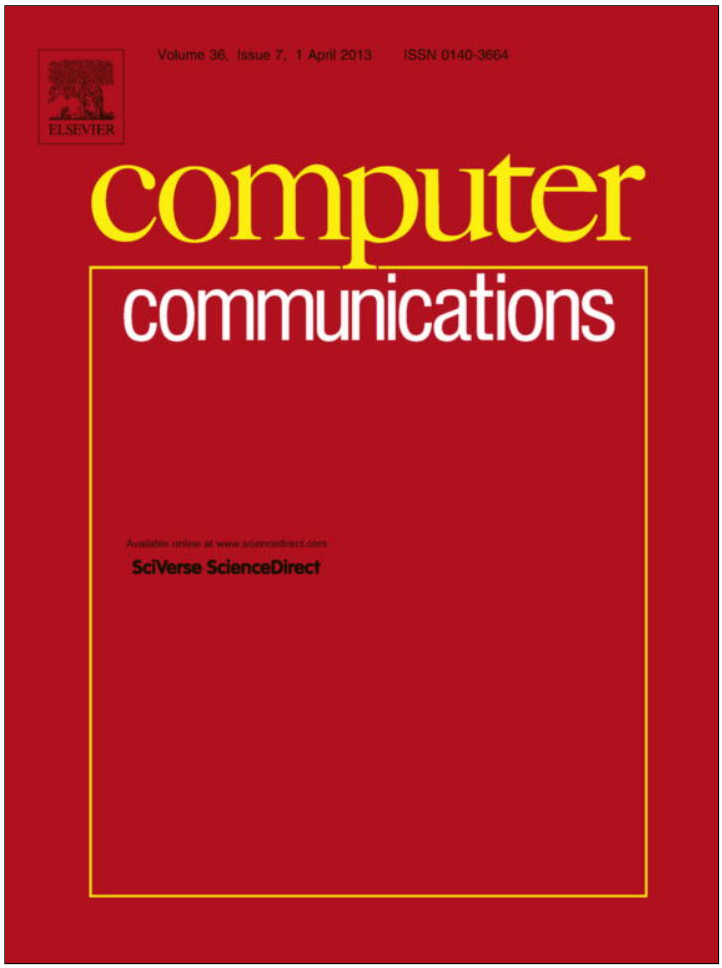
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/authorsrights

Author's personal copy

Computer Communications 36 (2013) 804–816

Contents lists available at SciVerse ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

An energy consumption analysis of the Wireless HART TDMA protocol

Osama Khader [a], Andreas Willig [b,*]

[a] Telecommunication Networks Group (TKN), Technical University of Berlin, Germany
[b] Department of Computer Science and Software Engineering, University of Canterbury, New Zealand

ARTICLE INFO

Article history:
Received 31 May 2012
Received in revised form 28 November 2012
Accepted 30 December 2012
Available online 12 January 2013

Keywords:
Wireless HART
Energy consumption analysis
Response surface methodology

ABSTRACT

In this paper we analyze in detail the energy consumption characteristics of the Wireless HART protocol when operated with a popular transceiver, the ChipCon CC2420. We analyze how much various factors contribute to the overall energy consumption over a longer period of 12 h. These factors include the amount of management traffic and the power levels required for various transceiver activities (transmit, receive, listen, sleep). It turns out that in light traffic scenarios and with only a minimum-complexity level of exploitation of the transceivers sleeping capabilities the energy spent in the sleep state over 12 h is quite substantial. We then proceed to analyze the energy consumption characteristics with a more complex usage of the transceivers sleeping capabilities in which each node individually selects its next sleep state according to its transmission/reception schedule. With this scheme the energy consumption in the sleep state (over 12 h) can be reduced substantially.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

In many application areas of embedded wireless networks, for instance in building automation or industrial control, source nodes send data frames periodically to a gateway or sink node across a set of forwarder nodes [32,33,9]. For cost-effective, quick and scalable deployment, sensor nodes often run on batteries and therefore have only a limited amount of energy. The sensed data should be transported reliably and in a timely fashion to the sink. At the same time the operation of the whole network and of individual nodes should be energy-efficient. Therefore, reporting the sensed data reliably while consuming the minimum amount of energy is of great concern. In many sensor node designs the radio chip is the largest consumer of energy. Since the medium access layer usually controls the states of the radio, it has a large impact on overall energy-consumption. Different media access methods result in different trade-offs between end-to-end delay and energy-efficiency. From among the large number of existing MAC protocols for wireless sensor networks (contention-based protocols include [6,27,38], contention-free protocols include [35,28], see [20] for a survey), a TDMA-based protocol has been chosen as a basis for the Wireless HART (WHART) standard [5]. A common view on TDMA-based protocols is that they offer good opportunities for energy-efficient operation of sensor nodes, as they allow nodes to enter a sleep state when they are not involved in any communications. Furthermore, TDMA is traditionally the method of choice for some of WHARTs major application areas like industrial and process automation, since it offers a level of determinism that is not achievable with other types of MAC protocols. WHART utilizes the physical layer of IEEE 802.15.4 and specifies a new MAC protocol. This new MAC protocol combines slow frequency hopping with a TDMA scheme. The TDMA slot allocation happens a priori at network configuration time. WHART supports multi-hop mesh topologies, and all devices must have routing capabilities.

The first contribution of this paper is a detailed, simulation-based analysis of the energy-consumption behavior of WHART when used with the popular IEEE 802.15.4-compliant ChipCon CC2420 radio transceiver [3] and 8 MHz Texas Instruments MSP430 microcontroller [23].

In doing this, we go beyond mere reporting of energy-consumption figures for certain deployment scenarios. Specifically, we apply the response surface methodology [24] to analyze how various factors influence the total energy consumption. Identifying the factors contributing most to the overall network energy-consumption allows to focus efforts to save energy to the most promising areas. It is worth noting that in our analysis we also include the various WHART protocol overheads, for example the impact of time synchronization and management slots. It is furthermore important to note that, while our analysis focuses on the CC2420, the methodology applies in the same way to other transceivers as well.

We perform this analysis in two stages. In the first stage we do consider sleeping, but we have not fully exploited the sleeping

* Corresponding author. Address: Department of Computer Science and Software Engineering, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand. Tel.: +64 3 364 2987x7869.
E-mail addresses: khader@tkn.tu-berlin.de (O. Khader), andreas.willig@canterbury.ac.nz (A. Willig).

0140-3664/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comcom.2012.12.008

Author's personal copy

capabilities of the CC2420. Instead, we have chosen the lightest available sleep mode. This lightest sleep mode has the highest power consumption of all sleep modes and the shortest wakeup time, and a nodes transceiver enters it after each slot in which the node was active (i.e. transmitting or receiving). This approach has minimal implementation complexity and does not need to consider the nodes TMDA schedule at all – it allows a node to start its wakeup operation at the beginning of a time slot and be ready for transmission or reception within the same slot early enough. Our statistical analysis, which looks at various factors contributing to the energy consumption over 12 h, reveals that the energy consumed in this sleep state contributes to approximately 40% of the total variation observed.

In the second stage of our analysis we exploit that the transceiver offers several different sleep states of varying depth. The deeper sleep states require more energy and time to wake up from them. We analyze a method that individual nodes can apply to their TDMA schedule (as received from a central entity at network configuration time). By this method a node determines after each active slot the deepest sleep mode that would still allow it to be awake when its next active slot comes – it has a slightly higher implementation complexity and runtime costs as the method used in the first stage. This method takes into account how much time it takes to switch from one sleep state into another or into the active state. Our results show that with applying this simple method substantial energy savings for the sleep energy can be achieved.

To the best of our knowledge, this paper provides the first detailed analysis of the main factors contributing to the energy-consumption of WHART. It extends the conference paper [17] by considering all sleep modes of the CC2420 transceiver instead of only the lightest one, by considering management cost in addition to time synchronization cost, by using more complex network topologies and by considering the microcontroller power consumption,

The remainder of the paper is structured as follows. Section 2 presents an overview of the WHART standard. Subsequently, in Section 3 we describe the simulation-based performance evaluation approach used in this paper. The sensitivity analysis results are presented in Section 4. In this section we also present the analysis of the performance cost of time synchronization and management slots. In Section 5 we analyze the transceivers energy consumption when the sleeping capabilities are more fully exploited. Related work is discussed in Section 6 and finally, Section 7 concludes the paper with some possible ideas for future work. Additional details to this paper are available as technical report [16].

2. Overview of Wireless HART

Wireless HART [10,12,11,13] (abbreviated as WHART in the following) is one of the first wireless communication standards specifically designed for process automation applications. The standard has been finalized in 2007, and at the beginning of 2010 it has been ratified as an IEC standard. On the physical layer, WHART adopts radios that are compliant to the IEEE 802.15.4 standard [21]. It operates in the 2.4 GHz band and offers a data rate of 250 kbit/s. On top of the physical layer, WHART employs a TDMA-based MAC protocol and additionally performs slow frequency hopping (hopping on a per-slot basis). The frequency hopping-pattern is determined from a well-known pseudo-random sequence. The TDMA slot allocation is centrally controlled (by the *network manager*) and slots are assigned at network configuration time. Furthermore, the network manager can re-schedule the network dynamically, based on health reports issued by each node every 15 min. Such a health report includes a list of the neighbors of a node and hence a part of the network topology. The network manager re-schedules only upon topology changes, which we consider to be a relatively rare event.

2.1. Network architecture

The network architecture of a WHART network features three different types of components (compare Fig. 1).

The WHART *field devices* are used for collecting measurement data from the field and for forwarding this data to a gateway node. They typically integrate wireless communication, sensing, and computational facilities. A field device might be either a genuine WHART device or it might be a legacy (wired) HART device equipped with a HART-specific wired-to-wireless adapter. In this paper we assume that field devices are battery-driven, so we are especially interested in their energy-consumption.

A *gateway* forms the boundary between a WHART segment and other (often wired) parts of an automation network and is usually not energy-constrained. The gateway is the point where all sensor data provided by WHART field devices is collected and prepared for further processing. It enables communication between host applications and field devices. There is only one gateway per network and all the WHART devices are known to the gateway.

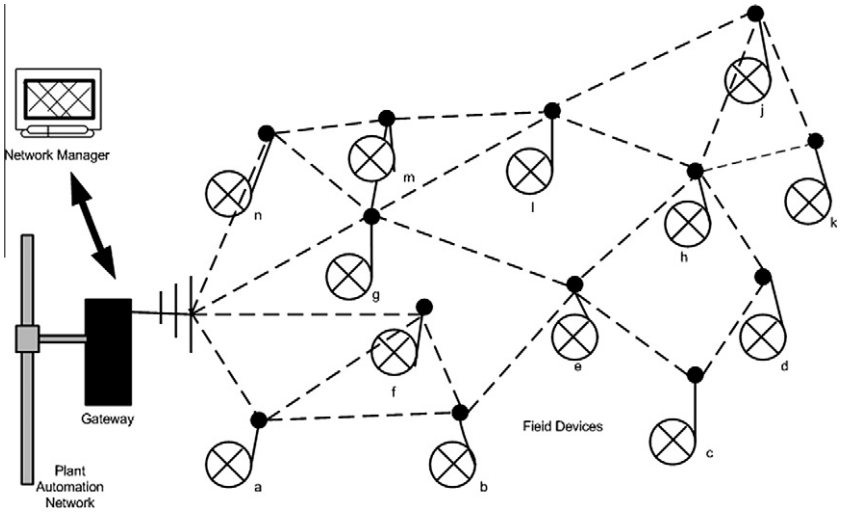


Fig. 1. WHART basic network components.

Author's personal copy

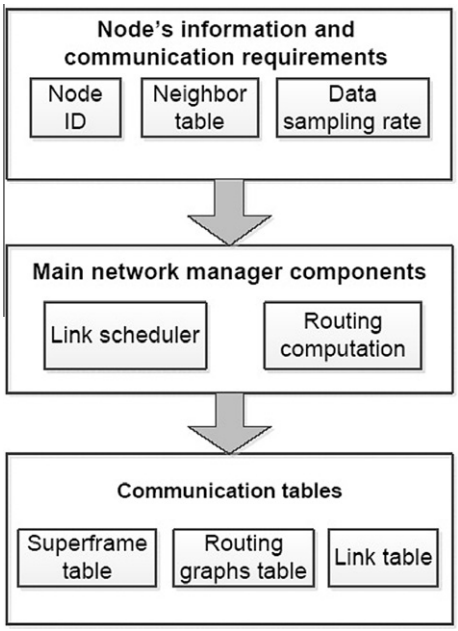


Fig. 2. Core network manager components.

The *network manager* is a centralized unit (Fig. 2 illustrates the core network manager components). It has global information about the network topology, link qualities and traffic flows. To achieve this, each node sends its local information (node ID, neighbor table including observed RSSI values) and its communication requirements (traffic rate, deadlines, priority) to the network manager. Based on this global information, the network manager computes the route graphs, constructs a TDMA schedule and disseminates both to the nodes. Most slots in the TDMA schedule are allocated to a pair consisting of a specific transmitter and a specific receiver, all other nodes are allowed to sleep during such a slot. The network manager not only allocates slots, but places those slots also on different frequencies. The network manager has further responsibilities, including the monitoring and health reporting of the WHART network, and adapting the network to ongoing changes. The network manager is not assumed to be energy-constrained.

2.2. The WHART TDMA scheme

In this section, we briefly discuss the WHART TDMA scheme. The timing hierarchy of WHART has three levels. On the lowest level we have individual *time slots*. Within a time slot one data frame and the accompanying immediate acknowledgment frame are exchanged. A time slot in WHART has a fixed length of 10 ms. A contiguous group of time slots of fixed length forms a *superframe*. On top of that, a contiguous group of superframes forms a *network cycle*. Within each cycle to each field device at least one time slot for data transmission is allocated, but certain devices may have more time slots than other devices because they have more data to report or they have additional forwarding duties.

To join the network and obtain a TDMA schedule, an individual field device first listens for advertisement frames from its neighbors. Once the neighborhood is known, the device sends an admission request frame to the network manager, including its neighborhood information. Afterwards, the device waits to receive its schedule from the network manager. If it does not receive any within a certain amount of time, it repeats the admission request. From its schedule the device knows the time slots where it transmits and those slots where it receives. Furthermore, a field device must maintain time synchronization in order to agree on slot boundaries with neighbored devices.

2.3. Dedicated time slots and time synchronization

In WHART two types of time slots are available: dedicated time slots and shared time slots. Here we consider only dedicated time slots, which are allocated to one specific sender-receiver pair. The internal structure of a dedicated slot is displayed in Fig. 3.

We first discuss the operation of the receiver node. It enters the receive state at the beginning of its time slot. The receiver measures the exact time when it has detected the start of the frame sent by the transmitter. The nominal time for this to happen is `TxOffset` seconds after the start of the slot. Since the time slot duration is fixed to 10 ms, a node can compute the start of the next time slot from the frame arrival time according to:

$$T_{\text{NextSlot}} = \text{FrameArrivalTime} + 10\ \text{ms} - \text{TxOffset} \quad (1)$$

The receiver computes the difference/timing offset between the actual time when it detects the start of the frame and the nominal time. When the receiver has successfully received the frame, it turns his transceiver into transmit mode, generates and sends an acknowledgment frame, and returns to receive mode. This acknowledgment frame includes the measured offset and allows the transmitter to adjust his local time accordingly.

The transmitter also starts a time slot with its transceiver being in receive mode. After time `TxOffset` it turns its transceiver into transmit mode, sends the data frame, switches back into receive mode and waits for the acknowledgment. The measured offset contained in the acknowledgment is used by the transmitter to re-calculate its local view on the start times of time slots and to adjust its local clock.

This approach to clock readjustment is the elementary building block for WHARTs overall time synchronization scheme. Since hardware clocks are generally imprecise, time synchronization is a problem, especially in multi-hop systems [7,8]. There are two main factors affecting accuracy of local time. The first is clock drift, which indicates the rate at which a clocks actual frequency deviates from its nominal frequency; and the second is clock offset, which is the difference from ideal time. For proper operation of the TDMA algorithm neighboring nodes need to agree on boundaries of time slots and therefore a clock synchronization algorithm is needed. WHART uses the above described device-to-device adjustment method for this purpose. To achieve network-wide synchronization, a synchronization tree is built with the gateway as its root. Essentially, devices at depth d in the tree synchronize their clocks to devices at depth $d-1$. The gateway has depth 0, its immediate neighbors have depth 1 and so on.

2.4. Route computation

According to the WHART specification (see [11]) there are two methods of routing frames in a WHART network, graph routing and source routing. We use graph routing in the remaining paper, since it promises a higher resilience to transmission errors. We use the multi-path routing protocol described in [25]. In graph routing, the network manager pre-computes (and subsequently uploads to all nodes) a number of distinct and identified network graphs, so that any node in such a graph has at least two neighbors through which it can send frames to reach a given destination. When sending a frame, a node selects a suitable graph depending on the destination and includes the graph-id into the network-layer frame header. Both graph and source routing require the network manager to explicitly pre-configure routing tables for each potential

Author's personal copy

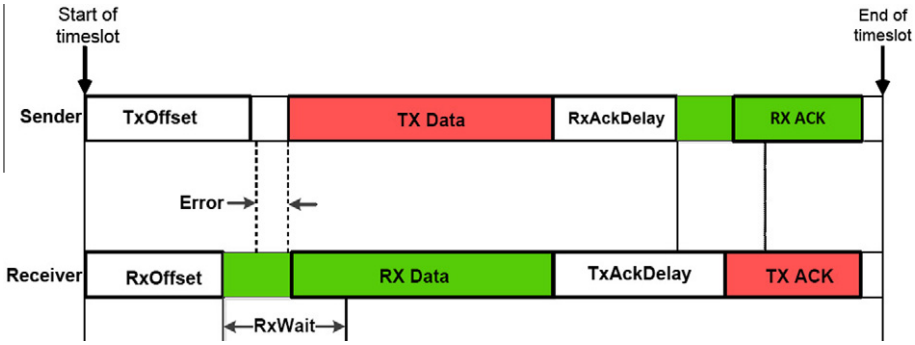


Fig. 3. Dedicated slot timing.

destination by exploiting the link information provided by the nodes.

Graph routing is illustrated in Fig. 4. In this figure, Source1 can communicate with the gateway using GraphA. To send a frame on that graph, Source1 may forward it to F1 or to F2. From those nodes, the frame may take several alternate routes (again on GraphA), but all routes will eventually end in the gateway. Similarly, Source2 can use GraphB to communicate with the GW and the GW can communicate with Source3 using the downstream GraphC.

2.5. TDMA slot allocation algorithm

One of the main tasks in designing a TDMA protocol is the allocation of time slots to sender-receiver pairs. Determining a throughput-optimal schedule for TDMA slot assignment in multi-hop networks is an NP-complete problem, even in linear networks [1]. The WHART standard leaves many details of the slot allocation open, but provides a number of constraints that any such algorithm has to obey. These include:

1. Management slots have priority over data slots.
2. Each device gets three slots every 15 min for health reports.
3. Each device gets at least one slot every minute for management frames (advertisement, join request/response, command request/response).
4. Each device gets a slot for keep-alive frames every 60 s.
5. Slots for stations having the fastest transmission periods are allocated first. We refer to this as the fastest-periodic-flow-first (FPFF) policy.

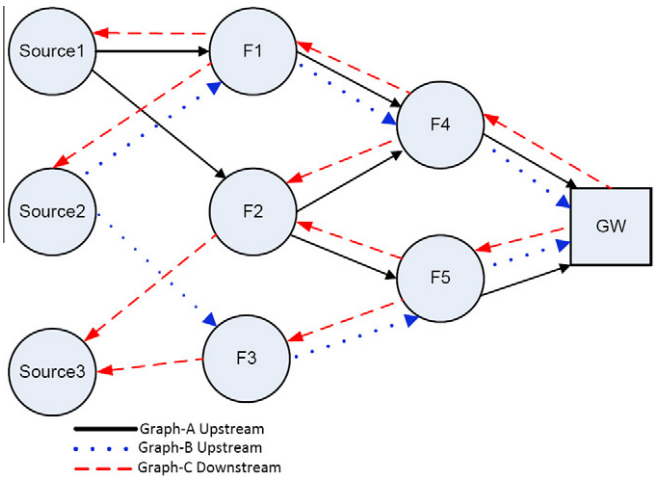


Fig. 4. WHART routing graphs.

6. Allocate at least one backup slot to each data slot to handle a retry.

Please note that even with the FPFF constraint scheduling can be done in several ways. We have used two different approaches, depth-first and breadth-first scheduling, both explained in our previous work [17,16]. The results and conclusions obtained when using either scheduling algorithm are essentially identical. In general, clearly the scheduling algorithm used by the network manager will impact the energy consumption, but it is beyond the scope of this paper and a possible subject of future work to characterize this dependence more thoroughly.

It is also important to note that the WHART standard foresees that extra slots be scheduled for the management frames. Clearly, it would be more energy-efficient if, for example, periodic keep-alive messages could be piggybacked onto periodic data frames.

3. Performance evaluation approach

In this section, we describe our approach for evaluating the energy consumption of the WHART TDMA protocol, focusing entirely on the energy consumed by the transceiver and the controller. Since we are mostly interested in the energy consumption of the field devices, we compute it only for the source and forwarder nodes. We do not take into account the gateway energy-consumption in our calculation as its assumed to be directly connected to a power source.

3.1. Simulation setup

In order to realize a simulation model to study the performance of WHART TDMA over a wireless multi-hop network, we have chosen the OMNeT++ [36] simulation environment together with the Castalia framework [26]. OMNeT++ is an open-source discrete-event simulator, Castalia is an OMNeT++ based framework designed specifically for wireless sensor networks. We set the radio parameters based on the IEEE 802.15.4-compliant ChipCon CC2420 radio chip [3]. The CC2420 operates in the 2.4 GHz ISM band and supports eight transmission power settings in the range between −25 dBm and 0 dBm. For channel errors we use a channel model provided by Castalia (please refer to the Castalia users manual [26]). In particular we use the log-normal shadowing model which has been presented as a reasonable model for the average path loss in [22]. Generally speaking, the path loss PL is a function of the distance d from the transmitter as shown in the following equation:

$$PL(d) = PL(d_0) + 10 \cdot \eta \cdot \log\left(\frac{d}{d_0}\right) + X_\sigma, \quad (2)$$

where $PL(d)$ is the path loss at distance d, $PL(d_0)$ is the known path loss at a reference distance d_0 (which we assume to be 1 m), η is the

Author's personal copy

path loss exponent, and X_σ is a Gaussian zero-mean random variable with standard deviation $\sigma > 0$. For the channel fading model we use a block Rayleigh fading model in which the Rayleigh-distributed fading gain changes for each slot. The values of the channel parameters are listed in Table 2. From the path loss, the current fading gain, and knowledge of the transmit power it then becomes possible to determine the SNR at the receiver and, by knowledge of the modulation scheme, subsequently the instantaneous bit error rate.

The parameters of the path loss model (Eq. (2)) are set according to [22]: $PL(d_0) = 55$ dB, $\eta = 2.4$, $\sigma = 4.0$. Please note that links between two nodes can be asymmetric, as each direction of such a link uses its own realization of the random shadowing coefficient X_σ. Moreover, we have defined -94 dBm as the sensitivity threshold of the CC2420 radio, which is the same value as specified in the data sheet. These calculations are carried out within Castalia.

For our implementation of the WHART TDMA scheme we have modified the radio and channel models of the simulator to support the 16 channels of the IEEE802.15.4 standard. We have also modeled the cost of channel switching. Channel switching is modeled in both receiving and transmission slots within the TxOffset and RxOffset time intervals, respectively. Based on the characteristics of the CC2420 we have assumed that channel switching takes 1 ms and consumes the same amount of power as the transceiver's listen state. The transceiver has four main operational states: transmit, receive, listen and sleep. Given that for long-running applications most of the energy will be spent in the operational phase of a MAC protocol, we have chosen to ignore the initialization phase, in which for example the nodes report their neighbor lists and channel qualities, the network manager disseminates the schedules, and initial time synchronization is established. The TDMA schedule is precomputed based on the fastest periodic flow first (FPFF) scheduling algorithm and given as input to the simulator. In Table 1 the main power consumption parameters of a CC2420 transceiver and MSP430 microcontroller are summarized assuming a 3.3 V supply voltage. These parameters are used in the physical layer model of our simulator to obtain energy-consumption results. The sleep power mentioned in the table refers to the lightest of four available sleep states. Please note that the microcontroller is active at the same times as the transceiver.

Further simulation parameters related to the physical and MAC layer properties and the node deployment are given in Tables 2 and 3, respectively.

In addition to the microcontroller we have also taken the power consumption of the microcontroller into account. More precisely, we use a MSP430 microcontroller from Texas Instruments. This controller alternates between two different states: an active state, in which it performs computations, and a sleep state in which major parts of its circuitry are switched down. The power consumption figures of both states are given in Table 1. Since we have not considered any specific application in this paper, we assume that the microcontroller is active at the same time as the transceiver. This corresponds to mainly event-driven or reactive systems, in which computations are only be carried out in response to external events (e.g. packet arrivals).

Table 1
Power consumption of CC2420 radio with 3.3 V supply voltage and of the MSP 430 microcontroller.

Notation	Parameters	I (mA)	Power (mW)
Main power consumption of CC2420 radio			
P_{Tx}	Transmit power (0 dBm)	17.4	57.42
P_{Rx}	Receive power	18.8	62.04
P_L	Listen power	18.8	62.04
P_S	Sleep power	0.426	1.406
CPU_A	CPU active power	1.8	6
CPU_S	CPU sleep power	0.045	1.48

Table 2
General parameters.

Parameter	Value
Main radio and MAC parameters	
Radio layer	CC2420
Data rate	250 kbps
Frame size	133
CC2420 sensitivity	−95 dBm
Noise floor	−100 dBm
$PL(d_0)$	55 dB
η	2.4
X_σ	4.0
Number of slots per superframe	100 s
Slot-time length	10 ms
Synchronization frame size	26 bytes
Resynchronization rate	60 s
Health report rate	15 min

3.2. Network topology and traffic

In order to perform our evaluation, we use randomly generated topologies. More precisely, we have generated 100 random topologies and for each setting of simulation parameters we correspondingly perform 100 replications, the energy consumption results of which are then averaged. For each random topology we have placed 150 nodes in an area of size 120×120 m^2, using a uniform distribution for node positions. The gateway is placed in the upper right corner of the field. Out of the 150 nodes we randomly pick ten nodes as source nodes. Each of these sources periodically generates frames of 133 bytes total size (including PHY and MAC parts). The generation period was varied, taking values of 1, 30, or 60 s, respectively. During each simulation-run, each source transmits frames based on its periodicity and then forwards these frames to the GW via some forwarders. MAC-layer acknowledgments are enabled and the size of the ACK frame is 26 bytes. If the frame is lost due to channel errors the sender tries to transmit the frame for a maximum of two retries.

3.3. Major performance measure

The simulation time is fixed to 12 h for each scenario and the main performance measure is the total energy spent by the radio transceiver and the microcontroller of a node over this period. This time duration of 12 h is on the one hand long enough to give meaningful results for the energy spent in sleep states, and on the other hand is not too taxing on simulation times.

For the transceiver, the simulation records the amount of time spent in various states (transmit, receive, listen, sleep and turnover) and calculates from this the total energy consumption of a node over a span of 12 simulated hours. We also model the energy spent in switching between states by multiplying the turnover time by the power consumed in the most power-consuming of the two involved states. We also take into consideration the energy consumed by the nodes microcontroller. We split the node's microcontroller energy consumption into two main states, active state and sleep state. The microcontroller is active at the same time as the radio. At the end of each run, the simulation computes the total energy consumed for all field devices in the network using the amount of energy consumed by the radio and microcontroller in each state.

This 12-h energy consumption is obtained for each source and router node separately. In the next sections we report mainly the average energy consumption, taken over all nodes.

4. Sensitivity analysis

The major goal of our study is not only to obtain insights into the overall network energy-consumption of the different types of

Author's personal copy

Table 3
WHART TDMA MAC parameters.

Notation	Parameters	Value (ms)
WHART MAC parameters		
TxOffset	The guard time at beginning of time slot at the sender side	2.12
RxOffset	The guard time at beginning of time slot at the receiver side	1.12
RxWait	The time to wait for start of frame	2.2
Maxframe	Maximum frame length	4.256
TxAckDelay	The time between end of frame and start of ACK at the receiver side	0.8
RxAckDelay	The time between end of frame and start of listening for ACK at the sender side	0.9
AckWait	The minimum time to wait for start of an ACK	0.4
Ack	ACK (26 bytes)	0.832
RxTx	TxRx turnaround time	0.192

nodes (sources and forwarders), but we also want to obtain some insights on how the average total consumption (taken over 12 h, see Section 3.3) breaks down into different factors. By identifying the factors contributing most to the overall energy-consumption, we can provide guidance on where to start with any effort geared towards saving energy.

It is reasonable to assume that, given the long time span of 12 h over which energy consumption is computed, the energy spent in the sleep state will play a significant role. Therefore, we need to clarify our assumptions on how the four available sleep states of the CC2420 (compare Section 5) are used in this analysis. As stated in the introduction, the only sleep state that works indiscriminately for every possible schedule is the lightest sleep state, which we have used here (the power consumption of which is given in Table 1). This sleep mode is entered after each active slot and its wakeup time is quick enough so that wakeup can commence at the beginning of the nodes next active slot. An advantage of this method is that there are no runtime costs for determining the next sleep state. In the next Section 5, we report a simple method which for a known schedule exploits the other available sleep states to obtain higher savings.

Our major tool for energy consumption analysis is the response surface methodology (RSM). RSM is a collection of mathematical and statistical techniques useful for modeling and analysis of problems in which a response of interest is influenced by several variables (factors). Usually the relation between the response variable Y (performance measure) and the input variables (or "factors") x_i is unknown. After identifying the factors and the response(s) under study, the next step in RSM is to find a suitable approximation for the response surface and check whether or not this model is adequate.

The two most popular choices for regression models are first-order (or linear) and second-order models. Linear models have the form

$$Y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \epsilon \tag{3}$$

and are mostly appropriate when one is interested in approximating the true response surface over a very small region of the input variables, where the response surface is approximately constant (no curvature). If there is curvature in the system, then a polynomial of higher degree must be used, such as the second-order regression model:

$$Y = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \sum_{i=1}^{k} \beta_{ii} x_i^2 + \sum_{i=1}^{k} \sum_{j<i} \beta_{ij} x_i x_j \quad \text{for } i = 1, 2, \ldots, k, \text{ and } j = 1, 2, \ldots, k \tag{4}$$

In both Eqs. (3) and (4) the variable Y is the response variable, $x_1, x_2, \ldots, x_k$ are the input factors, k is the number of factors, β_0 is the (unknown) intercept term of the system and β_i and $\beta_{i,j}$ are the unknown regression coefficients. These coefficients determine the expected change in the response Y per unit change in each factor x_i when the remaining factors are kept fixed. Many studies in RSM show that second-order models are capable of solving real response surface problems that have interactions and curvature. This is due to the fact that second order regression model offers a wider variety of functional forms. For these reasons we have decided to adopt the second-order model of Eq. (4). For more details about the RSM we point the reader to the [29,24].

We have used the total energy consumption taken over 12 h as the response variable Y. The factors x_i are described in the next Section 4.1, they include the transmission power, reception power, the rate of management packets and three others. Please note that in Eqs. (3) and (4) the response variable Y should have the same dimension as the input variables, and all factors x_i should enter with the same order of magnitude. In applications like ours, often the factors have incompatible units (like the transmit power expressed in mW or the management rate in Hz, see Section 4.1). To accommodate this, all variables are formally dimension-less, although the response variable will be interpreted as being in mW. Furthermore, the factors in the RSM method are not used directly, but one chooses two "extreme" values of a factor (e.g. the minimum and maximum transmit power) and encodes these as "−1" and "1", respectively. When there are k factors, one performs 2^k experiments over the set $\{-1, 1\}^k$ of all possible allocations to factors. Once the response values Y have been obtained for all experiments, the regression coefficients β_i and $\beta_{i,j}$ are determined for the encoded input factors x_i (i.e. assuming that these input factors assume the values −1 and 1, respectively). We have used the Matlab SUMO[1] and statistics toolboxes for this. Internally, these toolboxes use a least-squares algorithm for estimating these coefficients. In fact, for each of the 2^6 possible factor combinations we have performed 100 replications and used the average total energy consumption, taken over all these replications, as our response variable. This averaging is the reason why the model (4) does not include a separate error term, as with 100 independent replications the simulated averages will be close to the true averages.

After identifying the appropriate empirical fitted model, we examine this model to ensure that it provides an adequate approximation to the true responses and to verify that none of the assumptions for least squares regression are violated. There are several techniques for testing the model adequacy and the goodness of the fit, for example: testing for normality of residuals, analysis of variance (ANOVA), lack of fit test, R^2 and others. We will show some results of these tests in Section 4.2. If the fitted surface is an adequate approximation of the true response function, then analysis of the fitted surface will be approximately equivalent to analysis of the actual model.

A key information that can be derived from the regression model is the percentage contribution of each factor (and all considered factor combinations) in explaining the variation of results across the parameter combinations taken into account.

4.1. Factor screening

The first step in the RSM is to identify potential factors affecting the response being measured (factor screening). Since the average total energy-consumption is the main response, we consider the following factors:

[1] See <www.sumo.intec.ugent.be/?q=sumo_toolbox>.

Author's personal copy

Table 4
The factors and the levels of each factor.

Term	Factor	Level 1(−1)	Level 2(+1)
A	Tx power	32.67 mW	57.42 mW
B	Rx power	31.68 mW	62.04 mW
C	Listen power	31.68 mW	62.04 mW
D	Sleep power	0.72 mW	1.41 mW
E	Turnaround power	31 mW	62 mW
F	Management rate	60 s	120 s

F-
actor *A* – transmission power: The transmission power is the power consumed for transmitting data frames and control frames such as synchronization frames.

- Factor *B* – reception power: The receiving power is the power consumed while receiving data or control frames.
- Factor *C* – listening power: The listening power is the radio power consumption when the radio is on but not receiving or sending any frames.
- Factor *D* – sleeping power: The sleeping power is the power consumption while the radio is in the low-power state.
- Factor *E* – turnaround power: The turnaround power is the power consumed while switching the radio state between different modes.
- Factor *F* – management rate: The energy consumed for maintaining the WHART network operations. This is controlled by the frequencies of synchronization, advertisement, join request/response, commands, keep-alive and health report control frames. We assume that all these different frame types occur with the same frequency.

Please note that for a realistic application and hardware platform there would be some more factors influencing the energy consumption, for example the power consumed by the sensors. However, we have identified the above listed factors as the most relevant for our purposes. Table 4 lists the factors and the levels of each factor considered in our study.

4.2. Analysis of the results

Table 5 shows the percentage which the individual factors and their pairwise combinations contribute to the variation of total energy-consumption over all 2^6 different factor combinations.

From the statistical analysis it can be seen that factor *C* (listening power), *D* (sleep power) and *F* (management rate) contribute most to the total energy-consumption. Please note that most of the quadratic factors $x_{i,i}$ have negligible weights (see our technical report [18] for complete tables of all factors). The regression analysis using the least-squares estimation method on the values of the response obtained from the various combinations of the factors yields the following Eq. (5) for the total energy-consumption:

$$\begin{aligned} T_E = {} & 111.33 + 2.59A + 2.83B + 11.94C + 13.77D + 0.71E \\ & - 12.73F - 0.19AB - 0.34AC + 0.74AD - 0.34AE \\ & + 0.39AF - 0.34BC + 0.74BD - 0.34BE - 0.41BF \\ & + 0.59CD - 0.19CE - 3.66CF - 0.34DE - 0.11DF \\ & - 0.40EF + 0.34ABC - 0.74ABD \end{aligned} \quad (5)$$

It is always necessary to examine the fitted model to ensure that it provides an adequate approximation to the true system model. To check the model adequacy, and to get more statistical details of the factors affecting the total energy-consumption, we perform an analysis of variance (ANOVA) as shown in Table 6 (this and the other statistical tests were carried out with MATLAB). The ANOVA shows the effect of each design factor on the total energy-consumption and their statistical significance through the *F*-test and associated probability ($Prob > F$). The *F*-value column reports the ratio of the mean squares of the model over the mean squares of the residual. The *F*-value is compared to the reference distribution for *F*, in order to determine the probability of observing this result due to error. If the value in the last column of the table is less than 0.05 (at a 95% significance level), then the factor is statistically significant. In other words, there is a very small probability, near 0.01% that the differences in the factors model averages are due to the chance variation. The results given in Table 6 demonstrate that all the elementary factors *A* to *F* and the compound factor (*CF*) are highly significant, together they explain almost all the variation. Based on the ANOVA test we have simplified the regression model by excluding insignificant factors to become:

$$\begin{aligned} T_E = {} & 111.33 + 2.59A + 2.83B + 11.94C + 13.77D + 0.71E \\ & - 12.73F - 3.66CF \end{aligned} \quad (6)$$

We have applied a number of further tests (comparison of observed responses with responses predicted by the regression model, testing whether the residuals of total energy consumptions are normally distributed, and a check for lack of correlation between the residuals and the order in which runs are carried out), and all these tests confirm that the regression model is a very good approximation of the real total energy consumption. The details are reported in the technical report [18].

4.3. Impact of traffic density

In our next analysis we have looked at the impact of the intensity of data traffic on the overall energy consumption picture. To keep the presentation simple, we have analyzed how the coefficients β_i for the main factors *A*, *B*, *C*, etc. (which in total have far more impact on the response than the combined factors *AB*, *AC* and so on) change when either the number of data sources or the data generation rate changes.

Specifically, in Figs. 5 and 6 we show the impact of the number of sources and traffic rates, respectively. Please note that for Fig. 5 we vary the number of sources from 2 to 10 sources. It can be seen that the most sensitive parameter is the regression coefficient β_C for factor *C* (sleeping power). Fig. 6 shows the impact of the traffic

Table 5
The percentage of factors contribution.

Term	Sum of squares	Percentage contribution
A	428.82	1.26
B	513.17	1.51
C	9131.35	26.92
D	12129.30	35.76
E	32.02	0.09
F	10375.50	30.59
AB	2.38	$7.022 \cdot 10^{-3}$
AC	7.41	0.02
AD	34.90	0.10
AE	7.41	0.02
AF	9.70	0.03
BC	7.41	0.02
BD	34.91	0.10
BE	7.41	0.02
BF	10.71	0.03
CD	22.37	0.07
CE	2.38	$7.025 \cdot 10^{-3}$
CF	859.56	2.53
DE	7.41	0.02
DF	0.72	$2.11 \cdot 10^{-3}$
EF	10.03	0.03
ABC	7.41	0.02
ABD	34.91	0.10

Author's personal copy

Table 6
ANOVA for total energy-consumption.

Source	df	Sum of squares	Mean square	*F*-value	*Prob* > *F*-value
Model	23	$3.367E \cdot 10^4$	1464.23	238.30	0.0001
A	1	428.82	428.82	69.79	0.0001
B	1	513.17	513.17	83.52	0.0001
C	1	9131.35	9131.35	1486.08	0.0001
D	1	12129.30	12129.30	1973.99	0.0001
E	1	32.02	32.02	5.21	0.0278
F	1	10375.54	10375.54	1688.57	0.0001
AB	1	2.38	2.38	0.39	0.5371
AC	1	7.41	7.41	1.21	0.2787
AD	1	34.90	34.90	5.68	0.0220
AE	1	7.41	7.41	1.21	0.2786
AF	1	9.70	9.70	1.58	0.2162
BC	1	7.41	7.41	1.21	0.2787
BD	1	34.91	34.91	5.68	0.0220
BE	1	7.41	7.41	1.21	0.2786
BF	1	10.71	10.71	1.74	0.1942
CD	1	22.37	22.37	3.64	0.0636
CE	1	2.38	2.38	0.39	0.5370
CF	1	859.56	859.56	139.89	0.0001
DE	1	7.41	7.41	1.21	0.2787
DF	1	0.72	0.72	0.12	0.7346
EF	1	10.03	10.03	1.63	0.2088
ABC	1	7.41	7.41	1.21	0.2787
ABD	1	34.91	34.91	5.68	0.0220
Error	40	$2.4578 \cdot 10^{-6}$	$6.14 \cdot 10^{-8}$		
$R^2 = 0.99$					

rate on the regression coefficients β_i (here the number of sources is fixed to 10). One can see from this Fig. 6 that the regression coefficients β_i are more of less the same in case of low traffic rate such as 30 and 60 s.

4.4. Impact of control frames on the energy-consumption

In order to investigate the impact of the control frames (time synchronization and management frames) on the total energy-consumption, we compare the same setup as described in Section 3.1 with and without running the synchronization protocol as well as the management frames. Please note that in the WHART standard there are two separate superframes. One is used for data frames (data slots) and the other one used for management frames (management slots). Thus, management frames are transmitted separately and cannot be piggybacked onto data frames. We distinguish two types of control overhead slots: synchronization control overhead slots (SCOS) and management control overhead slots (MCOS). The SCOS includes the slots that are responsible for synchronization, such as keep-alive frames. On the other hand, MCOS includes the frames (advertisements, request/response frames, commands frames, health report frames, etc.) send throughout the network to establish and maintain the WHART operations.

To show how the SCOS and MCOS affect the total energy consumption, we first analyze the impact on the regression coefficients β_i for different management traffic rates. In Figs. 7–9 we show these results for management/synchronization traffic periods of 1 s, 30 s, and 60 s, respectively. The variations in management traffic rates have significant impact on the regression coefficients for factors *C* (listen power) and *F* (management rate). Note that the curve "w/o SCOS" refers to a setup in which neither synchronization nor management slots are present. We can also observe from Figs. 8 and 9 that with low management traffic rates of 30 and 60 s the β_i parameters are very stable for all the factors. Please

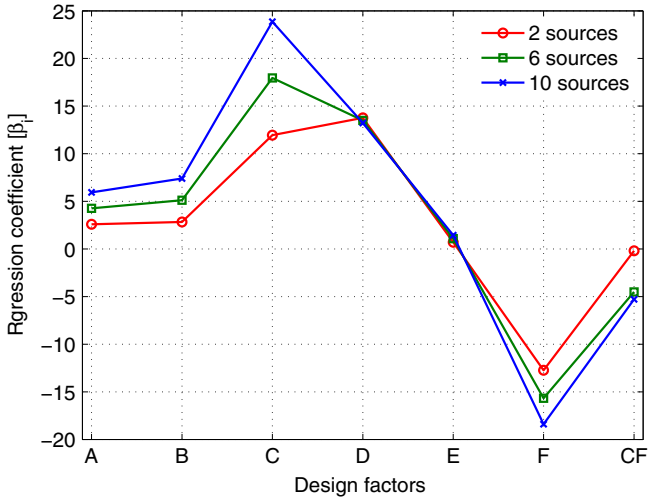


Fig. 5. Impact of the number of sources on regression coefficients β_i.

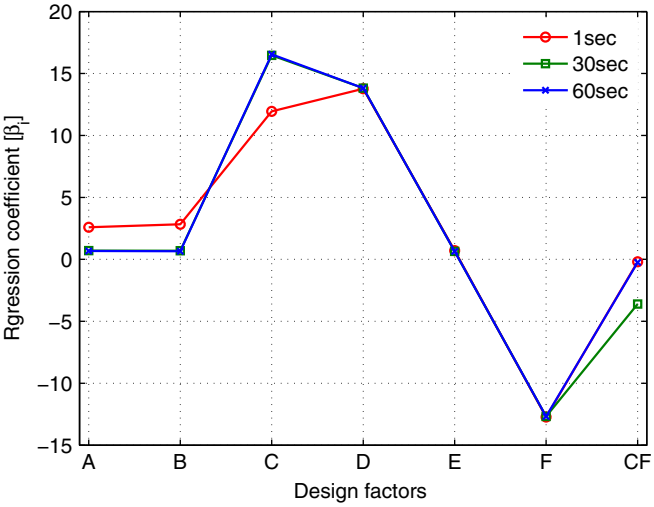


Fig. 6. Impact of traffic density on regression coefficients β_i.

note that, the remaining higher order terms such as *AA*, *AB*, *AC*, etc. have negligible impact on the regression analysis.

In addition, we also provide results showing how serious the impact of SCOS and MCOS is on nodes being in different hop distance to the gateway. Specifically, we compute the average total energy consumption for all nodes being one hop away from the gateway (H1), for all nodes being two hops away (H2) and so on. The energy-consumption of the nodes near the gateway varies is based on the number of control frames received by these nodes, this is shown in Fig. 10 for a scenario with ten sources and a data generation period of 1 s (the results for data generation periods of 30 and 60 s exhibit the same pattern and are shown in the technical report [18]).

We can also observe from Fig. 10 that all the nodes have the same energy-consumption with respect to hop number for the no-synchronization scenario (w/o SCOS). The energy-consumption in case of synchronization frames (SCOS) and management frames (MCOS) becomes higher as the nodes gets closer to the gateway. This is due to the high number of slots needed in order to forward the MCOS traffics. In light of that, it is much more economic to piggyback some percentage of the MCOS traffics on the data frames to save energy. Indeed, this might save a significant amount of energy in case of high data rates.

Author's personal copy

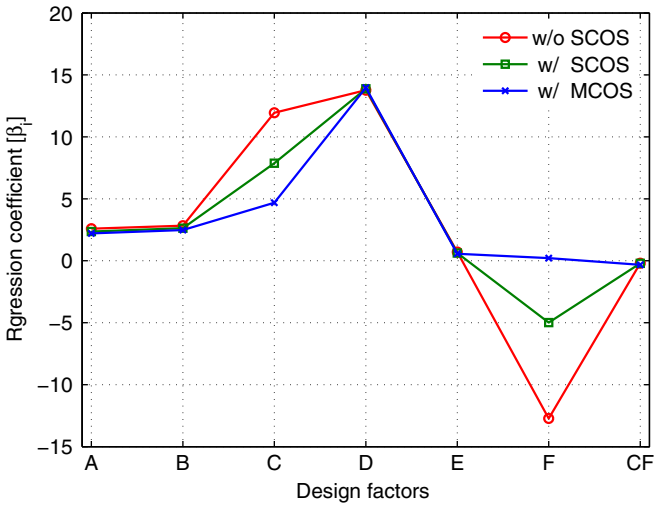


Fig. 7. Impact of control frames on regression coefficients β_i in case of 1 s.

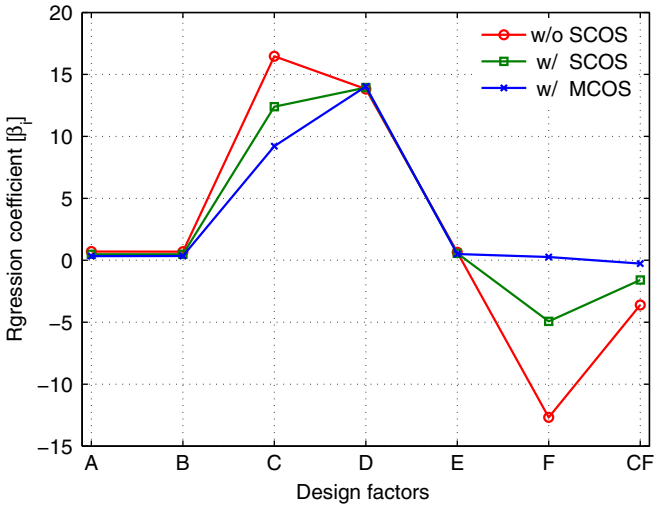


Fig. 8. Impact of control frames on regression coefficients β_i in case of 30 s.

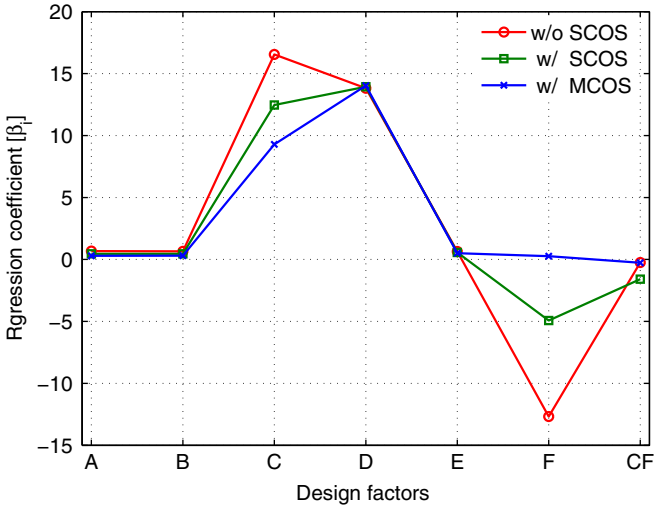


Fig. 9. Impact of control frames on regression coefficients β_i in case of 60 s.

Finally, we have a closer look at the influence of the power consumption of the microcontroller. Specifically, in Fig. 11 we compare the average total power consumption for varying hop distance to the sink with and without taking into account the microcontrollers power consumption. It can be seen that the microcontroller (which is active at the same time as the transceiver) accounts for difference of approximately 15% of the total power consumption, but clearly the transceiver has much bigger influence on total power consumption.

4.5. Discussion

Our results indicate that the time spent in the sleeping state is the major contributor to the total energy consumption. This is due to the long sleeping intervals that accumulate over the lifetime of the network and restricting to the lightest possible sleep state. The second main contributor is channel listening in the data slots and management slots. Most of the activity within these slots (waking up from sleep mode, switching transceiver into right mode, transmitting or receiving data or ACK frames) is inevitable, but especially receiving nodes have to spend some of their listening time (in total about 1 ms) to accommodate clock inaccuracies. The energy consumption increases when data frames are lost and there are retransmissions.

The presence of management and synchronization slots also has significant influence on total energy consumption. These slots require nodes to switch the transceiver on, and to transmit and receive frames without effecting useful data transfer. The frequency of these frames has a significant influence on the total energy consumption. WHART control frames include resynchronization, advertisements, request/response, commands, keep-alive and health report frames. The relative energy cost incurred from the management frame frequency becomes especially significant in case of low traffic rates. We believe that the issue of selecting the right management frequency is worth further investigation. One idea is to adapt the frequency of management traffic by starting with a high frequency, and as soon as the network becomes somewhat stable, the management rate can be reduced. This can significantly reduce energy saving at the expense of longer joining times and slower network update times resulting from topology changes. Another approach would be to use piggybacking more extensively, for example to use periodic data packets also for man-

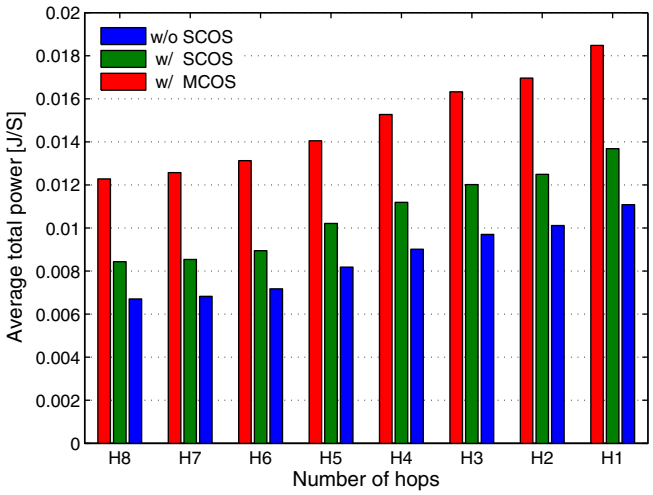


Fig. 10. Control overhead cost in case 1 s: each hop corresponds to the set of neighbors that are n hop away from the gateway.

Author's personal copy

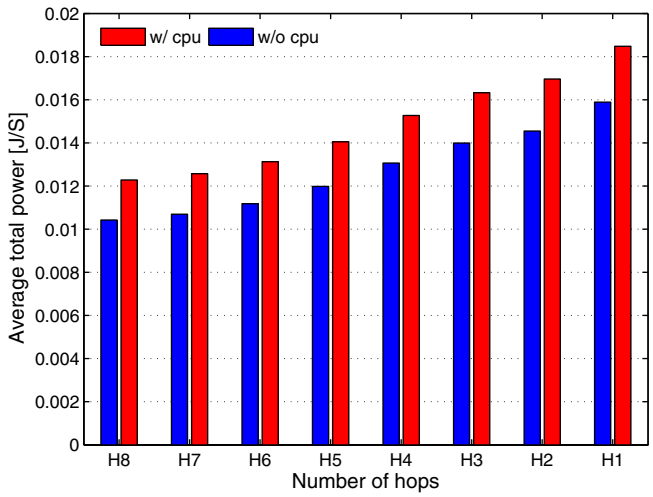


Fig. 11. Average energy consumption with and without considering the microcontroller for traffic period of 1 s. Each hop corresponds to the set of neighbors that are *n* hops away from the gateway.

agement purposes by piggybacking additional information (e.g. keep-alive and health reports).

5. Local dynamic sleep state scheduling

In this section we analyze how an improved usage of the transceiver sleep states can substantially reduce the overall energy-consumption over 12 h, thereby increasing the WHART TDMA system energy-efficiency. Based on the previous WHART sensitivity analysis using only the lightest sleep state we found that indeed the energy spent in sleep state is one of the major factors influencing the total energy-consumption.

Many modern radios have built-in support for several sleeping states of operation with each state consuming a different amount of power. The radio also requires some time to switch into and out of different sleep states. For example, the CC2420 has three sleeping states: the idle-sleep-state, the power-down-state, and voltage-regulator-off-state, hereafter referred to as sleep-mode-1, sleep-mode-2, and sleep-mode-3, respectively. These sleep modes and their possible transitions are illustrated in Fig. 12. In sleep-mode-1, both the voltage regulator and the crystal oscillator are enabled. The energy-saving in sleep-mode-1 state is obtained by disabling the radio frequency synthesizer which controls the channel selection and up/down RF conversion. Sleep-mode-1 has the fastest transition time of around 0.192 ms and consumes 1.4 mW of power, which is the highest among the sleep modes. In sleep-mode-2, the voltage regulator is enabled and the crystal oscillator is disabled. This mode consumes 0.07 mW of power. In sleep-mode-3, both the voltage regulator and the crystal oscillator are disabled. This mode has the slowest transition time and lowest power-consumption ($6.6 \cdot 10^{-5}$ mW). In general this mode switches off the radio chip completely, including radio RAM. As a result any frame waiting in the receiving or transmitting buffer is lost.

As we have discussed in Section 4.5, in the WHART TDMA system a node sleeps for the most part of its life and the energy consumed in sleeping state reaches a substantial share of overall energy consumption over 12 h when sleep-mode-1 is used throughout (which works with any schedule) – the sleep energy amounts to almost 40% of the total observed variation in responses.

One approach to exploit the other sleep states as well would be to make the fixed-length WHART timeslots of 10 ms duration somewhat longer, so as to allow wakeup from deeper sleep states within a timeslot. However, this would require a change to the standard itself, and it would also affect existing implementations.

In the remainder of this section we propose a simple approach that does not require any changes to the standard and which each node can apply individually, based on its schedule.

5.1. Dynamic multiple sleep states scheduling (DM3S)

In what follows, we propose a practical and effective dynamic multiple sleep states scheduling scheme, abbreviated as DM3S. It exploits the multiple sleep states of the CC2420 radio and utilizes them without any modification of the WHART TDMA standard. This approach is independent of the underlying link scheduling algorithm, but a node uses its given schedule to determine the right sleep states. Since many other modern radios do also have multiple sleep states with the same type of tradeoff between power consumption at sleep time and wakeup time, we believe that the general approach of DM3S is transferrable to other such radios as well.

Generally speaking, in WHART nodes activities are constrained to certain slots (whether these are exclusive or shared does not matter for the following presentation), whereas in all other slots they can sleep. We call the slots that a node might be involved in its *active slots*. There will generally be some active slots in which a node will have to wake up unconditionally, for example those slots in which the node is scheduled to receive, or those transmit slots where a frame is transmitted the first time. On the other hand, retransmission slots are only used when a transition in a pre-

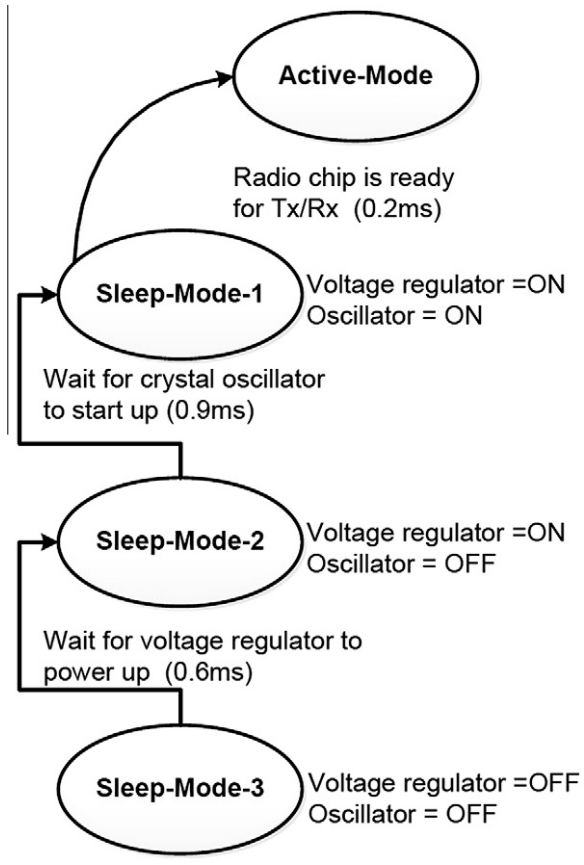


Fig. 12. Sleep transition states for CC2420 radio.

Author's personal copy

vious transmit slot has failed (i.e. the sender has not received an acknowledgment). A key observation is that at the end of a transmit slot the sender will know if it has to utilize a retransmission slot or not. More generally, based on its schedule and the transmission outcomes in the current active slot, at the end of the current slot a node can determine how much time will elapse before its next active slot starts.

The second key ingredient is borrowed from a technique used in dynamic power management to control the device's operational states, see [14,2]. Specifically, since the number of transceiver states and their switching time is known a priori, it is possible to construct a function $\phi(\cdot)$, which takes a non-negative time duration τ as a parameter and which returns a sleeping schedule that:

(i) ensures that after τ seconds the node transceiver is ready to transmit or receive,
(ii) sends the transceiver through a "monotone" sequence of sleep states (the deepest state at the beginning and the lightest state at the end), and that
(iii) ensures that the chosen sequence of states (and the times being spent in each visited state) has the smallest energy consumption over the time horizon of τ seconds.

For the CC2420 transceiver this function $\phi(\cdot)$ is straightforward to construct. Specifically, we need to determine three threshold values:

(i) a duration τ_1 that is minimally needed to make sleep-state-1 more energy-efficient than to stay awake;
(ii) a duration $\tau_2 > \tau_1$ that is minimally needed to make an initial choice of sleep-state-2, followed by a transition through sleep-state-1 and subsequent wakeup more energy-efficient than to start initially with sleep-state-1; and
(iii) a duration $\tau_3 > \tau_2$ that is minimally needed to make an initial choice of sleep-state-3, followed by a transition through sleep-state-2, sleep-state-1 and subsequent wakeup more energy-efficient than to initially start with sleep-state-2.

Table 7
The percentage of factors contribution for DM3S.

Term	Sum of squares	Percentage contribution
A	358.72	1.65
B	436.15	2.00
C	8796.50	40.38
D	216.85	1.00
E	15.14	0.07
F	11167.52	51.27
AB	0.05	$2.319 \cdot 10^{-4}$
AC	0.91	$4.177 \cdot 10^{-3}$
AD	1.56	0.01
AE	0.91	$4.176 \cdot 10^{-3}$
AF	1.81	$8.311 \cdot 10^{-3}$
BC	0.91	$4.177 \cdot 10^{-3}$
BD	1.56	0.01
BE	0.91	$4.176 \cdot 10^{-3}$
BF	2.26	$1.039 \cdot 10^{-2}$
CD	0.01	$2.398 \cdot 10^{-5}$
CE	0.05	$2.322 \cdot 10^{-4}$
CF	759.01	3.48
DE	0.91	$4.175 \cdot 10^{-3}$
DF	0.10	$4.724 \cdot 10^{-4}$
EF	1.96	$8.976 \cdot 10^{-3}$
ABC	0.91	$4.176 \cdot 10^{-3}$
ABD	1.56	0.01

When at the end of an active slot it takes a time τ before the next active slot starts, it is a simple matter of comparing τ to the three thresholds τ_1, τ_2 and τ_3 to figure out which sleep state (if any) should be entered next. The run-time overhead caused by this computation is only moderate.

5.2. Evaluation and results for the DM3S approach

In order to evaluate the efficiency of the DM3S approach we perform simulations using the same setup as described in Section 3.1. We first conduct a regression analysis similar to the one in Section 4, then we provide a breakdown of the average energy consumption based on the hop distance of nodes to the gateway.

The results of the regression analysis when DM3S is used are shown in Table 7. Specifically, this table shows the contribution of the individual factors and their pairwise combinations to the variation of total energy-consumption. From this analysis it can be seen that the impact of factor *D* (sleep power) has been reduced drastically, this factor now accounts for only ≈1% instead of 35% of the total variation in energy consumption. We can also observe that now factors *F* (management rate) and *C* (listen power) contribute most to the total energy-consumption. Please note that again most of the quadratic factors have negligible weights (see our technical report [18] for complete tables of all the factors). The analysis of variance (ANOVA) analysis (not shown here, but in [18], together with the results of the other statistical tests) confirms that, similar to the case without DM3S, all the elementary factors *A* to *F* and the compound factor (*CF*) are highly significant, together they explain almost all the variation. Based on the ANOVA test we have simplified the regression model using the least-squares estimation method by excluding insignificant factors to become:

$$T_E = 68.98 + 2.37A + 2.61B + 11.72C + 1.84D + 0.49E - 13.21F - 3.44CF \quad (7)$$

Please note that in comparison to Eq. (6) the intercept term has been reduced from 111.33 to 68.98, and the coefficient for factor *D* (sleeping energy) has reduced from 13.77 to 1.84.

Moreover, in Fig. 13 we show the average energy-consumption of each hop for both normal operation and DM3S approach in a scenario with ten sources generating traffic with a period of one second. We can see that the multiple sleep state scheduling leads to significantly lower energy consumption than the operation with just sleeping-mode-1. Similar trends are observed also for scenarios with 30 and 60 s traffic generation period.

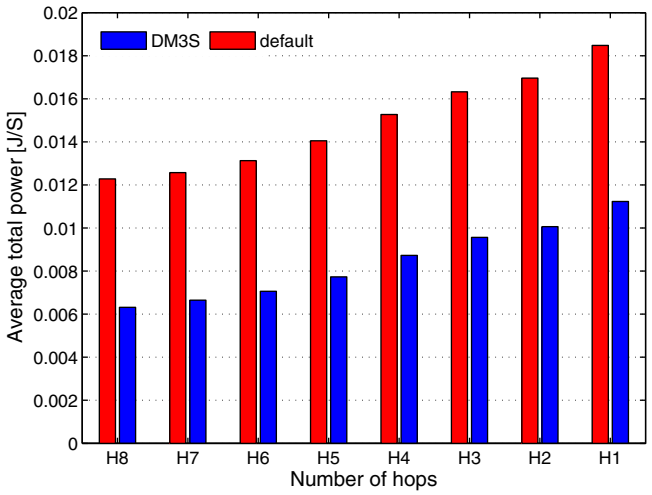


Fig. 13. Average energy-consumption between default mode and DM3S mode for 1 s rate.

Author's personal copy

6. Related work

In this section we discuss related work in the area of low-power MAC protocols for multi-hop wireless mesh networks. One of the main concerns of low-power MAC protocols is to switch the radio into sleep mode as much as possible, otherwise energy would be wasted. For low traffic scenarios the main factor contributing to the energy dissipation is idle listening (nodes listening on the channel in expectation of incoming frames). Other factors contributing to MAC energy consumption are: collisions, overhearing, control frames such as clock synchronization and management frames, and other protocol overheads. Energy-aware sensor network MAC protocols may be broadly classified into three main categories: asynchronous contention-based, synchronous contention-based, and schedule-based. An important example of asynchronous contention-based protocols are low-power listening protocols such as B-MAC [27] and WiseMAC [6]. These protocols are asynchronous, i.e. there is no need for nodes to coordinate their wakeup cycles and therefore there is no need for clock synchronization. Each node periodically wakes up and checks the channel activity for short time without receiving any data. If the channel is idle it goes to sleep, otherwise it stays awake to receive the frame. To rendezvous with receivers, senders send a long preamble before the actual message (longer than the checking interval).

In synchronous contention-based access protocols such as S-MAC [38] and T-MAC [34], nodes sleep and wake up in a synchronized fashion, and use a contention-based access protocol to transmit data in the awake periods. Time is organized into cycles of equal size. Each cycle is divided into two time intervals. In the first time interval nodes can exchange synchronization information. In the second interval nodes may send or receive, using a CSMA approach with RTS-CTS signaling. A general problem shared by all such synchronized protocols is that communication is grouped at the beginning of each slot, raising the chances of collisions and limiting their dynamic range to low traffic rates only.

Our work is most closely related to schedule-based access protocols, most of which are variations of TDMA. In TDMA protocols time is sub-divided into time slots, and time slots are exclusively allocated to one specific pair of nodes, a transmitting and a receiving node. In these protocols slot assignment algorithms and tight clock synchronization are of great concern. LMAC [35] uses a simple random slot assignment algorithm that ensures that nodes at 2-hop distance do not use the same slot number. It assumes a global time synchronization. Similar to LMAC, TRAMA [28] uses a distributed election scheme to determine particular time slots, however it uses more complicated policies that take traffic load into account, and which require relatively large amounts of memory for maintaining scheduling information among neighbors. A state-of-the-art solution for TDMA-based systems is the WHART standard. As explained in Section 2, WHART uses a centralized scheduling mechanism. WHART is used for example in energy, industrial and agriculture monitoring applications [4,30]. Major vendors such as ABB, Emerson Process Management Systems, Siemens and Yokogawa are producing WHART products and services for real-world applications. To the best of our knowledge, this study is the first attempt of analyzing the WHART energy consumption characteristics. We identify the factors contributing most to the overall network energy consumption. There are other standards targeting the same application domains (industrial automation, process automation) as WHART does, most notably the ZigBee [31] and ISA 100 [15] standards. Most of the energy consumption models proposed in the literature are based on the ZigBee standard [37,19], which in turn uses the IEEE 802.15.4 MAC protocol. In ZigBee there is no frequency diversity and the entire network operates in the same static channel. Also there is no path diversity, and a new path from source to destination has to be set up whenever the a link on the old path is broken. This increases both delay and overhead and leads to an increase in energy consumption. Therefore, energy consumption models proposed for ZigBee can not be applied to WHART.

7. Conclusions and future research

A state-of-the-art solution for TDMA based system is the WHART standard. WHART is one of the first wireless communication standards specifically designed for process automation applications. The standard has been finalized in 2007, and at the beginning of 2010 it has been ratified as an IEC standard. In this paper we have analyzed the energy consumption of the WHART protocol when used with a popular transceiver. The main contributions were:

(1) We have performed a sensitivity analysis using the response surface methodology to obtain some insights on how the overall energy consumption breaks down into different factors. By identifying the factors contributing most to the overall network energy consumption, one can obtain useful insights on where to start with any effort geared towards saving energy.
(2) We have looked at two different strategies for exploiting the sleep modes of the CC2420 transceiver and have highlighted that significant savings can be achieved with only moderate increases in run-time complexity.
(3) We have evaluated the impact of synchronization and management slots on the performance of WHART TDMA protocol.

Future work will focus on the further improvement of WHART TDMA protocol itself and of TDMA scheduling methods. One particularly interesting area is the design of TDMA scheduling algorithms which explicitly take the presence of a local sleep scheduling algorithm and multiple sleep states into account by constructing schedules in which the slots of individual nodes have larger separations in time, so as to allow them to enter deeper sleep modes. We also plan to further explore energy consumption optimizations within the WHART protocol.

Acknowledgments

We wish to thank Prof. Adam Wolisz from the Technical University of Berlin for fruitful and insightful discussions during various stages of manuscript preparation. We also thank the anonymous reviewers for their helpful and constructive comments.

References

[1] M. Adler, R.K. Sitaraman, A.L. Rosenberg, W. Unger, Scheduling time-constrained communication in linear networks 1998, in: Proc. ACM Symposium on Parallel Algorithms and Architectures (SPAA), 1998, pp. 269–278.
[2] L. Benini, A. Bogliolo, G.D. Micheli. A survey of design techniques for system-level dynamic power management, in: Proc. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 2000, vol. 8, no. 3, NJ, USA, June 2000, pp. 299–316.
[3] Chipcon, CC2420 2.4 GHz IEEE 802.15.4/ZigBee-ready RF Transceiver, 2004. Available from: <http://www.chipcon.com>.
[4] L. Doherty, J. Simon, T. Watteyne, Wireless sensor network challenges and solutions, MICROWAVE Journal (2012) 22–34. Milpitas, CA, USA, August 2012.
[5] Dust Networks, Wirelesshart technical data sheet, White paper, Dust Networks, September 2007.
[6] A. El-Hoiydi, J.-D. Decotignie, C. Enz, E.L. Roux, Poster abstract: Wisemac, an ultra low power mac protocol for the wisenet wireless sensor network, in: Proc. ACM SenSys 03, Los Angeles, California, November 2003, Poster Abstract.

Author's personal copy

[7] J. Elson, D. Estrin, Random, ephemeral transaction identifiers in dynamic sensor networks, in: Proc. 21st International Conference on Distributed, Computing Systems (ICDCS-21), April 2001.
[8] J. Elson, K. Romer, Wireless sensor networks: a new regime for time synchronization, in: Proc. HotNets-I First Workshop on Hot Topics In Networks, New Jersey, USA, October 28–29, 2002.
[9] S.D. Glaser, Some real-world applications of wireless sensor nodes, in: Proc. (SPIE) Symposium on Smart Structures and Materials/NDE 2004, San Diego, California, March 2004.
[10] HART Communication Foundation, HART Communication Protocol Specification, HCF SPEC 13 Revision 7.1, 05 June, 2008.
[11] HART Communication Foundation, Network Management Specification, HCF SPEC 085 Revision 1.1, 30 May, 2008.
[12] HART Communication Foundation, TDMA Data Link Layer Specification, HCF SPEC 075 Revision 1.1, 17 May, 2008.
[13] HART Communication Foundation, WirelessHART Device Specification, HCF SPEC 290 Revision 1.1, 22 May, 2008.
[14] C.-H. Hwang, A.C.-H. WU, A predictive system shutdown method for energy saving of event-driven computaion, in: Proc. ACM Transactions on Design Automation of Electronic Systems 2000, vol. 5, New York, USA, April 2000, pp. 226–241.
[15] ISA, ISA 100: wireless system for automation. Available from: <http://isa.zigbee.org>.
[16] O. Khader, A. Willig, A. Wolisz, A simulation model for the performance evaluation of wirelesshart tdma protocol, Technical report, Telecommunication Networks Group, Technical UniversityBerlin, TKN Technical Report Series TKN-11-001, Berlin, Germany, May 2011.
[17] O. Khader, A. Willig, A. Wolisz, WirelessHART TDMA protocol performance evaluation using response surface methodology, in: Proc. 6th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), Barcelona, Spain, October 2011.
[18] O. Khader, A. Willig, A. Wolisz, Analyzing the energy consumption of wirelesshart tdma protocol, Technical report, Telecommunication Networks Group, Technical UniversityBerlin, TKN Technical Report Series TKN-12-00x, Berlin, Germany, April 2012.
[19] M. Kohvakka, M. Kuorilehto, M. Hnnikinen, T.D. Hmlinen, Performance analysis of ieee 802.15.4 and zigbee for large-scale wireless sensor network applications, in: 3rd ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Aensor and Ubiquitous Networks, 2006, New York, NY, USA, October 2006, pp. 48–57.
[20] K. Langendoen, Medium access control in wireless sensor networks, in: H. Wu, Y. Pan (Eds.), Medium Access Control in Wireless Networks, Nova Science Publishers, May, 2008.
[21] LAN/MAN Standards Committee of the IEEE Computer Society, IEEE STD 802.15.4-2006, Wireless Medium Access Contril (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Task Group, 2006. Available from: <http://www.ieee802.org/15/pub/TG4.html>.
[22] B.K.M. Zuniga, Analyzing the transitional region in low power wireless links, in: First IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON), Santa Clara, CA, October 2004.
[23] MSP430, Texas Instruments MSP430x1xx MicroController, 2006. Available from: <http://www.ti.com>.
[24] R. Myers, D. Montgomery, Response Surface Methodology, John Wiley and Sons, Inc., 2002.
[25] S. Nath, P.B. Gibbons, S. Seshan, Z. Anderson, Synopsis diffusion for robust aggregation in sensor networks, in: Proc. ACM Transactions on Sensor Networks, 2008, vol. 2, March 2008.
[26] NICTA, The Castalia simulator for Wireless Sensor Networks. Available from: <http://castalia.npc.nicta.com.au>.
[27] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proc. 2nd International Conference on Embedded Networked Sensor Systems (ACM SenSys), Baltimore, MD, November 2004.
[28] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, Energy-efficient, collision-free medium access control for wireless sensor networks, in: Proc. ACM SenSys 03, Los Angeles, California, November 2003.
[29] R.Y. Rubinstein, B. Melamed, Modern Simulation and Modeling, Wiley Interscience, New York, 1998.
[30] J. Song, S. Han, X. Zhu, A. Mok, D. Chen, M. Nixon, Demo abstract: a complete WirelessHART network, in: Proc. 6th ACM Conference on Embedded Networked Sensor Systems (SenSys 2008), Demonstration session, Raleigh, North Carolina, November 2008.
[31] Specification October 2007, ZigBee Alliance. Available from: <http://www.zigbee.org>.
[32] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, D. Culler, An analysis of a large scale habitat monitoring application, in: Proc. ASM SenSys '04, 2004, pp. 214–226.
[33] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, W. Hong, A macroscope in the redwoods, in: Proc. ACM SenSys '05, 2005, pp. 51–63.
[34] T. van, D.K. Langendoen, An adaptive energy-efficient mac protocol for wireless sensor networks, in: Proc. ACM SenSys '03', Los Angeles, California, USA, November 2003.
[35] L. van Hoesel, P. Havinga, A lightweight medium access protocol for wireless sensor networks, in: Proc. INSS, 2004, 2004.
[36] A. Varga, OMNeT++ Discrete Event Simulation System. Available from: <http://www.omnetpp.org>.
[37] Q. Wang, M. Hempstead, W. Yang, A realistic power consumption model for wireless sensor network devices, in: Sensor and Ad Hoc Communications and Networks, 2006, Reston, VA, USA, September 2006, pp. 286–295.
[38] W. Ye, J. Heidemann, D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in: Proc. INFOCOM 2002, New York, June 2002, IEEE.