Duty-Cycle Optimization for IEEE 802.15.4 Wireless Sensor Networks

PANGUN PARK, University of California, Berkeley
SINEM COLERI ERGEN, Koc University
CARLO FISCHIONE, KTH, Royal Institute of Technology
ALBERTO SANGIOVANNI-VINCENTELLI, University of California, Berkeley

Most applications of wireless sensor networks require reliable and timely data communication with maximum possible network lifetime under low traffic regime. These requirements are very critical especially for the stability of wireless sensor and actuator networks. Designing a protocol that satisfies these requirements in a network consisting of sensor nodes with traffic pattern and location varying over time and space is a challenging task. We propose an adaptive optimal duty-cycle algorithm running on top of the IEEE 802.15.4 medium access control to minimize power consumption while meeting the reliability and delay requirements. Such a problem is complicated because simple and accurate models of the effects of the duty cycle on reliability, delay, and power consumption are not available. Moreover, the scarce computational resources of the devices and the lack of prior information about the topology make it impossible to compute the optimal parameters of the protocols. Based on an experimental implementation, we propose simple experimental models to expose the dependency of reliability, delay, and power consumption on the duty cycle at the node and validate it through extensive experiments. The coefficients of the experimental-based models can be easily computed on existing IEEE 802.15.4 hardware platforms by introducing a learning phase without any explicit information about data traffic, network topology, and medium access control parameters. The experimental-based model is then used to derive a distributed adaptive algorithm for minimizing the power consumption while meeting the reliability and delay requirements in the packet transmission. The algorithm is easily implementable on top of the IEEE 802.15.4 medium access control without any modifications of the protocol. An experimental implementation of the distributed adaptive algorithm on a test bed with off-the-shelf wireless sensor devices is presented. The experimental performance of the algorithms is compared to the existing solutions from the literature. The experimental results show that the experimental-based model is accurate and that the proposed adaptive algorithm attains the optimal value of the duty cycle, maximizing the lifetime of the network while meeting the reliability and delay constraints under both stationary and transient conditions. Specifically, even if the number of devices and their traffic configuration change sharply, the proposed adaptive algorithm allows the network to operate close to its optimal value. Furthermore, for Poisson arrivals, the duty-cycle protocol is modeled as a finite capacity queuing system in a star network. This simple analytical model provides insights into the performance metrics, including the reliability, average delay, and average power consumption of the duty-cycle protocol.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Performance, Standardization, Experimentation, Theory

Additional Key Words and Phrases: Wireless sensor networks, IEEE 802.14.5, MAC, duty-cycle, optimization

---

P. Park is supported in part by the NSF under CPS: ActionWebs (CNS-931843). S. C. Ergen is supported by Marie Curie Reintegration Grant IVWSN, PIRG06-GA-2009-256441. C. Fischione is supported by the Swedish Research Council, the EU STREP project HydroBioNets, and the EU NoE Hycon2.
Authors' addresses: P. Park (corresponding author) and A. Sangiovanni-Vincentelli, EECS Department, University of California at Berkeley, Berkeley, CA; email: {pgpark,alberto}@eecs.berkeley.edu. S. C. Ergen, Electrical and Electronics Engineering, Koc University, Istanbul, Turkey; email: sergen@ku.edu.tr. C. Fischione, Automatic Control Lab, ACCESS Linnaeus Center and Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden; email: carlofi@kth.se.

DOI: http://dx.doi.org/10.1145/2529979

ACM Reference Format:
Park, P., Ergen, S. C., Fischione, C., Sangiovanni-Vincentelli, A. 2013. Duty-cycle optimization for IEEE 802.15.4 wireless sensor networks. ACM Trans. Sensor Netw. 10, 1, Article 12 (November 2013), 32 pages.
DOI: http://dx.doi.org/10.1145/2529979

1. INTRODUCTION

Many applications using wireless sensor networks (WSNs) require a certain degree of the probability of successful packet reception (reliability) and timely data communication to a collection center under low traffic regime. These requirements are critical particularly for the stability of the WSN-based control and automation applications [Willig et al. 2005]. In these applications, if reliability and delay requirements are not met, the correct execution of control actions or decisions concerning the phenomena sensed may be severely compromised. Satisfying high reliability and low delay requirements, however, may demand significant power consumption. Maximizing reliability or minimizing delay is usually not an optimal design strategy: reliability and delay must be flexible design parameters that need to be adequate for the application requirements while minimizing the power consumption to ensure long lifetime of the network [Zhang et al. 2001]. Energy efficiency is critical for applications with battery-powered devices. The radio in WSNs consumes a considerable amount of energy, and listening to the radio channel consumes as much energy as receiving data. Idle listening should be minimized, since it does not contribute to the operation of the network, yet it may require a large amount of energy.

Several duty-cycle protocols have been proposed as an effective mechanism for reducing idle listening (see, e.g., GAF [Xu et al. 2001], SPAN [Chen et al. 2001], SMAC [Ye et al. 2004], and low power listening (LPL) [Hill and Culler 2002]). Such protocols are based on periodical cycling between a sleep and a listening state. A key parameter determining the duty cycle is the sleep time for a given listening time. The main advantage of duty cycling is that nodes do not require any additional hardware, such as a wake-up radio [Guo et al. 2001]. Even more importantly, it does not require complex control mechanisms, as in time division multiple access (TDMA) schemes, for discovering network topology, keeping the nodes synchronized [Coleri-Ergen and Varaiya 2006] running the schedules efficiently [Uysal-Biyikoglu et al. 2002]. Duty cycling is particularly appealing for dynamic networks where the locations of the sensor nodes and data traffic generated at each node are changing over time [Jurdak et al. 2010]. However, the intrinsic simplicity of the mechanism has the drawback of smaller energy saving potential as compared to the more complex solutions just listed, unless the duty cycling is adapted to the changes in data traffic and network topology.

Duty-cycling medium access control (MAC) protocols are of two types: synchronous and asynchronous. Synchronized protocols, such as SMAC [Ye et al. 2004], TMAC [Dam and Langendoen 2003], WiseMAC [El-Hoiydi and Decotignie 2004] and SyncWUF [Shi and Stromberg 2007], are based on time synchronization and scheduling among the neighbors to specify when the nodes are sleep and awake. Nodes periodically exchange packets for synchronization and operate the communication in common active/sleep schedules. Nodes must exchange scheduling information so that a node with packets to send can start transmitting a short time before its intended receiver wakes up. Because nodes have to maintain neighbors' schedules, synchronized protocols are not suitable for dynamic networks containing resource-limited nodes that are possibly heterogeneous and thus requiring high security. The required time synchronization introduces communication overhead and computation complexity, since the clocks of the nodes are subject to a clock drift. The frequency of the hardware oscillator in the clock runs vary unpredictably due to various environmental effects. Different hardware platforms might have different angular frequency of the hardware oscillator. Moreover, many WSN applications simultaneously need to share a common communication infrastructure. Exchanging the synchronization packets across applications is a challenging task. Furthermore, synchronized protocols are more insecure to Denial of Service attacks and eavesdropping in WSNs [Wood and Stankovic 2002].

Asynchronous protocols, on the other hand, are based on preamble sampling, which was first introduced as the well-known LPL [Hill and Culler 2002] and then followed by many protocols that have a similar concept, including BMAC [Polastre et al. 2004] and X-MAC [Anderson et al. 2006]. In this method, the receiver wakes up periodically to check whether there is a transmission, and the sender, instead of coordinating the neighbors' wake up times, sends a preamble that is long enough to ensure that the receiver wakes up during the preamble. Asynchronous protocols are more popular in practice, since the simple mechanism does not require either global synchronization nor topology knowledge [Bachir et al. 2010; Langendoen and Meier 2010]. Such a simple implementation allows for the coexistence of heterogeneous networks, enabling any subset of nodes to operate independently of the rest of the network and provide robust communication in highly mobile networks without relying on any topology knowledge. In this article, we focus on the asynchronous preamble sampling protocol.

Determining the duty cycle is a fundamental problem in the asynchronous preamble sampling protocols since it affects the delay, reliability, and energy consumption of the network. Lowering the duty cycle implies putting nodes in sleep mode for larger periods. While using a larger sleep time reduces the cost of idle listening at the receiver, it increases the transmission cost, as the transmitter uses a longer preamble. Hence, there is a trade-off between the receiving cost of idle listening and transmission cost of longer preamble. Furthermore, as the sleep time increases, the reliability, throughput, and delay significantly degrade due to the high contention in the medium with increasing traffic. The duty cycle should be determined by considering the trade-off between power consumption, reliability, and delay of the network based on the application requirements. Moreover, the duty cycle should not be fixed but be able to adjust to the time-varying or spatially nonuniform traffic loads.

We explicitly consider the random access mechanism of the unslotted IEEE 802.15.4 protocol to improve the reliability and delay performance of preamble sampling protocols. The IEEE 802.15.4 standard has received considerable attention as a low data rate and low power consumption protocol for WSN applications in industry, control, home automation, and healthcare [IEEE 2006]. It has been adopted with minor variations also by other protocols, such as ZigBee [Wheeler 2007] and ISA100 [ISA 2009]. We remark that the unslotted IEEE 802.15.4 protocol is not energy efficient, since there is no explicit mechanism for saving energy consumption. It is natural to combine the duty-cycle mechanism and the unslotted IEEE 802.15.4 protocol. However, it is not trivial to find the optimal duty cycle, because this optimal value depends upon several parameters, such as random access mechanism, traffic load, network topology, and hardware specifications, and needs to consider the reliability and delay requirements of applications while minimizing power consumption.

The goal of this article is to design an adaptive duty-cycle algorithm to achieve maximum lifetime while guaranteeing the reliability and delay constraints of the application. We focus on how to tune the duty cycle for IEEE 802.15.4 MAC instead of designing an entirely new asynchronous duty-cycle protocol. Solving a duty-cycling optimization problem requires a number of cost- and constraint-function evaluations. Unfortunately, the dependence of these functions on the design parameters is implicit and quite complicated [Fischione et al. 2009]. Consequently, solving the optimization problem online is out of the question if we use the full-fledged model given the limited computational resources of the nodes. The most important problem here is finding the tractable model of the optimization without significant loss of accuracy. Our work is inspired by the simple observation of the dependency of reliability and delay for different traffic loads of the network: without loss of generality, as the traffic load decreases, the linear factor of the reliability and delay dependence on the sleep time become more dominant than the nonlinear factor. The original contributions of this article are as follows.

(1) We demonstrate the existence of a linear relation between reliability, delay of the packets, and sleep time, and a quadratic relation between power consumption and sleep time for a given listening time under the low traffic regime. The effect of listening time on reliability and delay is negligible as we increase listening time above a certain value. A simple method can estimate the coefficients of these experimental-based models without requiring a high computational load.
(2) We propose an adaptive optimal duty-cycle (AODC) algorithm for unslotted IEEE 80215.4 standard to minimize power consumption while meeting reliability and delay requirements. The proposed algorithm explicitly considers the random access mechanism of the standard.
(3) The proposed AODC algorithm is implemented on a test bed using TelosB sensors.[1] Experimental results show that this algorithm meets reliability and delay requirements while achieving high power efficiency under both stationary and transient conditions of the network.

The rest of the article is organized as follows: Section 2 gives an overview of existing studies. Section 3 presents the system model. In Section 4, the optimization problem is formulated, and the challenges in solving this problem are stated. Section 5 gives the queueing model to calculate the approximation of the energy consumption, delay, and reliability. Section 6 describes a simple experimental-based model and its validation through extensive experiments. In Section 7, the solution of the optimization problem based on the experimental-based models of energy consumption, delay, and reliability is presented, and the adaptive algorithm to implement the solution is described. Numerical results achieved during stationary and transient conditions are reported in Section 8. Finally, Section 9 concludes the article.

2. RELATED WORK

B-MAC [Polastre et al. 2004] is an asynchronous preamble sampling protocol extending LPL technique by a user-controlled sleep interval. Each node independently repeats a sleep/active cycle without negotiating on the schedules. When a transmitter sends a data packet, it sends a preamble long enough to cover one complete sleep interval, which assures that the receiver can detect the signal and eventually the start symbol, followed by the data message. The X-MAC [Anderson et al. 2006] is a refinement of B-MAC for packet-based radios. The transmitter sends a packet strobe instead of sending a long wake-up preamble of B-MAC. Once the node receives the right packet strobe, it replies with an ACK. Then, the data message exchange takes place immediately. This ACK mechanism in X-MAC reduces the average preamble transmission time and so the time, energy, and overhead to transfer a data packet, since the entire preamble does not need to be sent if the receiver was already awake. Therefore, we integrate X-MAC with the unslotted IEEE 802.15.4 protocol in this article. X-MAC also includes a lookup table to adapt the duty-cycle of the nodes based on the traffic load. However, the proposed solution is suboptimal, since the random access mechanism is not considered in the optimization problem. Moreover, no delay or reliability constraint on packet delivery is considered, which means that the energy minimization proposed by X-MAC does not guarantee any timely successful packet delivery.

[1]Crossbow TelosB device. `http://www.xbow.com/`.

The idea of adaptive duty cycling of preamble sampling protocol is presented in Jurdak et al. [2010], where the authors use the energy consumption of each node in the routing decision of the cross-layer solution. Each node determines the preferred parent in the routing tree based on the routing cost that is a function of the ratio of the duty cycles of neighbors to the average duty cycle in the neighborhood. The duty cycle is then chosen proportional to the expected number of packets to transmit. This model however does not consider the reliability and delay requirements nor minimizes the power consumption.

Park et al. [2009] derive the energy consumption of a node as a function of the duty cycle. This model is then applied to formulate two optimization problems—one minimizing the total energy consumption and the other maximizing the network lifetime. These problems are solved by using an iterative algorithm that requires the global topology information. The analytical model of the energy consumption, however, does not take the collision and contention in sending a packet, the random access mechanism, the packet copy delay, and the delay to tune the transceiver into account. The proposed practical heuristic algorithms are based on the exchange of the information of the energy consumption of neighbors. The algorithms tune the duty cycle by following additive increase/additive decrease (AIAD) policy based on either the variation in the total energy consumption of the node itself and its neighbors or the comparison of its energy consumption with the maximum energy consumption of its neighbors. This study, however, does not consider either the delay or reliability. Furthermore, the proposed algorithms are analyzed through the simulation without any experimental validations.

A dynamic sleep time control approach for reducing control packet energy waste that uses available statistical network traffic information has been proposed [Ning and Cassandras 2010]. The authors propose two distinct approaches for dynamically computing the sleep time, depending on the objectives and constraints of the network. The first approach provides a dynamic sleep time policy that meets a specified average delay based on the packet waiting time. The second approach determines the optimal policy that minimizes total energy consumed. Both approaches require the interarrival time distribution of traffic loads. However, in practice, the network traffic information is not usually known in advance. Therefore, this article presents a quantile-based distribution approximation and learning algorithm to estimate a probability distribution. This approach is computationally demanding because each node needs to estimate the interarrival time distribution of traffic loads and solve an optimization problem using numerical methods. In addition, the control packet overhead increases since each node sends the interarrival time of traffic loads. Furthermore, this sleep time control algorithm does not clearly describe the mechanism when it deals with many-to-one communication. In particular, the reliability issue of this algorithm is critical for many-to-one communication.

Merlin and Heinzelman [2010] present two adaptive duty-cycle algorithms for meeting the target successful packet transmission rate while ensuring a longer lifetime of the network. The first algorithm, called asymmetric additive duty-cycle control (AADCC) [Merlin and Heinzelman 2010] is based on a linear increase/linear decrease of the duty-cycle depending on the comparison of the successfully received packet rate and its target value. Whenever five consecutive packets are successfully sent to the destination, the sleep time is increased by 0.1 s. Otherwise, each node decreases the sleep time by 0.25 s. The second algorithm, called dynamic duty-cycle control (DDCC), on the other hand aims to balance the reliability and energy consumption by using control theory. In DDCC, a simple control law is applied to adapt the sleep time for a deterministic noisy linear process representation of the network. Each node periodically updates the characteristics of the system model. The proposed algorithms are evaluated through

Matlab simulation without any implementation due to the difficulty in measuring the energy consumption and computation load. Even though the number of estimator coefficients is reduced, the computation load makes it hard to run the algorithms in sensor nodes. In multihop networks, the algorithm requires time synchronization along the routing path, which can be very difficult and is in contrast with the simplicity of asynchronous duty-cycle protocol. Furthermore, the algorithm does not guarantee by design a minimum energy consumption, a desired delay, and reliability in the packet delivery.

In Lai and Paschalidis [2008], the routing path from each node to a single gateway is selected for the duty-cycled WSNs. The selection of the optimal path is based on an optimization problem where the objective function is to minimize a weighted sum of the expected energy consumption and the characterization of the probability that the latency exceeds a certain threshold. By assuming the Markovian process on the sleeping schedules and the channel conditions, the expected energy consumption of transmitting a packet on any path to the gateway is computed. In addition, an upper bound for the latency probability on each path is derived. Two algorithms are proposed to solve this problem: a centralized algorithm based on convex polynomial underestimators, and a simulated annealing technique that can be implemented in a distributed fashion. However, the proposed algorithms are evaluated through simulations without any implementation due to the high computation complexity in practice.

Some analytical studies of the synchronous duty-cycle algorithms are presented [Cohen and Kapchits 2009; Kim et al. 2010] by formulating different optimization problems. Cohen and Kapchits [2009], pose the problem of determining the optimal duty cycle for minimization of energy consumption for a maximum latency requirement. Kim et al. [2010] propose a similar problem to minimize the delay and maximize the network lifetime of event-driven traffic pattern. However, the memory requirement and computation complexity to run these algorithms are still high for resource-constrained sensor nodes. An asynchronous random sleeping (ARS) mechanism is investigated in Hua and Yum [2007], whereby sensors wake up randomly and independently of others in each time slot to maximize the stationary coverage probability. The ARS offers statistical sensing coverage; its performance can be characterized by the stationary coverage probability and the coverage periods. The closed-form expressions of the stationary coverage probability, the expected k-coverage periods, and the expected k-vulnerable periods are derived using the renewal process theory. The homogenous wakeup probability is computed by using an analytical result. However, in general, the wakeup probability is heterogeneous, depending on the location, platform, and different application requirements. These papers [Cohen and Kapchits 2009; Kim et al. 2010; Hua and Yum 2007] validate their algorithms via simulation without experiments.

Other studies [Hill and Culler 2002; Polastre et al. 2004; Anderson et al. 2006; Park et al. 2009] focus on the minimization of the energy consumption of the network. We remark that our target is to design an adaptive duty-cycle algorithm in order to minimize the power consumption while meeting the reliability and delay requirements. There is no adaptive duty-cycle protocol in the literature that considers all these aspects. In addition, these studies do not consider the random access mechanism of the unslotted IEEE 802.15.4 protocol, and the packets are assumed to be always successfully received without collisions. Contrary to previous studies [Hill and Culler 2002; Jurdak et al. 2010; Polastre et al. 2004; Anderson et al. 2006; Park et al. 2009; Ning and Cassandras 2010; Merlin and Heinzelman 2010; Cohen and Kapchits 2009; Kim et al. 2010], we consider the timely reliability rather than packet reception rate or the expected number of packets to transmit.

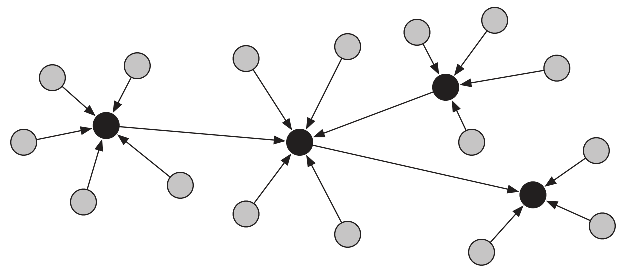


Fig. 1. Clustered network topology. The packets generated by the gray nodes are transmitted to the sink node depicted in the middle of each cluster.

3. SYSTEM MODEL

We assume that the nodes of the WSN are organized into clusters, as shown in Figure 1. Clustered network is an essential topology for a number of standardization groups [IEEE 2010] and commercial products [Wheeler 2007], such as asset tracking, process control, and building automation. Clustered network topology is supported in networks that require energy efficiency, since it allows local data aggregation and eliminates the disadvantages of unbalanced energy consumption in multihop routing and high energy consumption of transmitting directly to the base station [Heinzelman et al. 2000]. In a clustered topology, nodes organize themselves into clusters with a node acting as the cluster head. All non-cluster-head nodes transmit their data directly to the cluster head, while the cluster head receives data from all cluster members and transmits them to other cluster heads or a remote base station. We assume that the cluster heads, the cluster members, the routing path from each node to the base station, and the transmit power of the nodes fixed over the network are predetermined. Note that even such a simple topology presents highly challenging dynamics to model.

Throughout this article, we consider a probabilistic packet generation model rather than a periodic packet generation model, because the preamble sampling protocol is an asynchronous random access mechanism. We consider that the packet generation probability is uniformly distributed over the packet generation period. Given this source characteristic, the unslotted IEEE 802.15.4 is the natural MAC choice [IEEE 2006].

In preamble sampling protocols, the receiver wakes up periodically for a short time to sample the medium. When a sender has data, it transmits a series of short preamble packets, each containing the identifier of the target node, until it either receives an ACK packet from the receiver or a maximum wait time is exceeded. Note that the maximum wait time is greater than the sleep time. Following the transmission of each preamble packet, the transmitter waits for the timeout. If the receiver is the target and awake, it sends back an ACK. Upon reception of the ACK, the sender transmits the data packet to the destination then the receiver responds with an ACK packet.

Coherently with the IEEE 802.15.4 standard in the unslotted modality, we assume that the data and preamble packets are sent using random access, whereas the ACK frame is sent immediately upon reception of the preamble. For the packets that are sent using random access, the time duration between sending the packet to the MAC layer and over the physical link is random. In IEEE 802.15.4 standard, a node that sends a data frame shall wait for at most *macAckWaitDuration* for the corresponding ACK frame to be received. Hence, the timeout for receiving an ACK is equal to *macAckWaitDuration* of the standard. Consequently, the maximum listening time is the sum of the timeout and maximum back-off time of the random access. The time duration in random access may be much larger than the packet transmission time. In IEEE 802.15.4 radios with default parameter settings, the maximum back-off before

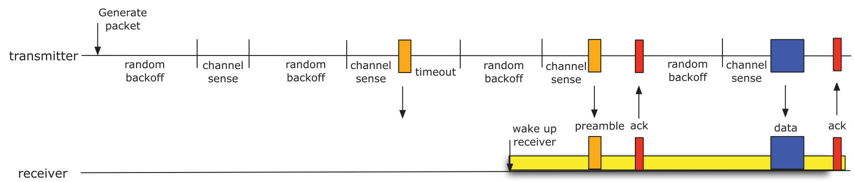


Fig. 2. Communication states between a transmitter and a receiver. A random number of preambles are sent before that one falls in the listening period of the receiver. Afterwards, the receiver sends an ACK. When the transmitter hears the ACK, the data packet is sent.

packet transmission is 27.4 ms, whereas the transmission time of a 56 byte packet is 1.79 ms, at 250 kbps. The amount of random access, which depends on the data traffic, network topology, and the parameters of the MAC protocol, should therefore be included in the power minimization problem, since random access (1) determines the time interval between the transmissions of two consecutive preamble packets; (2) determines the minimum value of the listening time required for the reception of a preamble packet if there is any active transmission; (3) is affected by the sleep time, since increasing sleep time increases both the expected number of preambles in the network and the time duration spent in random access.

To illustrate the dependency of the random access on the data traffic, the network topology and the parameters of the MAC protocol, we briefly explain the random access mechanism in IEEE 802.15.4 protocol next.

3.1. IEEE 802.15.4 Unslotted CSMA/CA Mechanism

In the unslotted IEEE 802.15.4 carrier sense multiple access with collision avoidance (CSMA/CA) mechanism, each node in the network has two variables: NB and BE. NB is the number of times the CSMA/CA algorithm has backed off while attempting the current transmission. NB is initialized to 0 before every new transmission. BE is the back-off exponent, which is related to how many back-off periods a node must wait before it attempts to assess the channel. The algorithm is implemented using units of time called back-off periods. The parameters that affect the random back-off are BE_{min}, BE_{max}, and NB_{max}, which correspond to the minimum and maximum of BE and the maximum of NB, respectively.

The unslotted CSMA/CA mechanism works as follows. NB and BE are initialized to 0 and BE_{min}, respectively (Step 1). The MAC layer delays for a random number of complete back-off periods in the range 0 to $2^{BE} - 1$ (Step 2) and then requests PHY to perform a clear channel assessment (CCA) (Step 3). If the channel is assessed to be busy (Step 4), the MAC sublayer increments both NB and BE by one, ensuring that BE is not more than BE_{max}. If the value of NB is less than or equal to NB_{max}, the CSMA/CA must return to Step 2. Otherwise, the CSMA/CA must terminate with a status of channel access failure. If the channel is assessed to be idle (Step 5), the MAC layer starts transmission immediately.

Figure 2 shows an example illustrating the communication states between a transmitter and a receiver. Upon the generation of the packet, the transmitter backs off for a random amount of time and senses the channel. Since the channel is busy, it backs off for another random amount of time and senses the channel. This time the channel is free, and the transmitter sends the preamble packet. However, since the receiver is in sleep state, it does not respond. Therefore, after waiting for the timeout time, the transmitter backs off for a random amount of time and senses the channel. Since the channel is free, it sends the preamble. This time, the receiver is awake, receives the preamble packet, and therefore responds with an ACK packet, which is then followed by the transmission of the data packet. When the node receives the data packet, it responds with an ACK packet.

The expected number of random back-offs is a function of the busy channel probability during channel sensing states, which depends on the channel traffic. Channel traffic, on the other hand, depends on the data traffic, network topology, and duty cycling, since they determine the expected number of preamble packets. This complex interdependence is investigated in the following sections.

4. PROTOCOL OPTIMIZATION

The goal of our duty-cycle protocol is to find the optimal sleep time and listening time of each receiver node such that the overall power of the network is minimized under reliability and delay constraints. The formulation of the optimization problem is as follows.

$$\min_{T_l, T_s} \quad E(T_l, T_s) \tag{1a}$$

$$\text{s.t.} \quad R(T_l, T_s) \geq R_{\min}, \tag{1b}$$

$$D(T_l, T_s) \leq D_{\max}, \tag{1c}$$

where $E(T_l, T_s)$ is the expected power consumption of the network, which includes transmit, receive, listen and sleep power; T_l and T_s are the listening time and sleep time, respectively; $R(T_l, T_s)$ and $D(T_l, T_s)$ are the expected reliability and delay of the network, respectively, $R_{\min}$ and $D_{\max}$ are the minimum acceptable reliability and maximum acceptable delay, respectively. The objective function $E(T_l, T_s)$ is the sum of the expected power consumption of the receiver and transmitter capturing the trade-off between the receiving cost of idle listening and transmitting cost of preamble packets. This objective function can be extended for different definitions of network lifetime in the literature of WSNs, such as maximizing the network lifetime and thus minimizing the maximum power consumption of any node [Dietrich and Dressler 2009], but is out of scope of this article. The reliability is defined as the probability of successful packet reception, whereas the delay is defined as the time interval from the instant the packet is generated until the transmission is successful after receiving the corresponding ACK from the receiver. The objective function and constraints are given by statistical expectation over time. The solution of the optimization problem gives the optimal sleep and listening times of the nodes. This optimization problem should be solved when the nodes are first deployed and in case of changes in the network topology or application requirements.

The power consumption, reliability, and delay depend on both T_l and T_s. The increase in T_s or decrease in T_l reduces the idle listening and therefore the energy consumption at the receiver, whereas the increase in T_s brings the additional cost of higher energy consumption in the transmission of the preambles at the transmitter. The increase in T_s also causes more contention due to the higher number of preamble packet transmissions, whereas the decrease in T_l results in missing some of the preamble packets if a large amount of time is spent in the random access before their transmission, both decreasing the reliability. Furthermore, increasing T_s increases the delay over each hop, resulting in higher end-to-end delay in multihop networks. Hence, the network designers need to consider the trade-off between power consumption, reliability, and delay of the network.

The exact computation of the analytical expressions in the optimization problem is a challenging task, since the sleep time and listening time affect the reliability, delay, and power consumption, along with the traffic load and network topology. Furthermore, the traffic load, channel condition, MAC parameters, and network topology affect the total back-off time of the random access mechanism, which then determines the number of preambles together with the listening time and sleep time of the receiver. Accurate analytical models of the expectations in Problem (1) have been investigated in Fischione et al. [2009]. Unfortunately, in these analytical models, the relation among the decision variables is highly nonlinear, which would require the use of sophisticated optimization tools to solve Problem (1). In the next section, we will provide the approximate expressions for the reliability, delay, and power consumption based on queueing models.

5. QUEUEING MODELS

In preamble sampling protocols, each receiver independently repeats a listen/sleep cycle without any coordination. When a sender has data, it competes with other senders to successfully transmit the data packet. This simple mechanism naturally leads us to envision the duty-cycle algorithm as a server and the contending nodes as clients receiving service in first-come-first-serviced (FCFS) queues. Essentially, when there are N contending nodes, each node receives a fraction $1/N$ of the system bandwidth. Thus, the duty-cycling algorithm can be regarded as a processor sharing system. To proceed further, we assume that the packet arrival process is given by a Poisson process with mean rate λ. For homogeneous nodes and uniform Poisson traffic, the arrival process is approximated as a Poisson process with rate $\lambda_N = N\lambda$. We assume the service time is equal to the maximum deterministic time $T = T_l + T_s$. Note that the protocol description in Section 3 reveals that the service times are not exponentially distributed. The size of buffers is related to the maximum wait time to send a data packet. We denote the size of buffers by $B = T_w/T$, where T_w is the maximum wait time. Recall that when a sender has data, it transmits a series of short preamble packets, until it either receives an ACK packet from the receiver or a maximum wait time is exceeded. For convenience, we assume that $B \geq 1$ is an integer. Therefore, our system model with the duty-cycle algorithm takes the form of a M/D/1/B queueing system.

The M/D/1/B model is a finite capacity queueing system, with $B - 1$ places in the buffer. Data packets are generated according to a Poisson process at rate λ_N. They are serviced according to the FCFS discipline, and the service time of each packet is the same constant T. Packets which upon arrival see a full system are rejected and do not further influence the system. The finite waiting time acts as a regulator on the queue size with the utilization factor $\rho = \lambda_N T$. Brun and Garcia [2009] derive the exact analytical stationary solution of this queue for steady-state probability distribution, mean number of packets, and average waiting time. This simple model provides insights into the performance metrics, including the reliability, average delay, and average power consumption of the duty-cycle algorithm. Although this model captures the fundamental trade-off between power consumption, reliability, and delay, it does not consider the random access mechanism, physical link quality, and hardware uncertainty.

Let $\beta_k = \rho^k/k!\mathrm{e}^{-\rho}$ denote the probability of k arrivals during a packet service. The probability transition matrix of the embedded Markov chain, ¶, takes the following form.

$$\P = \begin{bmatrix} \beta_0 & \beta_1 & \cdots & \beta_{B-2} & 1 - \sum_{k=0}^{B-2} \beta_k \\ \beta_0 & \beta_1 & \cdots & \beta_{B-2} & 1 - \sum_{k=0}^{B-2} \beta_k \\ 0 & \beta_0 & \cdots & \beta_{B-3} & 1 - \sum_{k=0}^{B-3} \beta_k \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \beta_0 & 1 - \beta_0 \end{bmatrix}. \tag{2}$$

By using this stochastic matrix, it is possible to derive the stationary probability distribution $\boldsymbol{\pi} = (\pi_0, \ldots, \pi_B)$ of the number of packets in the M/D/1/B queue.

THEOREM 1. *The stationary probability distribution is*

$$\pi_0 = \frac{1}{1 + \rho b_{B-1}},$$
$$\pi_i = \frac{b_i - b_{i-1}}{1 + \rho b_{B-1}}, \quad i = 1, \ldots, B-1,$$
$$\pi_B = 1 - \frac{b_{B-1}}{1 + \rho b_{B-1}},$$

where the coefficients b_j are given by $b_0 = 1$ and

$$b_j = \sum_{k=0}^{j} \frac{(-1)^k}{k!}(j-k)^k \mathrm{e}^{\rho(j-k)} \rho^k, \quad \forall j \geq 1.$$

Packets are never fed back into the queue: they are either successfully transmitted and thus leave the system or they are dropped because the maximum wait time has been reached. Therefore, the approximated reliability is

$$\tilde{R}(T_l, T_s) = 1 - \pi_B. \tag{3}$$

Using Theorem 1, it is straightforward to derive the stationary mean waiting time of the M/D/1/B system. The average delay D is approximated as the sum of the stationary mean waiting time and the average service time. We approximate the average service time as $T/2$.

THEOREM 2. *The expected value of the approximated delay $\tilde{D}$ that includes both the stationary mean waiting time in the M/D/1/B queue and the average service time is*

$$\tilde{D}(T_l, T_s) = \left(B - \frac{1}{2} - \frac{\sum_{k=1}^{B-1} b_k - B}{\rho b_{B-1}} \right) T. \tag{4}$$

Now, we compute the average power consumption of the M/D/1/B system. The expected value of the approximated power consumption is

$$\tilde{E}(T_l, T_s) = \tilde{E}_{rx}(T_l, T_s) + \tilde{E}_{tx}(T_l, T_s), \tag{5}$$

where $\tilde{E}_{rx}(T_l, T_s)$ and $\tilde{E}_{tx}(T_l, T_s)$ are the expected value of the average power consumption to receive and send data packets, respectively. Nodes sleep and wake up periodically to save energy. Hence, the expected value of the approximated power consumption to receive data packets is

$$\tilde{E}_{rx} = \frac{P_l T_l + P_s T_s}{T_l + T_s}, \tag{6}$$

where P_l and P_s are the average power consumption in the listening and sleep states, respectively. The packet generation process follows a Poisson process with mean rate λ. By considering Theorem 2, the expected value of the approximated power consumption to send data packets is

$$\tilde{E}_{tx} = P_w \tilde{D} \lambda, \tag{7}$$

where P_w is the average power consumption during the transmission of the packet. This average power consumption P_w, including transmitting preambles and data packets, and listening during a timeout is

$$P_w = \frac{T_{\text{data}}}{\tilde{D}} P_{tx} + \left(\frac{\tilde{D} - T_{\text{data}}}{\tilde{D}} \right) \left(\frac{P_{tx} T_p + P_l T_{\text{wait}}}{T_p + T_{\text{wait}}} \right), \tag{8}$$

where P_{tx} is the transmit power consumption, T_{wait} is the timeout, and T_{data}, T_p are the packet length of data packets and preambles, respectively. Two terms $T_{\text{data}}/\tilde{D}$ and $(\tilde{D} - T_{\text{data}})/\tilde{D}$ present the weight factor to send data packets and preambles, respectively.

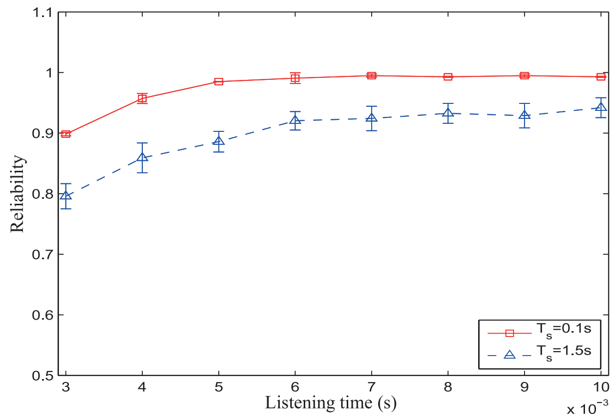
Similar to the exact analytical expressions derived in Fischione et al. [2009], the approximate expressions based on queueing models provide highly nonlinear relations among the decision variables, which would require the use of sophisticated optimization tools to solve Problem (1). Clearly, this is difficult or impossible to implement in resource-constrained sensor nodes. To overcome these problems, we propose an experiment-based model, where the objective function and the constraints of Problem (1) are approximated by quadratic and linear expressions, respectively, based on the observations from an extensive set of experiments. We will see in the next section that the regression coefficients representing these approximations can easily be computed adaptively in sensor nodes.

6. EXPERIMENT-BASED MODELS

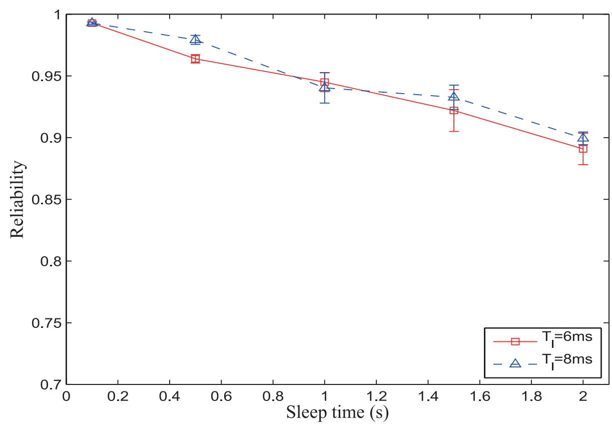
We present the analysis of the dependency of the total power consumption, reliability, and delay on the listening time and sleep time of the nodes. Simple experimental relations of the functions in Problem (1) are derived so that the problem can be quickly solved by the nodes. Accurate analytical models of the reliability, delay, and power consumption of the duty-cycle algorithm with the random access control have been presented in Fischione et al. [2009]. The analytical expressions are a function of listening time, sleep time, MAC parameters, and traffic load of a network. The drawback of these models is that they are highly nonlinear expressions that are difficult to use in practice. This motivates the experimental study of this article.

The duty-cycle algorithm of IEEE 802.15.4 protocol was implemented on a test bed using TelosB sensors running the Contiki operating system [Dunkels et al. 2004] based on the specifications of the IEEE 802.15.4 [IEEE 2006]. The implementation is available for download [Qin and Park 2011]. The values used for power consumption are those of the radio transceiver CC2420, which is featured by TelosB. The length of the preamble, ACK, and data packets are 24, 11, and 56 bytes for a data payload of 35 bytes, respectively. $BE_{\min} = 2$, $BE_{\max} = 3$, and $NB_{\max} = 2$ unless otherwise stated. The IEEE 802.15.4 defines one back-off as 20 symbols that correspond to $320\,\mu s$ for 2.45 GHz. Since the hardware timer available for TelosB is based on a 32,768 Hz clock, we use a back-off with duration of $305\,\mu s$ instead of $320\,\mu s$. The current drawn is 18.8 mA in receive mode, 17.4 mA when transmitting at 0 dBm, 20 μA in idle mode, and 1 μA in sleep mode. We set the maximum wait time $T_w = 2T$, where $T = T_l + T_s$. We consider a typical indoor environment with concrete walls. Each node is at a distance of around 5 m from the cluster head.

We consider a star topology consisting of a number of nodes up to 12 and packet generation periods varying from 10 s to 60 s. We let r be the average packet generation period by each node. Every node asynchronously generates a packet with probability p for each slot time unit S_b, where $p = \frac{S_b}{r}$ and $S_b = 0.125$ s. The experiment-based models are validated for different MAC parameters and transmission power. Linear regression is then used to compute the parameters of the experimental based models using the experimental results. We have chosen a linear regression because this allows us to model the relations with quadratic functions and yields a closed-form solution to Problem (1).



(a) Reliability as a function of listening time with different sleep times.



(b) Reliability as a function of the sleep time with different listening times.

Fig. 3. Reliability obtained by the experiments as a function of the listening time $T_l = 3, \ldots, 10$ ms and sleep time $T_s = 0.1, \ldots, 2$ s with the data generation period $1/\lambda = 30$ s for the number of transmitters $N = 8$. The vertical bars indicate the standard deviation as obtained out of five experimental runs of 90 min each.

6.1. Reliability Constraint

In this section, we provide an experiment-based model for the reliability constraint (1b) of Problem (1), where we recall that the reliability is defined as the probability of successful packet reception. Figures 3(a) and 3(b) show the reliability as obtained by the experiments with a data generation period of $1/\lambda = 30$ s and the number of transmitters $N = 8$ as a function of the listening time and sleep time, respectively. The vertical bars indicate the standard deviation as obtained out of 5 experimental runs of 90 min each.

Figure 3(a) shows that the reliability improves as the listening time increases. The improvement of the reliability however is negligible as the listening time increases above a certain value, that is, $T_l \geq 6$ ms. The reason being that this listening time value is able to accommodate the total time spent for random back-off before sending a preamble at the transmitter and in handling hardware interrupts at the receiver most of the time when the traffic load is low. Figure 3(b), on the other hand, shows that the reliability decreases linearly as the sleep time increases for $T_l \geq 6$ ms. As

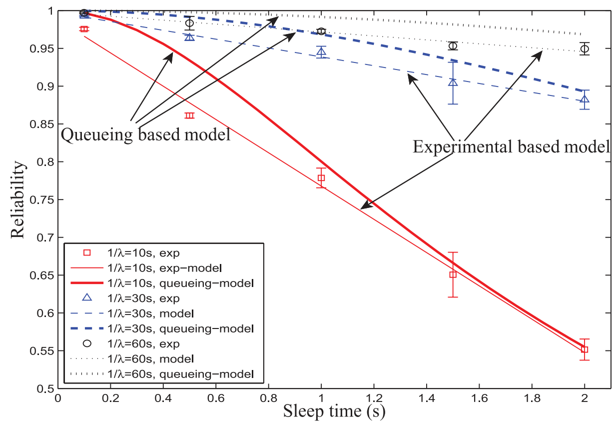


Fig. 4. Reliability obtained by the experiments, the experiment-based model given in Eq. (9), and the queueing-based model given in Eq. (3) as a function of the sleep time with different data generation periods: $1/\lambda = 10, 30, 60$ s for the number of transmitters $N = 8$. The vertical bars indicate the standard deviation as obtained out of five experimental runs of $180/\lambda$ s each (e.g., 30 min for $1/\lambda = 10$ s).

the sleep time increases, the expected number of preambles increases, which increases contention during listening time.

From the observation of the dominant effect of sleep time on the reliability, we propose the following simple experiment-based model for the reliability of Problem (1).

$$R(T_s) \approx i_R + r_R T_s\,, \tag{9}$$

for $T_l \geq 6$ ms, where i_R represents the intercept and r_R denotes the slope of the line. Without loss of generality, we assume that the coefficients of the reliability are $i_R > 0$, and $r_R < 0$, since the reliability decreases as the sleep time increases. The best value of T_l will be determined to be 6 ms in order to minimize power consumption once a similar observation is made for the delay in Section 6.2. We remark that the analytical model of the reliability proposed in Fischione et al. [2009] validates the dominant effect of the sleep time on the reliability.

Figure 4 shows the reliability as obtained by the experiments, the simple experiment-based model given in Eq. (9), and the queueing-based model given in Eq. (3) as a function of the sleep time $T_s = 0.1, \ldots, 2$ s, with data generation periods. $1/\lambda = 10, 30, 60$ s and the number of transmitters $N = 8$. The vertical bars indicate the standard deviation as obtained out of five experimental runs of $180/\lambda$ s each (e.g., 30 min for $1/\lambda = 10$ s). We increase the simulation time as the data generation period increases to achieve statistical significance. The experimental model is obtained by computing the coefficients of Eq. (9) via a simple linear regression. We observe that the queueing-based model follows the experiments while providing slightly better reliability than the experiments due to the main assumption of the perfect physical link and hardware.

The linear relation for reliability has been verified for various scenarios of the network with different network parameters, such as traffic load, number of nodes, channel condition, network topology, and MAC parameters, based on the experiments. To characterize quantitatively the error of the experiment-based model, we use the adjusted coefficient of determination because it gives the proportion of the variance of one variable that is predictable from the other variable [Cameron and Frank 1997]. Furthermore, it penalizes the statistics, as extra variables are included in the model.

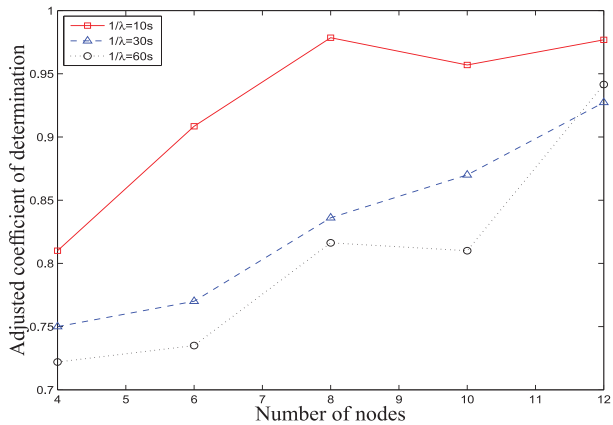


Fig. 5. Adjusted coefficient of determination of the reliability between the experimental results and the experiment-based models given in Eq. (9) as a function of different data generation periods $1/\lambda = 10, 30, 60$ s and number of nodes $N = 4, \ldots, 12$. The closer the adjusted coefficient of determination is to one, the better the experimental based model fits the data.

The adjusted coefficient of determination can be calculated as follows.

$$\overline{\gamma} = 1 - (1 - \gamma)\frac{M - 1}{M - q - 1},$$

where q is the total number of regressors in the model (not including the constant term), M is the sample size, and γ is the coefficient of determination defined as

$$\gamma = 1 - \frac{\sum_{i=1}^{M}(y_i - f_i)^2}{\sum_{i=1}^{M}(y_i - \overline{y})^2},$$

where y_i is an observed value, f_i is the associated modeled value, and $\overline{y} = \sum_{i=1}^{M} y_i/M$ is the mean of the observed data. It is a measure that allows us to determine how sure one can be in making predictions from a certain model. The better the experiment-based model fits the data in comparison to the simple average, the closer the adjusted coefficient of determination is to one.

Figure 5 shows the adjusted coefficient of determination between the experimental results and the experiment-based models given in Eq. (9) as a function of different numbers of nodes $N = 4, \ldots, 12$ and data generation periods $1/\lambda = 10, 30, 60$ s. The simple linear models for reliability are good approximations for different number of nodes and traffic load of the network. The adjusted coefficient of determination of the experiment-based model given in Eq. (9) shows that the sleep time is a dominant parameter for the reliability. Furthermore, we remark that the adjusted coefficient of determination increases as the traffic load increases in the lower traffic regime. Even though the adjusted coefficient of determination is small for the low traffic regime $1/\lambda = 60$ s, the absolute error is negligible in Figure 4. In a similar way, Figure 6 reports the adjusted coefficient of determination between the experimental results and the experiment-based models given in Eq. (9) as a function of different MAC parameters $NB_{\max} = 1, \ldots, 5$ and transmission power level TX $= 0, -1, -3$ dBm. In general, we observe that the experiment-based model gives lower adjusted coefficients of determination for low transmission power due to the hidden node problem. These comparisons show that the reliability is well approximated by the linear relation given in Eq. (9) for the application we are concerned with in this article. The effect of the listening time on the reliability is negligible compared to that of the sleep time.

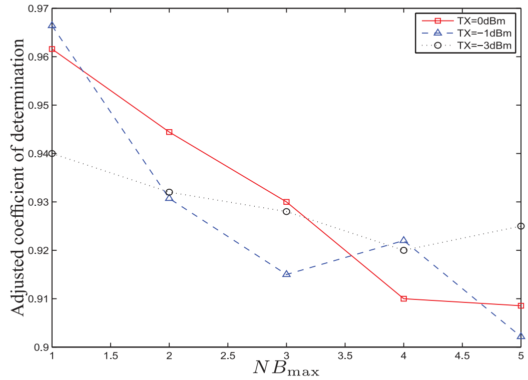


Fig. 6. Adjusted coefficient of determination of the reliability between the experimental results and the experiment-based models given in Eq. (9) as a function of different MAC parameters $NB_{\max} = 1, \ldots, 5$ and transmission power level TX $= 0, -1, -3$ dBm.

We use the experiment-based model of the reliability to find the solution of Problem (1) in Section 7. Now, we turn our attention to the delay constraint.

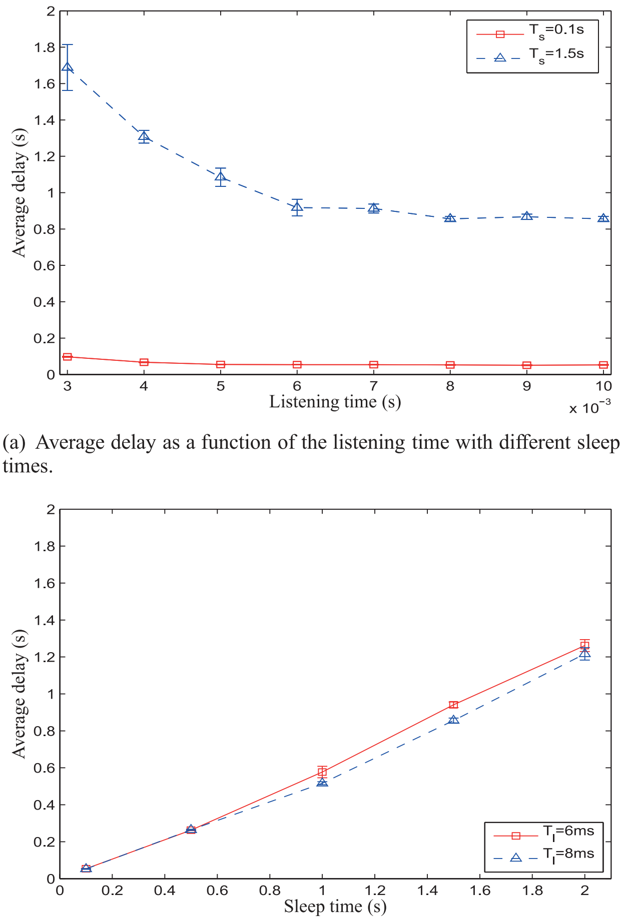
6.2. Delay Constraint

In this section, we provide an experiment-based model for the delay constraint (1c) of Problem (1). Recall that the delay for a successfully transmitted packet is defined as the time interval from the instant the packet is generated until the transmission is successful after receiving the corresponding ACK from the receiver. Figures 7(a) and 7(b) show the average delay as obtained by the experiments with data generation period $1/\lambda = 30$ s and number of transmitters $N = 8$ as a function of the listening time and sleep time, respectively. As the listening time decreases from $T_l = 6$ ms, the delay increases, as seen in Figure 7(a). The reason being that the receiver frequently misses the preambles when the listening time is too short, therefore the expected number of preambles to send a data packet, so the delay increases. Once the listening time is large enough, most of the packets are received in the first listening time, so the small value of the listening time compared to the sleep time results in negligible effect on the average delay for $T_l \geq 6$ ms. On the other hand, we observe a good linear relationship between delay and sleep time. Based on this observation, we propose the following simple experiment-based model for the average delay of Problem (1).

$$D(T_s) \approx i_D + r_D T_s \,, \tag{10}$$

for $T_l \geq 6$ ms, where i_D represents the intercept and r_D denotes the slope of the line. We assume that the coefficients of the average delay are $i_D > 0$ and $r_D > 0$, because the average delay increases as the sleep time increases. This relationship is valid only when $T_s \geq T_l$. However, this is not a limitation, because, to save power, sensors have to use duty cycles much smaller than 50%, which is compatible with $T_s \geq T_l$. The coefficients i_D and r_D are determined based on the experiments for different network parameters. A good linear relationship between the delay and sleep time is validated also through the analytical model of the delay proposed in Fischione et al. [2009].

Figure 8 compares the average delay of the experimental results, the experiment-based model given in Eq. (10) using linear regression, and the queueing-based model given in Eq. (4) as a function of the sleep time with different traffic generation periods $1/\lambda = 10, 30, 60$ s for the number of transmitters $N = 8$. The linear model given in Eq. (10) predicts well the experimental results. The average delay of the queueing-based model increases as the sleep time increases.



(a) Average delay as a function of the listening time with different sleep times.

(b) Average delay as a function of the sleep time with different listening times.

Fig. 7. Average delay obtained by the experiments as a function of the listening time $T_l = 3, \ldots, 10$ ms and sleep time $T_s = 0.1, \ldots, 2$ s with data generation period $1/\lambda = 30$ s for the number of transmitters $N = 8$.

In Figures 9 and 10, we validate the experiment-based models for delay given in Eq. (10) for different network parameters. Figure 9 shows the adjusted coefficient of determination between the experimental results and the experiment-based models given in Eq. (10) as a function of different data generation periods $1/\lambda = 10, 30, 60$ s and number of nodes $N = 4, \ldots, 12$. We observe high values of the adjusted coefficient of determination for various numbers of nodes and traffic loads. Similarly, Figure 10 reports the adjusted coefficient of determination between the experimental results and the experiment-based models given in Eq. (10) as a function of different MAC parameters $NB_{\max} = 1, \ldots, 5$ and transmission power level $\text{TX} = 0, -1, -3$ dBm. The effect of the listening time on the average delay is negligible, similar to its effect on the reliability. We conclude that the delay is well approximated by the linear model given in Eq. (10) for the application we are concerned with in this article.

We will use the experiment-based model of the average delay to find the solution of Problem (1) in Section 7. Now we investigate the power consumption.

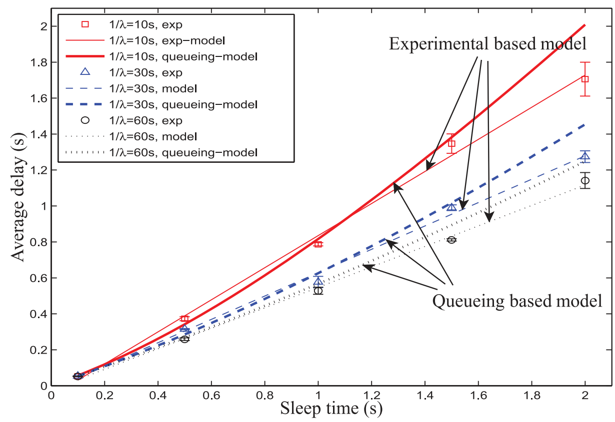


Fig. 8. Average delay obtained by the experiments, the experiment-based model given in Eq. (10), and the queueing-based model given in Eq. (4) as a function of the sleep time with different data generation periods $1/\lambda = 10, 30, 60\,\text{s}$ for the number of transmitters $N = 8$.

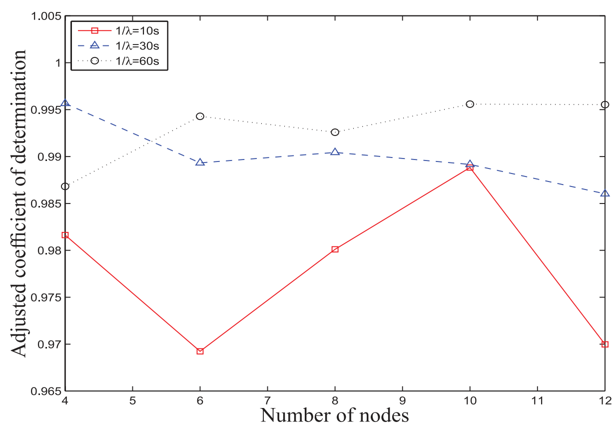


Fig. 9. Adjusted coefficient of determination of the average delay between the experimental results and the experiment-based models given in Eq. (10) as a function of different data generation periods $1/\lambda = 10, 30, 60\,\text{s}$ and number of nodes $N = 4, \ldots, 12$. Note the rather high values of the adjusted coefficient of determination on the y axis.

6.3. Power Consumption

In this section, we provide an experiment-based model for the average power consumption (1a) of Problem (1). We recall that the average power consumption is the sum of the expected power consumption for receiving and sending data packets. As we did for reliability and delay, it is possible to interpolate the power values obtained through the experiments as a function of the listening time and sleep time. Figures 11(a) and 11(b) show the average power consumption as obtained by the experiments with data generation period $1/\lambda = 30\,\text{s}$ and number of transmitters $N = 8$ as a function of the listening time and sleep time, respectively. Recall that our optimization problem is to minimize the power consumption while meeting the reliability and delay requirements in the packet transmission. Because the power consumption increases as the listening time increases for $T_l \geq 6\,\text{ms}$ in Figure 11(a), it is natural to reduce the listening time by considering both reliability and delay performance. We set the listening time to $T_l = 6\,\text{ms}$ because the reliability and delay significantly degrade for $T_l < 6\,\text{ms}$, and

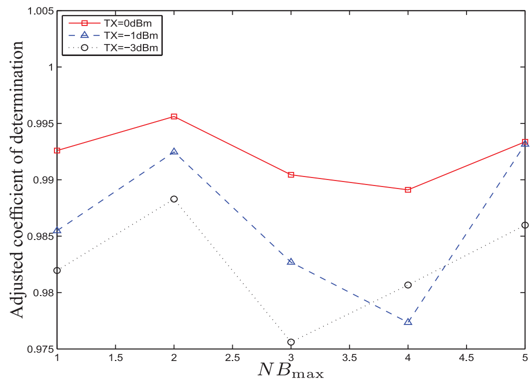


Fig. 10. Adjusted coefficient of determination of the average delay between the experimental results and the experiment-based models given in Eq. (10) as a function of different MAC parameters $NB_{\max} = 1, \ldots, 5$ and transmission power level TX $= 0, -1, -3$ dBm.

the improvement of reliability and delay are negligible for $T_l > 6$ ms, as observed in Figures 3(a) and 7(a). In Figure 11(b), we observe a trade-off between the receiving cost of idle listening and the transmission cost of preambles. While using a longer sleep time reduces the cost of idle listening at the receiver, it increases the transmission cost, as the transmitter sends more preambles with possible contention. There is optimal value for the sleep time beyond which nodes waste more power in transmission than they save in reception.

In order to derive simple experiment-based models, for a given T_l, we separate the average power consumption to receive and send data packets, $E_{rx}(T_s)$ and $E_{tx}(T_s)$, respectively. Such simple experiment-based models for $E_{rx}(T_s)$ and $E_{tx}(T_s)$ and the average total power consumption, $E(T_s)$, result in the following.

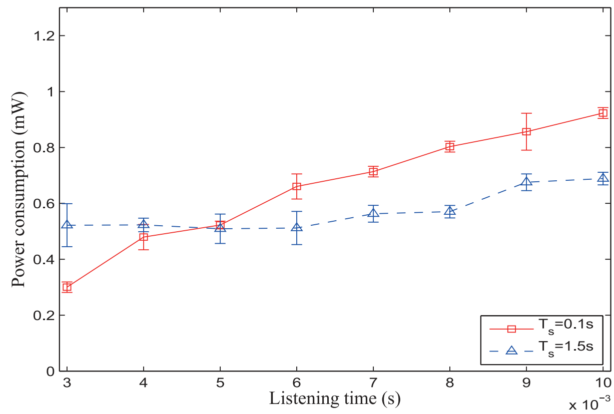
$$E_{rx}(T_s) \approx i_{E_{rx}} + \frac{g_E}{T_s}\,, \tag{11}$$

$$E_{tx}(T_s) \approx i_{E_{tx}} + r_E T_s\,, \tag{12}$$

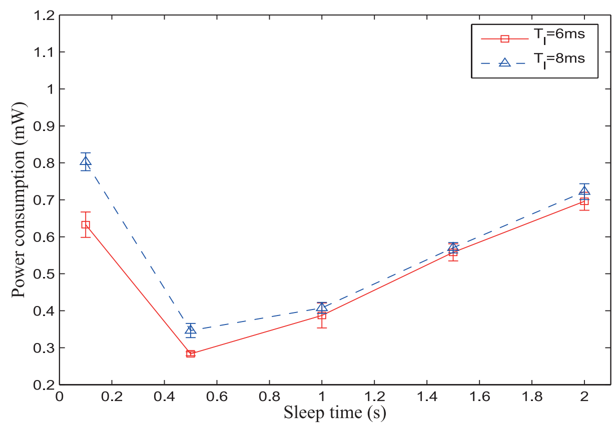
$$E(T_s) \approx i_E + \frac{g_E}{T_s} + r_E T_s\,. \tag{13}$$

where $i_E = i_{E_{rx}} + i_{E_{tx}}$ and the coefficients $i_{E_{rx}}, i_{E_{tx}}, r_E, g_E$ are determined based on the experiments using linear regression. We remark that the analytical model of the average power consumption proposed in Fischione et al. [2009] validates this relation.

Figures 12(a), 12(b), and 12(c) show the average power consumption to receive and send data packets as well as and the total power consumption respectively as obtained by the experiments, the experiment-based model given in Eqs. (11) and (12), the queueing-based model given in Eqs. (6) and (7), as a function of the sleep time with data generation periods $1/\lambda = 10, 30, 60$ s for $N = 8$ transmitters. The experiment-based model for power consumption follows well the experimental results. Note that the coefficient of the average power consumption to receive data packets is $g_E > 0$, since this average power consumption decreases as the sleep time increases, as shown in Figure 12(a). The coefficient of the average power consumption to send data packets $r_E > 0$, since this average power consumption increases as the sleep time increases, as shown in Figure 12(b). The average power consumption to receive and send data packets of the queueing-based model and resulting total power consumption follows the experimental results, as demonstrated in Figures 12(a), 12(b), and 12(c), respectively. In Figure 12(c), we clearly observe the trade-off between the receiving cost of



(a) Average power consumption as a function of the listening time with different sleep times.

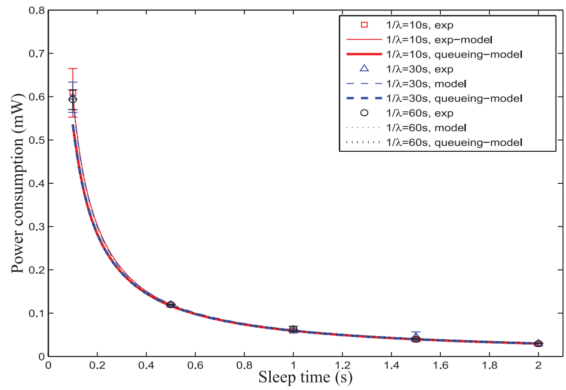


(b) Average power consumption as a function of the sleep time with different listening times.

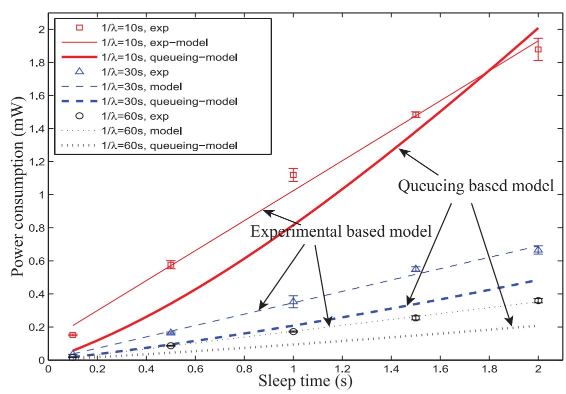
Fig. 11. Average power consumption obtained by the experiments as a function of the listening time $T_l = 3, \ldots, 10$ ms and sleep time $T_s = 0.1, \ldots, 2$ s with the data generation period $1/\lambda = 30$ s for the number of transmitters $N = 8$.

idle listening and transmission cost of preambles. Therefore, it is critical to determine the optimal sleep time to balance the average power consumption to receive and send data packets.

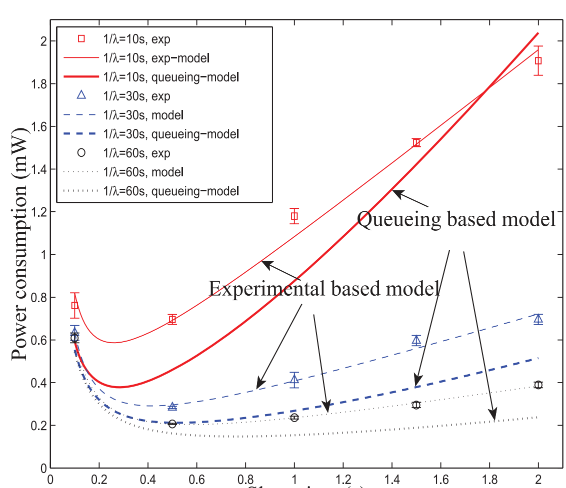
Further analysis of the average power consumption reveals that the approximation given in Eq. (13) is good for various network scenarios. Figure 13 shows the adjusted coefficient of determination between the experimental results and the experimental based models given in Eq. (13) as a function of different data generation periods $1/\lambda = 10, 30, 60$ s and number of nodes $N = 4, \ldots, 12$. The adjusted coefficient of determination of the experiment-based model given in Eq. (13) is high. Figure 14 reports the adjusted coefficient of determination between the experimental results and the experiment-based models given in Eq. (13) as a function of different MAC parameters $NB_{\max} = 1, \ldots, 5$ and transmission power level TX $= 0, -1, -3$ dBm. We observe that the adjusted coefficient of determination decreases as $NB_{\max}$ increases due to the increase in the random back-off time of the unslotted IEEE 802.15.4. These comparisons



(a) Average power consumption to receive data packets as a function of the sleep time.



(b) Average power consumption to transmit a data packet as a function of the sleep time.



(c) Average total power consumption as a function of the sleep time.

Fig. 12. Average power consumption obtained by the experiments, the experiment-based model given in Eqs. (11), (12), and (13), and the queueing-based model given in Eqs. (6), (7), and (5) as a function of the sleep time with packet generation periods $1/\lambda = 10, 30, 60\,s$ for number of transmitters $N = 8$.

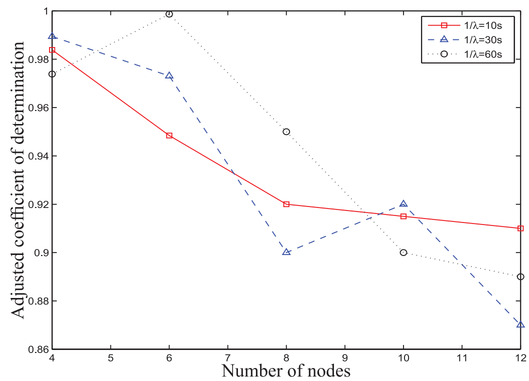


Fig. 13. Adjusted coefficient of determination of the average power consumption between the experimental results and the experiment-based models given in Eq. (13) as a function of different data generation periods $1/\lambda = 10, 30, 60$ s and number of nodes $N = 4, \ldots, 12$.

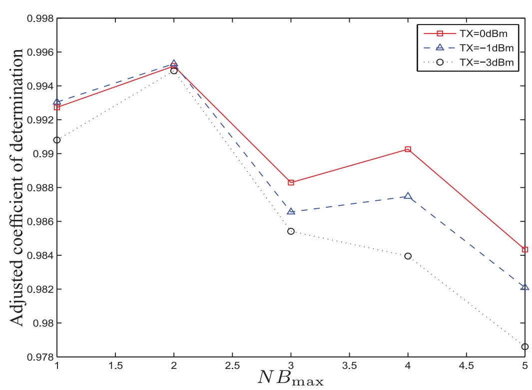


Fig. 14. Adjusted coefficient of determination of the average power consumption between the experimental results and the experiment-based models given in Eq. (13) as a function of different MAC parameters $NB_{\max} = 1, \ldots, 5$ and transmission power level TX $= 0, -1, -3$ dBm. Note the rather high values of the adjusted coefficient of determination on the y axis.

show that the average power consumption is well approximated by the model given in Eq. (13).

7. ADAPTIVE DISTRIBUTED ALGORITHM

In this section, we solve the optimization problem based on the experiment-based models derived in Section 6. Furthermore, we describe the AODC algorithm to implement in practice the optimal solution.

As stated in Section 6, we set the listening time to $T_l = 6$ ms and find the optimal value of sleep time. The reason being that the reliability and delay significantly degrade for $T_l < 6$ ms with negligible improvement for $T_l \geq 6$ ms, as shown in Figures 3 and 7, whereas the power consumption increases as the listening time increases.

By putting the experiment-based models of the reliability and delay constraints and power consumption given in Eqs. (9), (10), and (13), respectively, it is possible to

ALGORITHM 1: Pseudocode for the AODC Algorithm

Input: $T_0, C_{\max}, \alpha_R, \alpha_D, \alpha_E, \delta, R_{\min}, D_{\max}$
Output: T_s
begin
  $T_s \leftarrow T_0$;
  $E^* \leftarrow -\infty$;
  for *ever* **do**
    Update $\tilde{R}, \tilde{D}, \tilde{E}, \tilde{E}_{rx}, \tilde{E}_{tx}$ for duration T_{up};
    if $\tilde{R} < (1-\alpha_R)R_{\min}$ **or** $\tilde{D} > (1+\alpha_D)D_{\max}$ **or** $\tilde{E} > (1+\alpha_E)E^*$ **or** $\tilde{E} < (1-\alpha_E)E^*$ **then**
      $C \leftarrow C+1$;
      if $C > C_{\max}$ **then**
        // Learning phase.
        $\tilde{R}_1 \leftarrow \tilde{R}, \tilde{D}_1 \leftarrow \tilde{D}, \tilde{E}_{rx,1} \leftarrow \tilde{E}_{rx}, \tilde{E}_{tx,1} \leftarrow \tilde{E}_{tx}$;
        $T_s \leftarrow \delta T_s$;
        Update $\tilde{R}, \tilde{D}, \tilde{E}_{rx}, \tilde{E}_{tx}$ for duration T_{up};
        $\tilde{R}_2 \leftarrow \tilde{R}, \tilde{D}_2 \leftarrow \tilde{D}, \tilde{E}_{rx,2} \leftarrow \tilde{E}_{rx}, \tilde{E}_{tx,2} \leftarrow \tilde{E}_{tx}$;
        Update $\hat{i}_R, \hat{r}_R, \hat{i}_D, \hat{r}_D, \hat{i}_E, \hat{g}_E, \hat{r}_E$;
        // Optimization phase.
        $T_s \leftarrow \min\left(\sqrt{\frac{\hat{g}_E}{\hat{r}_E}}, \frac{R_{\min}-\hat{i}_R}{\hat{r}_R}, \frac{D_{\max}-\hat{i}_D}{\hat{r}_D}\right)$;
        $E^* \leftarrow \hat{i}_E + \frac{\hat{g}_E}{T_s} + \hat{r}_E T_s$;
        $C \leftarrow 0$;
      end
    else
      $C \leftarrow 0$;
    end
  end
end

reformulate Problem (1) as follows.

$$\min_{T_s} \quad i_E + \frac{g_E}{T_s} + r_E T_s \tag{14a}$$

$$\text{s.t.} \quad T_s \leq \frac{R_{\min} - i_R}{r_R}, \tag{14b}$$

$$T_s \leq \frac{D_{\max} - i_D}{r_D}. \tag{14c}$$

This problem is convex because the objective function is convex and the constraints are in standard linear form. The objective function is convex, since $\partial^2 U/\partial T_s^2 = 2g_E/T_s^3 > 0$, where U is the objective function and $g_E > 0$. The optimal solution can then be expressed in closed form after using standard Lagrangian methods as follows [Boyd and Vandenberghe 2004].

$$T_s^* = \min\left(\sqrt{\frac{g_E}{r_E}}, \frac{R_{\min} - i_R}{r_R}, \frac{D_{\max} - i_D}{r_D}\right). \tag{15}$$

In Eq. (15), the first term is derived by taking the derivative of the objective function with respect to T_s and setting it to 0, whereas the second and third terms are computed by using the reliability and delay constraints, respectively.

We propose the AODC algorithm described in Algorithm 1 at each receiver node. The main parameters of the algorithm and their default values are given in Table I. The

Table I. Main Symbols Used in Algorithm 1

Symbol	Meaning
T_0	Initial sleep time
$C_{\max}$	Threshold of consecutive infeasible sets to activate the optimization algorithm, default 10
α_R	Relaxation factor of $R_{\min}$, default 0.1
α_D	Relaxation factor of $D_{\max}$, default 0.1
α_E	Relaxation factor of E^*, default 0.1
δ	Update ratio of the sleep time, default 0.1
T_{up}	Update duration for estimating $\tilde{R}$, $\tilde{D}$, $\tilde{E}_{rx}$, $\tilde{E}_{tx}$, default wait time to get at least 100 samples

goal of the algorithm is that a receiver node finds the optimal sleep time that minimizes power consumption for given reliability and delay constraints, $R_{\min}$ and $D_{\max}$, respectively, based on the solution provided by Eq. (15). The node learns the coefficients of the mathematical models of Eqs. (9), (10), and (13) in an adaptive manner to the changes in the environment and network topology. The AODC algorithm therefore is composed of two phases: learning phase and optimization phase. The learning phase deals with learning the coefficients of the functions used in the optimization problem, which is then solved in the optimization phase. The learning phase is needed to avoid recording in a look-up table the coefficients of the model for each possible configuration of the network. The size of the table needed to keep this information is not manageable. Moreover, for every receiver node of the network, it is usually not possible to know the exact configuration of the neighbors and their traffic. The learning phase of our algorithm does not require any explicit information about the traffic load and topology of the network, thus minimizing the extra communication overhead throughout the network. We describe the running of the algorithm in the following.

The AODC algorithm requires that each receiver node estimates the reliability $\tilde{R}$, delay $\tilde{D}$, and power consumption $\tilde{E}$ from the neighbors upon reception of each packet for a duration T_{up} (line 5). Then, the node checks whether the desired reliability $R_{\min}$, delay $D_{\max}$, and power consumption E^* values, as requested by the application, are achieved within a certain accuracy. If the desired values $R_{\min}$, $D_{\max}$, and E^* are not met within certain factors for more than $C_{\max}$ times (lines 6–8), then the learning phase is activated (lines 9–13). Next we describe in detail how the estimation is performed.

The reliability is easy to estimate by using the sequence number of received data packets (line 5). To estimate the delay, each transmitter adds the delay between the packet generation time and the packet sending time to the payload of the data packet. However, this solution introduces an extra delay due the limited speed of the serial peripheral interface (SPI) bus and the internal delay of the operating system. Note that before sending a packet, the microcontroller copies the packet data into the transmit buffer of the radio transceiver over the SPI bus. Osterlind and Dunkels [2008] show that packet copying is a critical issue when forwarding a packet. Hence, the transmitter adds the delay information of the previous data packet into the payload of the current data packet. Furthermore, by recording the transitions among transmit, receive, idle, and sleep states, the transmitter is able to estimate its own average power consumption to receive and send packet packets. The transmitter then includes this information in the payload of the data packet.

When a receiver gets a data packet, it retrieves the packet delay and power consumption from the payload and estimates the reliability by tracking the sequence numbers for the corresponding neighbor. For the reliability and delay estimation, the receiver just finds the averages over the estimated values of each neighbor. For the power consumption, on the other hand, the receiver estimates its own average power consumption by recording its own state transitions and then averages together with the average power consumption of the neighbors. Note that each data packet includes the information required to estimate the reliability, average delay, and average power consumption so the algorithm does not require the transmission of extra control packets minimizing the protocol overhead. Because the number of measurements for estimating the reliability, delay and power consumption is small, the effect of measurement errors is critical for the accuracy of the experiment-based model. We use the sliding window method to smooth the performance measurement for a given sleep time.

The condition for checking whether the reliability and average delay requirements are met is specified as $\tilde{R} < (1 - \alpha_R)R_{\min}$ and $\tilde{D} > (1 + \alpha_D)D_{\max}$, respectively, where $0 < \alpha_R < 1$ and $0 < \alpha_D < 1$ (line 6). The optimality of power consumption, on the other hand, is checked by $\tilde{E} > (1 + \alpha_E)E^*$ and $\tilde{E} < (1 - \alpha_E)E^*$; recall that E^* is the expected optimal power consumption and $0 < \alpha_E < 1$. $E > (1 + \alpha_E)E^*$ can appear if new nodes enter the network or the link connectivity changes. On the other hand, $\tilde{E} < (1-\alpha_E)E^*$ can happen if nodes leave the network; hence the contention of random access mechanism and traffic load decreases while meeting the requirement $\tilde{R} > (1-\alpha_R)R_{\min}$ and $\tilde{D} < (1 + \alpha_D)D_{\max}$. In this case, since E^* is not the optimal value anymore, each node consumes more power than the actual optimal one, even though the reliability and delay meet the application requirement. The constraints are relaxed by introducing the factors α_R, α_D, and α_E to take into account the stochastic behavior. Each node keeps track of the number of times the requirements are not met (lines 7−8). If this number is greater than a threshold value, that is, $C > C_{\max}$, the node activates the learning phase (lines 9–13), which we describe next.

In the learning phase, each node estimates the power consumption $\tilde{E}$, reliability $\tilde{R}$, and delay $\tilde{D}$ for different sleep time T_s, then runs simple linear regression to compute the coefficients of the experiment-based model in Eqs. (9), (10), (13). In general, the linear regression gives better estimation as the number of the measurements increases. However, the higher the number of the measurements, the larger the memory requirement and computation load to run the linear regression. Therefore, each node uses the least number of measurements necessary to learn the coefficient of the experiment-based model in each step.

When the learning phase starts, the node saves the current reliability $\tilde{R}_1$, average delay $\tilde{D}_1$, and average power consumption $\tilde{E}_{rx,1}$, $\tilde{E}_{tx,1}$ (line 9). Then, the node reduces the sleep time T_{s_1} to $T_{s_2} = \delta T_{s_1}$, where $0 < \delta < 1$ (line 10). By doing so, each node improves the reliability and delay while learning the change of the network environment. After reducing the sleep time, the node measures the reliability $\tilde{R}_2$, average delay $\tilde{D}_2$, and average power consumption $\tilde{E}_{rx,2}$, $\tilde{E}_{tx,2}$ corresponding to the sleep time T_{s_2} (lines 11–12). Note that the reliability difference between $\tilde{R}_1$ and $\tilde{R}_2$ shows the effect of changing the sleep time from T_{s_1} to T_{s_2}. Then, the parameters of the experiment-based model are computed (line 13). For the reliability experiment-based model given in Eq. (9), the estimated parameters of the intercept and slope are

$$\hat{\imath}_R = \frac{1}{\delta - 1}(\delta\tilde{R}_1 - \tilde{R}_2),$$
$$\hat{r}_R = \frac{1}{(\delta - 1)T_{s_1}}(\tilde{R}_2 - \tilde{R}_1), \tag{16}$$

where $\tilde{R}_1$ and $\tilde{R}_2$ are the estimated reliability corresponding to the sleep times T_{s_1} and T_{s_2}, respectively. Note that the form of the linear regression of the reliability is the same as the one used for the delay constraint and the average power consumption to send a data packet, so equations similar to Eq. (16) allow us to estimate $\hat{\imath}_D, \hat{r}_D, \hat{r}_E$, and $\hat{\imath}_{E_{tx}}$. Similarly, we compute the coefficients of the experiment-based model of the average power consumption to receive data packets given in Eq. (11) by estimating the average power consumption for two different sleep times T_{s_1}, T_{s_2}. The learned coefficients are

$$\hat{i}_{E_{rx}} = \frac{\delta}{\delta - 1}\left(\tilde{E}_{rx,2} - \frac{\tilde{E}_{rx,1}}{\delta}\right),$$
$$\hat{g}_E = \frac{\delta\, T_{s_1}}{\delta - 1}\left(\tilde{E}_{rx,1} - \tilde{E}_{rx,2}\right), \tag{17}$$

where $\tilde{E}_{rx,1}$ and $\tilde{E}_{rx,2}$ are the estimated average power consumptions for receiving data packets corresponding to the sleep times T_{s_1} and T_{s_2}, respectively. The sliding window is initialized to estimate the reliability, delay, and power consumption. Otherwise, the convergence rate for estimating these parameters is very slow.

Once a receiver node learns the network environment by knowing the coefficients of the experiment-based model of Eqs. (9), (10), and (13), the optimization phase of the algorithm starts (lines 14–15). The node sets the sleep time to its optimal value T_s^* by using the solution derived in Eq. (15). If the problem is not feasible, it means that it is not possible to meet the reliability $R_{\min}$ and delay requirements $D_{\max}$ by tuning the sleep time. The application requirements must be relaxed so that feasibility is ensured and the problem can be solved.

The adaptive algorithm described so far assumed that all packet losses are due to the long sleep time. This assumption has allowed us to simplify the dependency of the reliability, delay, and power consumption on the duty cycle. However, in practice, links of IEEE 802.15.4 are bursty between bad and good delivery performance [Srinivasan et al. 2008]. If a node has a bad delivery link, reducing the sleep time does not improve the reliability and delay, but increases the power consumption. A node can avoid adjusting the parameters to a short burst length by keeping the length of the sliding window over which the averages for reliability, delay, and power consumption are taken long enough compared to the burst length.

8. EXPERIMENTAL RESULTS

In this section, we analyze the performance of the AODC algorithm for tuning the duty cycles under both stationary and transient conditions based on an extensive set of real-world experiments. In the stationary condition, the application requirements and network scenario are constant, whereas they vary over time in the transient case. The experimental setup was described in Section 6. As we presented in Section 7, each receiver node estimates the reliability, delay, and power consumption of the network to run the AODC algorithm. The sequence number of the IEEE 802.15.4 MAC header is used to estimate the reliability without extra overhead. Each transmitter adds the delay of the previous data packet into the payload of the current data packet. In addition, each node records the radio state transitions among transmit, receive, idle, and sleep state to estimate its own average power consumption for receiving and sending data packets and adds corresponding information into the payload. When a node receives ACK to the transmitted packet, it resets the number of state transitions regarding the power consumption. When each node receives a data packet, it first retrieves the information of sequence number, packet delay, and power consumption of neighbors. Then, it computes the average reliability, delay, and power consumption.

8.1. Protocol Behavior in Stationary Conditions

In this section, we analyze the performance metrics of the AODC algorithm under the stationary condition, namely, without changing the application requirements (i.e., $R_{\min}$ and $D_{\max}$) and network scenarios.

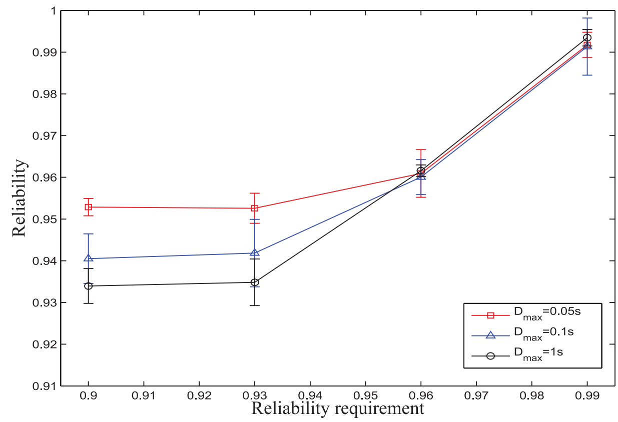


Fig. 15. Reliability as a function of different delay requirements $D_{\max} = 0.05, 0.1, 1\,s$ and reliability requirements $R_{\min} = 0.9, 0.93, 0.96, 0.99$ for the number of transmitters $N = 8$.

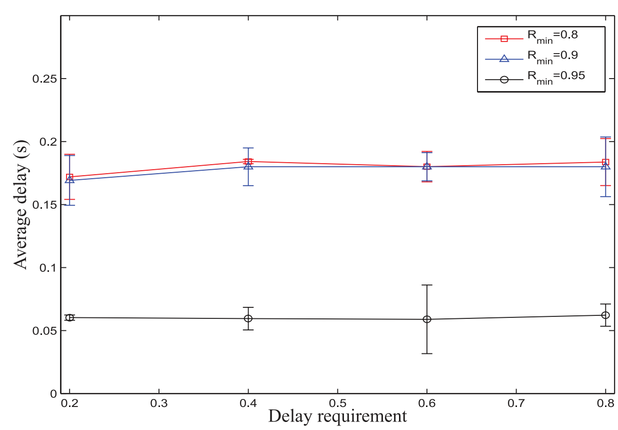


Fig. 16. Average delay as a function of different reliability requirements $R_{\min} = 0.8, 0.9, 0.95$ and delay requirements $D_{\max} = 0.2, 0.4, 0.6, 0.8\,s$ for the number of transmitters $N = 8$.

First, we validate our optimization algorithm for different reliability and delay requirements. The optimal duty-cycle is obtained by using the AODC algorithm. Figure 15 shows the reliability obtained by this algorithm with different reliability constraints $R_{\min} = 0.9, 0.93, 0.96, 0.99$ and delay constraints $D_{\max} = 0.05, 0.1, 1\,s$, whereas Figure 16 shows the average delay of the algorithm for different reliability requirements $R_{\min} = 0.8, 0.9, 0.95$ and delay requirements $D_{\max} = 0.2, 0.4, 0.6, 0.8\,s$. As the delay requirement becomes more strict decreasing from $D_{\max} = 1\,s$ to $D_{\max} = 0.05\,s$ in Figure 15, the reliability requirement $R_{\min} = 0.9, 0.93$ is inactive. As observed in Figure 16, the effect of both the reliability $R_{\min} = 0.8, 0.9$ and delay requirements $D_{\max} = 0.2, 0.4, 0.6, 0.8\,s$ for the average delay is negligible, because the sleep time minimizing the power consumption is the dominant factor of the optimization problem. The average delay decreases as the reliability constraint becomes more strict $R_{\min} = 0.95$ because the sleep time decreases to meet the reliability constraint.

Figure 17 shows the power consumption obtained by X-MAC and the AODC algorithm. Recall that X-MAC does not take into account random back-off, reliability, and delay constraints. Therefore, for the sake of comparison of the AODC algorithm and X-MAC, we pose $R_{\min} = 0$ and $D_{\max} = \infty$, which implies neglecting the reliability

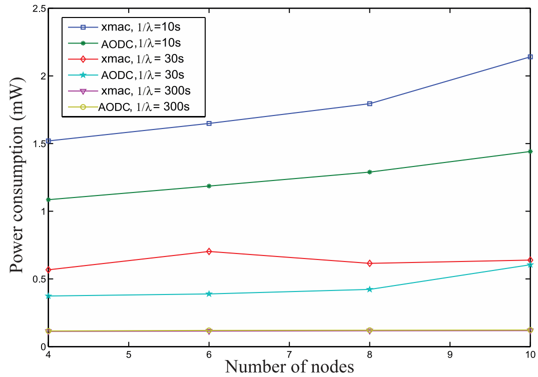


Fig. 17. Comparison of the power consumption of X-MAC and AODC algorithm.

and delay requirements, that is, the power is minimized without constraints, as done in X-MAC. Our protocol outperforms X-MAC in all the scenarios considered. Specifically, when the packet generation period is high (300 s) the difference is small, but as the packet generation period decreases, the improvement is substantial. The main reason for this difference is that the nodes consume much less power in packet transmission compared to the model in [Buettner and Han 2006]. X-MAC is based on the assumption that the transmitter sends preamble packets back to back until the receiver wakes up while actually there is random backoff before packet transmissions during which the transmitter puts its radio in sleep mode. Since the transmission cost of preambles dominates the receiving cost of idle listening much earlier according to the model in Anderson et al. [2006], the optimal sleep time is computed to be much smaller than the actual optimal sleep time.

8.2. Protocol Behavior in Transient Conditions

The performance analysis carried out so far assumed that the number of nodes and traffic configuration are fixed. This assumption has allowed us to verify the effectiveness of the AODC algorithm for IEEE 802.15.4 in steady-state conditions. However, one of the critical issues in the design of wireless networks is the time varying condition. Therefore, in the following analysis, we will investigate the performance of the AODC algorithm when the number of nodes and traffic load changes over time.

We now compare our AODC algorithm to the algorithm AADCC proposed in Merlin and Heinzelman [2010]. AADCC employs a simple linear increase/linear decrease of the sleep time, where whenever five consecutive packets are successfully sent to the destination, the sleep time is increased by 0.1 s. Otherwise, each node decreases the sleep time by 0.25 s. We consider AADCC due to its implementation simplicity with respect to the much more complex DDCC algorithm also proposed in Merlin and Heinzelman [2010]. We remark that AADCC considers only the reliability, while the AODC algorithm controls both reliability and delay of the network. Hence, AADCC does not support different reliability and delay requirements of applications, while our algorithm is adaptive to them, as shown in Figures 15 and 16.

Figure 18 shows the variations in sleep time, reliability, and packet delay of the AODC and AADCC algorithms when the number of nodes changes from $N = 10$ to $N = 15$. At time 300 s, the number of nodes suddenly increases to 15, whereas the experiment-based model in use is still the one for $N = 10$. This causes a significant decrease of the reliability due to the high contention level, as shown in the Figure 18(b).

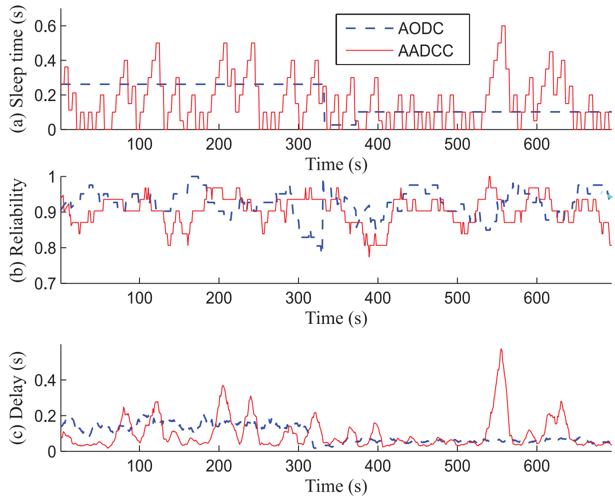


Fig. 18. Robustness of AODC and AADCC algorithms to the changes in the number of nodes: sleep time, reliability and delay when the number of nodes changes sharply from $N = 10$ to $N = 15$ at time 300 s. The packet generation period is $1/\lambda = 10$ s and the reliability and delay constraints are $R_{\min} = 0.9$ and $D_{\max} = 0.5$ s, respectively. Note that AADCC refers to the adaptive algorithm in Merlin and Heinzelman [2010].

The sleep time is updated by starting the learning phase of our adaptive algorithm. When the receiver node detects the change of network condition due to greater number of nodes, it initializes the sliding window and decreases the sleep time to δT_s, where $\delta = 0.1$. After changing the sleep time, the node measures the reliability and delay of the network. The node then runs the optimization phase of the AODC algorithm and updates the sleep time to 0.102 s. We observe that the convergence of the sleep time of the AODC algorithm is very fast without significant oscillations. Also, we observe the high correlation between the sleep time and packet delay. By contrast, the sleep time of AADCC oscillates between 0 s and 0.65 s instead of converging, which is not desirable. Note that the sleep time of AADCC is zero at some points in time due to its simple linear increase/linear decrease mechanism. Although the reliability of the AODC and AADCC are similar, the delay of AADCC has a very high variance.

Figure 19 presents the behavior of AODC and AADCC when the traffic load changes suddenly from $1/\lambda = 30$ s to $1/\lambda = 10$ s at time 300 s. The experiment-based model estimated by the AODC algorithm for $1/\lambda = 30$ s needs to be changed once the traffic load changes. Similar to the case where the number of the nodes changes, the node runs the learning phase of the AODC algorithm, since the measured reliability does not meet the reliability requirement due to the high traffic load. The algorithm then finds the new optimal sleep time during the optimization phase. Figure 19(a) shows that the node updates the sleep time from 2.34 s to 0.27 s due to the poor reliability after the traffic load changes at time 300 s. The figure indicates that the system reacts correctly to the changes of traffic configuration after updating the experiment-based model in few seconds. After the sleep time is optimized, the average delay converges to around 0.18 s. We observe that the packet delay is about five times lower than the one measured before time 300 s in Figure 19(c). Specifically, we have a reduction in the average delay and a shorter tail for the delay distribution after changing the sleep time. The reliability requirement $R_{\min} = 0.9$ is fulfilled with some fluctuations after the traffic load increases. The sleep time of AADCC oscillates between 0 s and 2.4 s

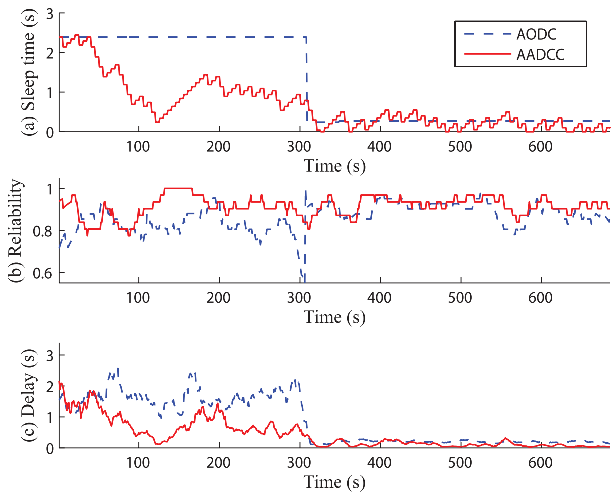


Fig. 19. Robustness of the AODC and AADCC algorithms to the changes in the traffic load: sleep time, reliability, and delay when the traffic load changes sharply from $1/\lambda = 30$ s to $1/\lambda = 10$ s at time 300 s. The number of transmitters is $N = 10$, the reliability and delay constraints are $R_{\min} = 0.8$ and $D_{\max} = 3$ s, respectively.

without converging. Although the reliability of AADCC is higher than the adaptive algorithm, it consumes more power. Furthermore, AADCC does not have control of the delay. Recall that our target is to meet the reliability and delay requirements rather than just improving the reliability or delay performance.

9. CONCLUSIONS

9.1. Summary

We presented the AODC algorithm to minimize the power consumption while guaranteeing reliability and delay requirements of the application for the IEEE unslotted 802.15.4 sensor networks. This approach represents a major advancement with respect to the existing solutions, such as X-MAC and AADCC protocols, because the parameters of the underlying model are able to gracefully adapt to the variations in the application requirements and network topology. The AODC algorithm is easily implementable on top of random access mechanism of the unslotted IEEE 802.15.4 standard. Simple experiment-based models are used to derive the objective function and constraints of the optimization problem as a function of the sleep time. This simplification allows for solving the optimization problem in closed form, hence making it possible to compute the optimal solution at the sensor nodes. The learning phase of the experiment-based model is proposed to adaptively react to the changes in the network. We provided a test-bed implementation of the protocol with TelosB sensors and Contiki OS. Furthermore, we investigated the performance of the AODC algorithm under both stationary and transient conditions by experiments. Experimental results showed that the AODC algorithm is efficient and ensures a longer lifetime of the network. We showed that even if the number of active nodes and traffic configuration suddenly changes, the AODC algorithm allows the network to adapt quickly and operate at the optimal parameter by continuously learning the experiment-based models. Furthermore, for homogeneous Poisson arrivals, the duty-cycle protocol is modeled as aM/D/1/B queuing system in a star network. We obtain analytical expressions for the reliability, average delay, and average power consumption of the duty-cycle protocol.

9.2. Discussion and Future Work

In the future, we are planning to extend the adaptive duty-cycle optimization protocol proposed for one-hop networks to multihop networks. The main challenge in this extension is dealing with the performance degradation due to the interconnection between the nodes and analyzing the convergence time to the optimal duty cycle in each node. The performance degradation occurs when a node sends a preamble packet during its listening time, which prevents it from receiving preambles or data packets from other nodes. This problem can be solved by allowing the nodes to send preamble packets except during the listening time. Moreover, in multihop networks, most of the nodes are both receiver and transmitter of data packets and therefore adapt their duty cycles by running our proposed algorithm. The convergence time to the optimal duty-cycle value at each node needs to be analyzed considering the interactions between the transmissions due to the common wireless channel.

Another interesting future work is extending the objective of the optimization problem for minimizing total energy consumption to other power-related optimization criteria, such as maximizing the network lifetime and minimizing the maximum power consumption of any node.

REFERENCES

ANDERSON, E., BUETTNER, M., YEE, G., AND HAN, R. 2006. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of ACM (SenSys)*.

BACHIR, A., DOHLER, M., WATTEYNE, T., AND LEUNG, K. K. 2010. MAC essentials for wireless sensor networks. *IEEE Commu. Surv. Tutori. 12*, 2, 222–248.

BOYD, S. AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press.

BRUN, O. AND GARCIA, J. M. 2009. Analytical solution of finite capacity M/D/1 queues. *App. Probab. Trust 37*, 4, 1092–1098.

CAMERON, A. C. AND WINDMEIJER FRANK, A. G. 1997. An R-squared measure of goodness of fit for some common nonlinear regression models. *J. Econometrics 77*, 2, 329–342.

CHEN, B., JAMIESON, K., BALAKRISHNAN, H., AND MORRIS, R. 2001. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. In *Proceedings of ACM MobiCom*.

COHEN, R. AND KAPCHITS, B. 2009. An optimal wake-up scheduling algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network. *IEEE/ACM Trans. Netw. 17*, 2, 570–581.

COLERI-ERGEN, S. AND VARAIYA, P. 2006. PEDAMACS: Power efficient and delay aware medium access protocol for sensor networks. *IEEE Trans. Mobile Comput. 5*, 7, 920–930.

VAN DAM, T. AND LANGENDOEN, K. 2003. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of ACM (SenSys)*.

DIETRICH, I. AND DRESSLER, F. 2009. On the lifetime of wireless sensor networks. *ACM Trans. Sen. Netw. 5*, 1, 1–39.

DUNKELS, A., GRONVALL, B., AND VOIGT, T. 2004. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 1st Workshop on Embedded Networked Sensors (EmNets)*.

EL-HOIYDI, A. AND DECOTIGNIE, J. D. 2004. WiseMAC: An ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In *Proceedings of the IEEE Symposium on Computers and Communication (ISCC)*.

FISCHIONE, C., COLERI-ERGEN, S., PARK, P., JOHANSSON, K. H., AND SANGIOVANNI-VINCENTELLI, A. 2009. Medium access control analytical modeling and optimization in unslotted IEEE 802.15.4 wireless sensor networks. In *Proceedings of IEEE (SECON)*.

GUO, C., ZHONG, L. C., AND RABAEY, J. M. 2001. Low-power distributed MAC for ad hoc sensor radio networks. In *Proceedings of Globecome*.

HEINZELMAN, W., CHANDRAKASAN, A., AND BALAKRISHNAN, H. 2000. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS)*.

HILL, J. L. AND CULLER, D. 2002. Mica: A wireless platform for deeply embedded networks. *IEEE Micro 22*, 6, 12–24.

Hua, C. and Yum, T.-S. P. 2007. Asynchronous random sleeping for sensor networks. *ACM Trans. Sen. Netw. 3*, 3.

IEEE. 2006. IEEE 802.15.4 Standard: Wireless medium access control and physical layer specifications for low-rate wireless personal area networks. IEEE. http://www.ieee802.org/15/pub/TG4.html.

IEEE. 2010. IEEE 802.15 Task Group 4e: wireless medium access control and physical layer specifications for low-rate wireless personal area networks. IEEE. http://www.ieee802.org/15/pub/TG4e.html.

ISA. 2009. *ISA-100.11a-2009 Wireless Systems for Industrial Automation: Process Control and Related Applications*. International Society of Automation.

Jurdak, R., Ruzzelli, A. G., and O'Hare, G. M. P. 2010. Radio sleep mode optimization in wireless sensor networks. *IEEE Trans. Mobile Comput. 9*, 7, 955–968.

Kim, J., Lin, X., and Shroff, N. B. 2010. Minimizing the delay and maximizing lifetime for wireless sensor networks with anycast. *IEEE/ACM Trans. Netw. 12*, 2, 515–528.

Lai, W. and Paschalidis, I. C. 2008. Optimally balancing energy consumption versus latency in sensor network routing. *ACM Trans. Sen. Netw. 4*, 4, 1–28.

Langendoen, K. and Meier, A. 2010. Analyzing MAC protocols for low data-rate applications. *ACM Trans. Sen. Netw. 7*, 2, 1–40.

Merlin, C. J. and Heinzelman, W. B. 2010. Duty cycle control for low-power-listening MAC protocols. *IEEE Trans. Mobile Comput. 9*, 11, 1508–1521.

Ning, X. and Cassandras, C. G. 2010. Dynamic sleep time control in wireless sensor networks. *ACM Trans. Sen. Netw. 6*, 3, 1–37.

Osterlind, F. and Dunkels, A. 2008. Approaching the maximum 802.15.4 multi-hop throughput. In *Proceedings ACM (HotEmNets)*.

Park, T. R., Park, K. J., and Lee, M. J. 2009. Design and analysis of asynchronous wakeup for wireless sensor networks. *IEEE Trans. Wirel. Commun. 8*, 11, 5530–5541.

Polastre, J., Hill, J., and Culler, D. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of ACM (SenSys)*.

Qin, Y. and Park, P. 2011. IEEE 802.15.4 Project using ContikiOS. Tech. rep. KTH. Royal Institute of Technology. http://www.ee.kth.se/~pgpark/code/duty-cycle-wpan.zip.

Shi, X. and Stromberg, G. 2007. SyncWUF: An ultra low-power MAC protocol for wireless sensor networks. *IEEE Trans. Mobile Comput. 6*, 1, 115–125.

Srinivasan, K., Kazandjieva, M. A., Agarwal, S., and Levis, P. 2008. The beta-factor: Measuring wireless link burstiness. In *Proceedings of ACM (SenSys)*.

Uysal-Biyikoglu, E., Prabhakar, B., and Gamal, A. El. 2002. Energy-efficient packet transmission over a wireless link. *IEEE/ACM Trans. Netw. 10*, 12, 487–499.

Wheeler, A. 2007. Commercial applications of wireless sensor networks using ZigBee. *IEEE Commun. Mag. 45*, 4, 70–77.

Willig, A., Matheus, K., and Wolisz, A. 2005. Wireless technology in industrial networks. *Proc. IEEE 93*, 6, 1130–1151.

Wood, A. D. and Stankovic, J. A. 2002. Denial of service in sensor networks. *Computer 35*, 10, 54–62.

Xu, Y., Heidemann, J., and Estrin, D. 2001. Geography-informed energy conservation for ad-hoc routing. In *Proceedings of ACM (MobiCom)*.

Ye, W., Heidemann, J., and Estrin, D. 2004. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw. 12*, 3, 493–506.

Zhang, W., Branicky, M. S., and Phillips, S. M. 2001. Stability of networked control systems. *IEEE Control Syst. Mag. 21*, 1, 84–99.

Received June 2012; revised November 2012; accepted March 2013