks. In *Proc. of the 6th International Workshop on Factory Communication Systems (WFCS)*, pages 183–192, June 2006.

[16] A. Koubâa, 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. In *Proc. of the 2nd Workshop on Real Time Networks (RTN)*, pages 574–579, July 2006.

[17] A. Koubâa, M. Alves, and 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 (WPDRTS)*, Apr. 2006.

[18] K. Neumann, C. Schwindt, and J. Zimmermann. *Project Scheduling with Time Windows and Scarce Resources*. Springer, 2003.

[19] OPNET Technologies, Inc. The Opnet Modeler network simulator, 2009. (http://www.opnet.com).

[20] C. P. Singh, O. P. Vyas, and M. K. Tiwari. A Survey of Simulation in Sensor Networks. In *Proc. of the 4th International Conf. on Computational Intelligence for Modelling Control and Automation (CIMCA)*, pages 867–872, Dec. 2008.

[21] A. Varga. OMNeT++ 4.0, 2009. (http://www.omnetpp.org).

[22] A. Willig. Recent and Emerging Topics in Wireless Industrial Communications: A Selection. *IEEE Transaction on Industrial Informatics*, 4(2):102–124, May 2008.

[23] J. Zheng and M. J. Lee. *A Comprehensive Performance Study of IEEE 802.15.4*, chapter 4, pages 218–237. IEEE Press, Wiley InterScience, 2006.

[24] ZigBee Standards Organization. ZigBee Specification. Number 053474r13, 2006.