

Breath: An Adaptive Protocol for Industrial Control Applications Using Wireless Sensor Networks

Pangun Park, Carlo Fischione, *Member, IEEE*, Alwise Bonivento, Karl Henrik Johansson, and Alberto L. Sangiovanni-Vincentelli, *Fellow, IEEE*

Abstract—An energy-efficient, reliable and timely data transmission is essential for Wireless Sensor Networks (WSNs) employed in scenarios where plant information must be available for control applications. To reach a maximum efficiency, cross-layer interaction is a major design paradigm to exploit the complex interaction among the layers of the protocol stack. This is challenging because latency, reliability, and energy are at odds, and resource-constrained nodes support only simple algorithms. In this paper, the novel protocol Breath is proposed for control applications. Breath is designed for WSNs where nodes attached to plants must transmit information via multihop routing to a sink. Breath ensures a desired packet delivery and delay probabilities while minimizing the energy consumption of the network. The protocol is based on randomized routing, medium access control, and duty-cycling jointly optimized for energy efficiency. The design approach relies on a constrained optimization problem, whereby the objective function is the energy consumption and the constraints are the packet reliability and delay. The challenging part is the modeling of the interactions among the layers by simple expressions of adequate accuracy, which are then used for the optimization by in-network processing. The optimal working point of the protocol is achieved by a simple algorithm, which adapts to traffic variations and channel conditions with negligible overhead. The protocol has been implemented and experimentally evaluated on a testbed with off-the-shelf wireless sensor nodes, and it has been compared with a standard IEEE 802.15.4 solution. Analytical and experimental results show that Breath is tunable and meets reliability and delay requirements. Breath exhibits a good distribution of the working load, thus ensuring a long lifetime of the network. Therefore, Breath is a good candidate for efficient, reliable, and timely data gathering for control applications.

Index Terms—Wireless sensor networks, control over multihop WSNs, cross-layer design, duty cycle, optimization.

1 INTRODUCTION

WIRELESS Sensor Networks (WSNs) are networks of tiny sensing devices for wireless communication, monitoring, control, and actuation. Given the potential benefits offered by these networks, e.g., simple deployment, low installation cost, lack of cabling, and high mobility, they are specially appealing for control and industrial applications [2], [3], [4]. The variety of application domains and theoretical challenges for WSNs has attracted research efforts for more than a decade. Nevertheless, a lively research and standardization activity is ongoing [3], [4], [5].

Although WSNs provide a great advantage for process, manufacturing and industry, they are not yet efficiently deployed. This is because the software for these applications is usually written by process and software engineers that are expert in process control technology, but know little of the

network and sensing infrastructure that has to be deployed to support control applications. On the other side, the communication infrastructure is designed by communication engineers that know little about process control technology. Moreover, the adoption of wireless technology further complicates the design of these networks. Being able to satisfy high requirements on communication performance over unreliable communication channels is a difficult task.

Standard practice for control system design over communication networks is as follows: First, deploy the networked embedded system on a predefined distributed architecture, chosen on the basis of experience and heuristic considerations. Then, tweak the software implementation of the control algorithm to meet latency, bandwidth, and reliability offered by the network. In many control designs, the network imperfections are completely disregarded, assuming instead that sensor and control data instantaneously reach the controller and actuator node, respectively.

This is far from ideal, because many control systems are highly cost-sensitive, and using a nonoptimized network is clearly expensive. Moreover, the complexity of large networked embedded systems continues to increase, making heuristic and experience-based design practices inadequate at best. To bridge this gap and derive a correct and efficient implementation, a system-level approach has been proposed in [6], [7]. By a system-level design for WSNs, the control algorithm designers impose a set of requirements on reliability, packet delay and energy consumption that the communication infrastructure must satisfy.

• P. Park, C. Fischione, and K.H. Johansson are with the ACCESS Linnaeus Centre, Electrical Engineering Department, Royal Institute of Technology, 6th Floor, Osquidas Väg 10, KTH, SE-100 44 Stockholm, Sweden. E-mail: {pgpark, kallej}@kth.se, carlofi@ee.kth.se.

• A. Bonivento is with Fondo Atlante Ventures, Intesa Sanpaolo Group, Piazza Scala 6, 20100 Milano, Italy. E-mail: alwisebonivento@yahoo.com.

• A.L. Sangiovanni-Vincentelli is with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, 231 Cory Hall MC# 1770, Berkeley, CA 94720-1770. E-mail: alberto@eecs.berkeley.edu.

Manuscript received 7 Dec. 2009; revised 1 July 2010; accepted 26 Aug. 2010; published online 16 Nov. 2010.

For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2009-12-0532. Digital Object Identifier no. 10.1109/TMC.2010.223.

An efficient system-level design process for operations of WSNs in industrial control applications poses extra challenges compared to more traditional communication networks, namely:

- **Reliability:** Sensor information must be sent to the sink of the network with a given probability of success, because missing these data could prevent the correct execution of control actions or decisions concerning the phenomena sensed. However, maximizing the reliability may increase substantially the network energy consumption [3]. Hence, the network designers need to consider the trade-off between reliability and energy consumption.
- **Delay:** Sensor information must reach the sink within some deadline. A probabilistic delay requirement must be considered instead of using average packet delay since the delay jitter can be too difficult to compensate for, especially if the delay variability is large. Retransmission of old data to maximize the reliability may increase the delay and is generally not useful for control applications [8].
- **Energy efficiency:** The lack of battery replacement, which is essential for affordable WSN deployment, requires energy-efficient operations. Since high reliability and low delay may demand a significant energy consumption of the network, thus reducing the WSN lifetime, the reliability and delay must be flexible design parameters that need to be adequate for the requirements. Note that controllers can usually tolerate a certain degree of packet losses and delay [9], [10], [11]. Hence, the maximization of the reliability and minimization of the delay are not the optimal design strategies for the control applications we are concerned within this paper.
- **Adaptation:** The network operation should adapt to application requirement changes, varying wireless channel, and network topology. For instance, the set of application requirements may change dynamically and the communication protocol must adapt its design parameters according to the specific requests of the control actions. To support changing requirements, it is essential to have an analytical model describing the relation between the protocol parameters and performance indicators (reliability, delay, and energy consumption).
- **Scalability:** Since the processing resources are limited, the protocol procedures must be computationally light. These operations should be performed within the network, to avoid the burden of too much communication with a central coordinator. This is particularly important for large networks. The protocol should also be able to adapt to size variation of the network, as, for example, caused by moving obstacles, or addition of new nodes.

In this paper, we offer a complete design approach that embraces all the factors mentioned above. We propose the Breath protocol, a self-adapting efficient solution for reliable and timely data transmission. Since the protocol adapts to the network variations by enlarging or shrinking next-hop distance, sleep time of the nodes, and transmit radio power, we think that it behaves like a breathing organism.

The rest of the paper is organized as follows: In Section 1.1, we motivate our study and summarize existing work. Section 1.2 presents the main contributions of the paper. In Section 2, we define the system scenario. In Section 3, we introduce Breath in detail. In Section 4, an optimization problem is posed to optimize the protocol, whereas in Section 5 the constraints and cost function of the protocol are modeled. In Section 6, we derive the optimal solution and in Section 7, we present an adaptive algorithm to obtain the working point of the protocol. The fundamental working limits of Breath are given in Section 8. A complete experimental implementation of the protocol is presented in Section 9. Finally, in Section 10 concluding remarks and future perspectives are given.

1.1 Related Works

There have been many contributions to the problem of protocol design for WSNs, both in academia (e.g., [3], [12]) and industry (e.g., [13], [14], [15]). New protocols have been built around standardized low-power protocols such as IEEE 802.15.4 [5], Zigbee [16], and WirelessHART [17]. WirelessHART is a promising solution for the replacement of the wired HART protocol in industrial contexts. However, the power consumption is not a main concern in WirelessHART, whereas the data link layer is based on Time Division Multiple Access (TDMA), which requires time synchronization and prescheduled fixed length time slots by a centralized network manager. Such a manager should update the schedule frequently to consider reliability and delay requirements and dynamic changes of the network, which demands complex hardware equipments. WirelessHART is thus in contrast with the necessity of simple protocols able to work with limited energy and computing resources. In Table 1, we summarize the characteristics of the protocols that are relevant for the category of applications we are concerned within this paper. In the table, we have evidenced whether indications as energy E , reliability R , and delay D have been included in the protocol design and validation, and whether a cross-layer approach has been adopted. We discuss these protocols in the following.

We categorize first the MAC protocols as follows: random-based, TDMA-based, and hybrid-based access mechanism. In random-based MAC protocol, each nodes periodically wake up, listen to the channel, and then go back to sleep again. X-MAC [18] improves B-MAC [19] by using a strobed sequence of short packets including the target ID allowing for fast shutdown and response. This scheme addresses the overhearing overhead of long preambles of Low-Power-Listening (LPL) protocols and saves the energy consumption of the nontarget receivers. X-MAC also includes a lookup table to adapt the duty cycle of the nodes based on the traffic load. However, this is suboptimal solution when there are multiple transmitters and receivers in the network since X-MAC only optimizes the energy consumption of the network with only one receiver. The reverse approach to LPL was redesigned for WSNs as the Receiver-Initiated (RI)-MAC [20]. The receiver sends out beacon messages at regular intervals and a sender must wait until it receives one and respond by sending the message in the rendezvous action to minimize channel usage (i.e., no long preambles). However, the main drawback is that beacon messages interfere with ordinary traffic as well as with each other.

TABLE 1
Protocol Comparison

Protocol	E	R	D	Layer
X-MAC [18]	⊕	·	·	MAC
B-MAC [19]	⊕	·	·	MAC
RI-MAC [20]	⊕	⊕	·	MAC
SCP-MAC [21]	⊕	⊕	·	MAC
Crankshaft [22]	⊕	⊕	·	MAC
Funneling [23]	⊕	⊕		MAC
GAF [24]	○	·	·	bridge
SPAN [25]	○	·	·	bridge
Fetch [26]	·	⊕	·	phy, MAC, routing
Dozer [27]	⊕	⊕		MAC, routing
Koala [28]	⊕	⊕		MAC, routing
GERAF [29]	○		○	MAC, routing
CTP [30]	⊕	⊕		phy, MAC, routing
BCP [31]	·	⊕	⊕	MAC, routing
MMSPEED [32]		○	○	routing
Breath	⊕	⊕	⊕	phy, MAC, routing

The letters E, R, and D denote energy, reliability, and communication delay. The circle denotes that a protocol is designed by considering the indication of the column, but it has not been validated experimentally. The circle with plus denotes that the protocol is designed by considering the indication and experimentally validated. The dot denotes that the protocol design does not include indication and hence cannot control it, but simulation or experiment results include it. The term “bridge” means that the protocol is designed by bridging MAC and routing layers.

In TDMA-based MAC protocols, nodes only wake up and listen to the channel in assigned slots and then go back to sleep in other slots. Scheduled Channel Polling (SCP)-MAC [21] does not require long preambles of LPL, and is able to operate ultralow duty cycles when traffic is light by synchronizing the channel polling times based on a common slot structure. Crankshaft [22] is similar to SCP-MAC and specifically targeted for dense network. It employs node synchronization and offset wake-up schedules to restraint the main cause of inefficiency in dense networks. However, the throughput decreases at low traffic load due to idle slots. Furthermore, it is difficult to synchronize all the nodes of the network to eliminate the clock drifting and adapt to the changes of the topology. Moreover, adapting the slot assignment is not easy within a decentralized environment for traditional TDMA.

Some hybrid MAC protocols are proposed by combining the advantages of both a random access with contention and a TDMA without contention. To offer flexible quality of service to several classes of applications, the IEEE 802.15.4 standard provides optional hybrid MAC mechanism based on its superframe structure. Funneling-MAC [23] mainly uses a CSMA/CA mechanism in the network except a localized TDMA algorithm of the funneling region closer to the sink. The sink node manages the TDMA scheduling of the funneling region instead of the whole sensor field.

GAF [24] and SPAN [25] consider the energy efficiency as a performance indicator, which is attained by algorithms under the routing layer and above the MAC layer so-called bridge layer. Both protocols are similar since it activates only a fraction of the nodes in a certain area at any given time. A

major weakness of GAF is precisely the requirement that the routing feature be guaranteed, which results in inefficiency in terms of latency and energy consumption. In SPAN, the energy consumption significantly increases as the number of nodes increases. Simulation results of reliability and delay are reported in [24], [25]. In summary, these protocols [18], [19], [20], [21], [22], [23], [24], [25] have not been designed out of an analytical modeling of reliability and delay, so there is not systematic control of them.

Fetch [26] and Dozer [27] are designed for monitoring application, which mainly deals with lower traffic load than control applications. The latency of Fetch [26] is significantly dependent on the depth of the routing tree and is around some hundred seconds. Dozer [27] comprises the MAC and routing layer to minimize the energy consumption while maximizing the reliability of the network, but an analytical approach has not been followed. Furthermore, Dozer requires synchronized sleep schedules and nodes to persistently maintain routing trees. In addition, experimental results of Dozer [27] show good energy efficiency and reliability under very low traffic intensity (with data sampling interval of 120 s) but the delay in the packet delivery is not considered, which is essential for control applications [8], [9]. Koala [28] is a protocol similar to Dozer. It coordinates the sleep schedules for bulk transfer applications, whereas it does not consider control applications. Furthermore, the packet delay of Koala is much higher than Dozer because Koala does not use synchronized sleep schedules and maintenance of routing trees. Energy efficiency with delay requirement for MAC and randomized routing is considered in GERAF [29], without simulation or experimental validation. Collection Tree Protocol (CTP) [30] is a best-effort tree-based anycast protocol used to collect the data from sensor nodes. Note that CTP is mainly targeted for low traffic rates such as monitoring applications. CTP estimates first the quality of the link, then it decides the parent node merely based on the link quality. However, this approach may incur a load balancing problem because the node with good quality link will be selected as the preferred parent and consumes more energy. Furthermore, CTP uses a very aggressive retransmission policy, i.e., the default value of the maximum number of retransmissions is 32 times. Note that retransmission of old data to maximize the reliability may increase the delay and is generally not useful for control applications [8]. Backpressure Collection Protocol (BCP) [31] considers alternative approach based on dynamic backpressure routing of wireless networks. In BCP, the routing and forwarding decision is made on a per-packet basis by computing a backpressure weight of each outgoing link that is a function of localized queue and link state information. Therefore, the overhead due to the backpressure algorithm depends on the all possible forwarding nodes of the next hop. Furthermore, the backpressure algorithm does not prevent loops of the routing and may incur in large delay.

The focuses of the protocols mentioned above [24], [25], [26], [27], [28], [29], [30], [31] are the maximization of the energy efficiency or reliability, or just minimization of the delay, without considering simultaneously application requirements in terms of reliability and delay in the packet

delivery. In other words, these protocols are mostly designed for monitoring applications and do not support typical control requirements. Control and industrial applications are able to cope with a certain degree of packet losses and delay [9], [10], which implies that the approaches followed in the protocols mentioned above are not the ideal solution for these applications. The maximization of the energy efficiency and reliability may give a long delay, which are bad for the stability of the closed-loop control system. Analogously, the maximization of the reliability may be energy demanding and may give long delay, all of which are not tolerable for control applications. In addition, the protocols mentioned above do not support an adaptation to the changes of the reliability and delay, which may be required by the controllers.

The protocols MMSPEED [32] and SERAN [33] are appealing for control and industrial applications. However, MMSPEED is not energy-efficient because it considers a routing technique with an optimization of reliability and delay without energy constraints. The protocol satisfies a high reliability requirement by using duplicated packets over multipath routing. Duplicated packets increase the traffic load with a negative effect on the stability and energy efficiency of the network. In SERAN, a system-level design methodology has been presented for industrial applications, but even though SERAN allows the network to operate with low energy consumption subject to delay requirements, it does not consider tunable reliability requirements nor duty-cycling policies, which are essential to reduce energy consumption. Furthermore, SERAN focuses on low traffic networks. These characteristics limit the performance of SERAN both in terms of energy and reliability in our application setup.

Given the availability of numerous techniques to reduce energy consumption and ensure reliability and low delays, a cross-layer optimization is a natural approach to integrate the protocol layers. Some cross-layer design challenges of the physical, MAC, and network layers to minimize the energy consumption of WSNs have been surveyed in [34], [35], [36]. Many of the cross-layer solutions proposed in the literature are hardly useful for the application domain we are targeting, because they require sophisticated processing resources, or instantaneous global network knowledge, which are out of reach of the capabilities of real nodes. Network design can be formulated as an optimization problem. However, as it was noted in [37], the complex interdependence of the decision variables (sleep disciplines, clustering, MAC, routing, power control, etc.) leads to difficult problems even in simple network topologies, where the analytical relations describing packet reception rate, delay, and energy consumption may be highly non-linear expressions. Such a difficulty is further exacerbated when considering non-TDMA scheme [38]. We propose next a design approach that offers a computationally attractive solution by simplifications of adequate accuracy.

1.2 Original Contribution

In this paper, we present Breath, an adaptive protocol for WSNs for reliable and timely data gathering. Our system model considers nodes that have to send packets to the sink via multihop routing under tunable reliability and delay

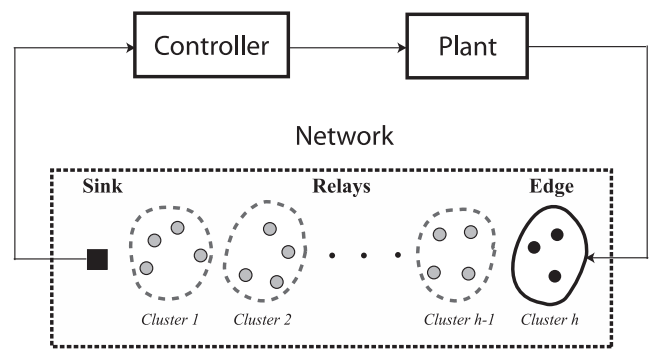


Fig. 1. Wireless control loop. An wireless network closes the loop from sensors to controller. The network includes nodes (black dots) attached to the plant, $h - 1$ relay clusters (gray dots), and a sink (black rectangular) attached to the controller.

requirements. We present a solution based on randomized routing, CSMA/CA MAC and randomized sleep discipline that are jointly optimized for energy consumption. To the best of our knowledge, no efficient and simple cross-layer protocol that includes all the relevant characteristics of the physical layer, MAC, routing, duty cycling, load balancing that minimizes the energy consumption of the network under reliability and delay requirements has been proposed. No protocol in the literature guarantees adaptation to reliability and delay requirements over multihop communication, with optimizing the energy consumption. Especially, our original contribution is as follows:

1. We provide explicit analytical relations of the reliability, delay, and total energy consumption as a function of MAC, routing, physical layer, duty cycle, and radio power. The approach is based on simple yet good approximations whose accuracy is systematically verified.
2. The analytical relations allow us to pose and solve a mixed integer-real optimization problem where the energy minimization is achieved under tunable reliability and delay requirements.
3. Based on this optimization, we develop a novel algorithm that allows for rapid deployment and self-adaptation of the network to traffic variations and channel conditions, and guarantees the application requirements without heavy computation or communication overhead.
4. The protocol is implemented on a testbed using Tmote sensors [13]. We show by analysis and experimental evaluation the benefits of our solution.

2 SYSTEM SCENARIO

We consider the scenario depicted in Fig. 1, where a plant is remotely controlled over a WSN [8], [10]. Outputs of the plant are sampled at periodic intervals by the sensors with total packet generation rate of λ pcks/s (see Table 2 for main symbols used in the paper). We assume that packets associated to the state of the plant are transmitted to a sink, which is connected to the controller, over a multihop network of uniformly and randomly distributed relaying nodes. No direct communication is possible between the plant and the sink. Relay nodes forward incoming packets.

TABLE 2
Main Symbols Used in the Paper

Symbol	Meaning
$D(h, \mu_c)$	Distribution of delay
E_{ca}	Average energy consumption in CSMA/CA
E_{tot}	Total energy consumption of the network
E_{pck}	Data packet transmission Energy consumption
E_r	Energy consumption for receiving a data packet
E_{wu}	Energy consumption for wake-up and becoming
h	Number of hops from edge cluster to sink
M_{ca}	Maximum number of CSMA/CA TX tries
N	Number of relays of the network
p_{min}	Minimum successful packet reception probability
P_t	Radio transmission power
Q_b	Energy consumption for a beacon transmission
Q_m	Energy consumption for a packet transmission
$R(h, \mu_c)$	Probability of successful packet delivery
S	Distance from the edge cluster to the sink
T	Total time
T_{ac}	Maximum listening time to receive a data packet
T_{ca}	A unit of backoff period
T_W	Wake-up time from sleep mode
W	Maximum number of random backoff time
A_{rx}	Power consumption at RX mode
A_s	Power consumption to scan a channel
A_{tx}	Power consumption at TX mode
A_w	Power consumption to wake-up
α_i	Exponentially distributed time of intensity μ_c
ε_i	Uniformly distributed backoff time of CSMA/CA
ζ	Percentage of slots in the backoff time counter 0
λ	Traffic rate
μ_c	Cumulative cluster wake-up rate
μ_k	Wake-up rate of node k
ν	Busy channel probability in CSMA/CA
τ	Required delay
$\psi_{sc}(n)$	Probability of successful TX in CSMA/CA
$\psi_{sb}(\mu_c, n)$	Competition probability of n nodes in $1/\lambda$ s
Δ	Required delay probability
Ω	Required reliability

When the controller receives the measurements, they are used in a control algorithm to compensate the control output. The control law induces constraints on the communication delay and the packet loss probability. Packets must reach the sink within some minimum reliability and maximum delay. These boundaries are denoted as *application requirements* throughout this paper. The application requirements are chosen by the control algorithm designers. Since they can change from one control algorithm to another, or a control algorithm can ask to change the application requirements from time to time, we allow them to vary. We assume that nodes of the network cannot be recharged, so the operations must conserve energy. The system scenario is quite general, because it applies to any interconnection of a plant by a multihop WSN to a controller tolerating a certain degree of data loss and delay [9], [10], [11].

A typical example of the scenario described above is an industrial control application. In particular, a WSN with nodes uniformly distributed in the walls or in the ceiling can be deployed as the network infrastructure that support the control of the state of the robots in a manufacturing cell. Typically, a cell is a stage of an automation line. Its physical dimensions range around 10 or 20 meters on each side.

Several robots cooperate in the cell to manipulate and transform the same production piece. The typical way of monitoring the state of a robot is to observe its vibration pattern of the different parts of a robot. If the values of these vibrations are above a given threshold, a controller send the control message to these robots. Hence, each node senses vibrations and has to report the data to the controller within a delay. The decision-making algorithm runs on the controller, which is usually a processor placed outside the cell. Multi-hop communication is needed to overcome the deep attenuations of the wireless channel due to moving metal objects and save energy consumption.

3 THE BREATH PROTOCOL

The Breath protocol groups all N nodes between the cluster of nodes attached to the plant and the sink with $h - 1$ relay clusters. Data packets can be transmitted only from a cluster to the next cluster closer to the sink. Clustered network topology is supported in networks that require energy efficiency, since transmitting data through relays consumes less energy than routing directly to the sink [39]. In [40], a dynamic clustering method adapts the network parameters. In [39] and [41], a cluster header is selected based on the residual energy levels for clustered environments. However, the periodic selection of clustering may not be energy-efficient, and does not ensure the flexibility of the network to a time-varying wireless channel environment. A simpler geographic clustering is instead used in Breath. Nodes in the forwarding region send short beacon messages when they are available to receive data packets. Beacon messages are exploited to carry information related to the control parameters of the protocol. When a node receives a beacon message with the updated number of clusters $h - 1$, then the node adapts to its cluster based on a rough knowledge of its location.

In the following sections, we will describe the protocol stack and state machine of Breath in Sections 3.1 and 3.2, respectively.

3.1 The Breath Protocol Stack

Breath uses a randomized routing, a CSMA/CA mechanism at the MAC, radio power control at the physical layer, and sleeping disciplines. We give details in the following.

In many industrial environments, the wireless conditions vary heavily because of moving metal obstacles and other radio disturbances. In such situations, routing schemes that use fixed routing tables are not able to provide the flexibility over mobile equipments, physical design limitations, and reconfiguration typical of an industrial control application. Fixed routing is inefficient in WSNs due to the cost of building and maintaining routing tables. To overcome this limitation, routing through a random sequence of hops has been introduced in [29]. The Breath protocol is built on an optimized random routing, where next hop route is efficiently selected at random. Randomized routing allows us to reduce overhead because no node coordination or routing state needs to be maintained by the network. Robustness to node failures is also considerably increased by randomized routing. Therefore, nodes route data packets to next-hop nodes randomly selected in a forwarding region.

Each node, either transmitter or receiver, does not stay in an active state all time, but goes to sleep for a random amount of time, which depends on the traffic and channel conditions. Since traffic, wireless channel, and network topology may be time-varying, the Breath protocol uses a randomized duty-cycling algorithm. Sleep disciplines turn off a node whenever its presence is not required for the correct operation of the network. GAF [24], SPAN [25], and S-MAC [42] focus on controlling the effective network topology by selecting a connected set of nodes to be active and turning off the rest of the nodes. These approaches require extra communication, since nodes maintain partial knowledge of the state of their individual neighbors. In Breath, each node goes to sleep for an amount of time that is a random variable dependent on traffic and network conditions. Let μ_c be the cumulative wake-up rate of each cluster, i.e., the sum of the wake-up rates that a node sees from all nodes of the next cluster. The cumulative wake-up rate of each cluster must be the same for each cluster to avoid congestions and bottlenecks.

The MAC of Breath is based on a CSMA/CA mechanism similar to the IEEE 802.15.4. Both data packets and beacon packets are transmitted using the same MAC. Specifically, the CSMA/CA checks the channel activity by performing Clear Channel Assessment (CCA) before the transmission can commence. Each node maintains a variable NB for each transmission attempt, which is initialized to 0 and counts the number of additional backoffs the algorithm does while attempting the current transmission of a packet. Each backoff unit has duration T_{ca} ms. Before performing CCAs, a node takes a backoff of $\text{random}(0, W - 1)$ backoff units, i.e., a random number of backoffs with uniformly distributed over $0, 1, \dots, W - 1$. If the CCA fails, i.e., the channel is busy, NB is increased by one and the transmission is delayed of $\text{random}(0, W - 1)$ backoff periods. This operation is repeated at most M_{ca} times, after which a packet is discarded.

The Breath protocol assumes that each node has a rough knowledge of its location. This information, which is commonly required for the applications we are targeting [3], can be obtained running a coarse positioning algorithm, or using the Received Signal Strength Indicator (RSSI), which is typically provided by off-the-shelf sensor nodes [43]. Some radio chips already provide a location engine based on RSSI [44]. Location information is needed for tuning the transmit radio power and to change the number of hops, as we will see later. The energy spent for radio transmission plays an important role in the energy budget and for the interference in the network. Breath, therefore, includes an effective radio power control algorithm.

3.2 State Machine Description

Breath distinguishes between three node classes: edge nodes, relays, and the sink.

The edge nodes wake up as soon as they sense packets generated by the plant to be controlled. Before sending packets, the edge node waits for a beacon message from the cluster of nodes closer to the edge. Upon the reception of a beacon, the node sends the packet.

Consider a relay node k . Its detailed behavior is illustrated by the state machine of Fig. 2, as we describe in the following:

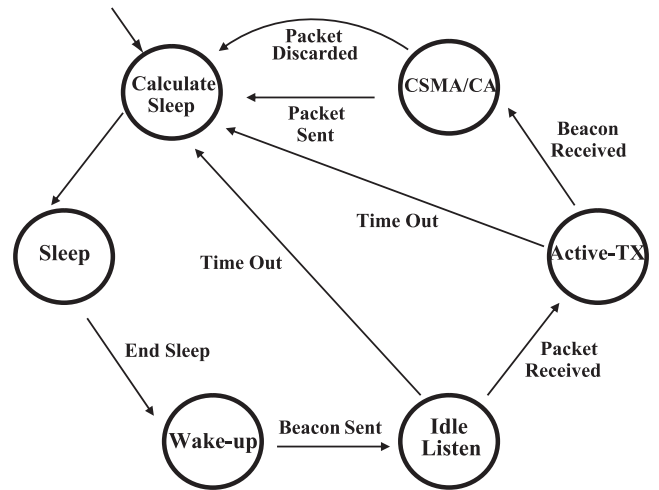


Fig. 2. State machine description of a relay node executing the Breath protocol.

- **Calculate Sleep State:** the node calculates the parameter μ_k for the next sleeping time and generates an exponentially distributed random variable having average $1/\mu_k$. After this, the node goes back to the Sleep State. μ_k is computed such that the cumulative wake-up rate of the cluster μ_c is ensured.
- **Sleep State:** the node turns off its radio and starts a timer whose duration is an exponentially distributed random variable with average $1/\mu_k$. When the timer expires, the node goes to the Wake-up State.
- **Wake-up State:** the node turns its beacon channel on, and broadcasts a beacon indicating its location. Then, it switches to listen to the data channel, and it goes to the Idle Listen State.
- **Idle Listen State:** the node starts a timer of a fixed duration that must be long enough to receive a packet. If a data packet is received, the timer is discarded, the node goes to the Active-TX State, and its radio is switched from the data channel to the beacon channel. If the timer expires before any data packet is received, the node goes to the Calculate Sleep State.
- **Active-TX State:** the node starts a waiting timer of a fixed duration. If the node receives the first beacon coming from a node in the forwarding region within the waiting time, it retrieves the node ID and goes to the CSMA/CA State. Otherwise, if the waiting timer is expired before receiving a beacon, the node goes to the Calculate Sleep State.
- **CSMA/CA State:** the node switches its radio to hear the data channel, and it tries to send a data packet to a node in the next cluster by the CSMA/CA MAC. If the channel is not clean within the maximum number of tries, the node discards the data packet and goes to the Calculate Sleep State. If the channel is clear within the maximum number of attempts, the node transmits the data packet using an appropriate level of radio power and goes to the Calculate Sleep State.

The sink node sends periodically beacon messages to the last cluster of the network to receive data packets. Such a node estimates periodically the traffic rate and the

wireless channel conditions. By using this information, the sink runs an algorithm to optimize the protocol parameters, as we describe in Section 4. Once the results of the optimization are achieved, they are communicated to the nodes by beacons.

According to the protocol given above, the packet delivery depends on the traffic rate, the channel conditions, number of forwarding regions, and the cumulative wake-up time. In the next sections, we show how to model and optimize online these parameters.

4 PROTOCOL OPTIMIZATION

The protocol is optimized dynamically by a constrained optimization problem. The objective function, denoted by $E_{\text{tot}}(h, \mu_c)$, is the total energy consumption for transmitting and receiving packets from the edge cluster to the sink. The constraint are given by the end-to-end packet reception probability and end-to-end delay probability. The optimization problem is

$$\min_{h, \mu_c} E_{\text{tot}}(h, \mu_c) \quad (1a)$$

$$\text{s.t.} \quad R(h, \mu_c) \geq \Omega, \quad (1b)$$

$$\Pr[D(h, \mu_c) \leq \tau] \geq \Delta, \quad (1c)$$

$$h \geq 2, \quad (1d)$$

$$\mu_{\min} \leq \mu_c \leq \mu_{\max}. \quad (1e)$$

The decision variables are the cumulative wake-up rate μ_c of each cluster and the number of relay clusters, $h - 1$. $R(h, \mu_c)$ is the probability of successful packet delivery (reliability) from the edge cluster to the sink, and Ω is the minimum desired probability. $D(h, \mu_c)$ is a random variable describing the delay to transmit a packet from the edge cluster to the sink. τ is the desired maximum delay, and Δ is the minimum probability with which such a maximum delay should be achieved. Constraint (1d) is due to that there is at least two hops from the edge cluster to the sink. Constraint (1e) is due to that the wake-up rate cannot be less than a minimum value μ_{\min} , and larger than a maximum value μ_{\max} due to hardware reasons. Note that Problem (1) is a mixed integer-real optimization problem, because μ_c is real and h is integer. We need to have Δ and Ω close to one. We let $\Delta \geq 0.95$ and $\Omega \geq 0.9$, namely we assume that the delay τ must be achieved at least with a probability of 95 percent, and the reliability must be larger than 90 percent. We remark that τ , Δ , and Ω are application requirements, and h , μ_c , and nodes' radio transmit power are *protocol parameters* that must be adapted to the traffic rate λ , the wireless channel conditions, and the application requirements for an efficient network operation.

In the following, we shall propose an approach to model the quantities of Problem (1), along with a strategy to achieve the optimal solution, namely the values of h^* and μ_c^* that minimize the cost function and satisfy the application requirements. As we will see later, the system complexity prevents us to derive the exact expressions for the analytical relations of the optimization problem. An approximation of the requirements and an upper bound of the energy consumption will be used.

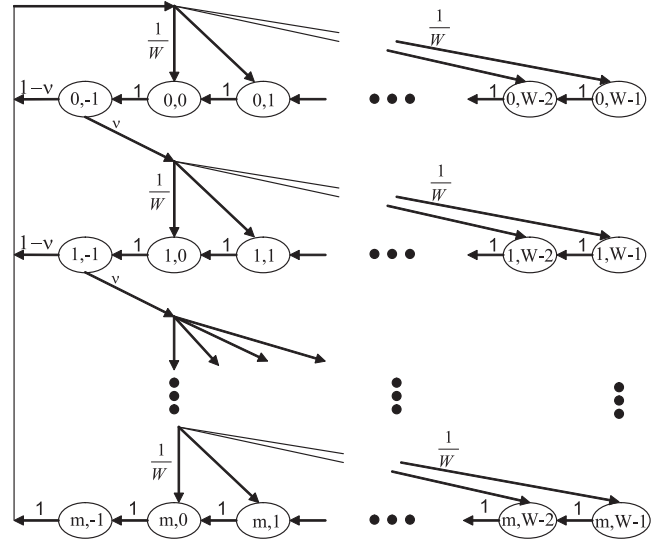


Fig. 3. Markov chain model for CSMA/CA state evolution of Breath.

5 MODELING OF THE PROTOCOL

In this section, we model the reliability, packet delay distribution, μ_c , and total energy consumption of the network.

5.1 Reliability Constraint

In this section, we provide an analytical expression for the reliability constraint (1b) in Problem (1).

A data packet can be lost at a hop because of a bad wireless channel or packet collisions. The collision probability is determined by the CSMA/CA MAC. Therefore, to analyze such a behavior, we use a Markov chain. The approach is similar to the one proposed in [45] and [46] (see also [47], [48], [49]). Let m be the maximum backoff stage, and W be the maximum backoff time of CSMA/CA. Let $s(t) \in \{0, \dots, m\}$ and $b(t) \in \{0, \dots, W - 1\}$ be the stochastic processes representing the backoff stage and the backoff time counter, respectively. The delay spent before a node senses the channel idle is modeled by the Markov chain depicted in Fig. 3. The Markov chain state is $(s(t), b(t))$, where $b(t) = -1$ refers to the assessment of the channel state during CCA. Denote the Markov chain's steady-state probabilities by $b_{i,k} = \Pr\{(s(t), b(t)) = (i, k)\}$. They allow us to compute the probability of successful transmission in CSMA/CA as the probability that exactly one node transmits and $n - 1$ are silent:

$$\psi_{\text{sc}}(n) = \frac{\zeta(1 - \zeta)^{n-1}}{1 - (1 - \zeta)^n},$$

where

$$\zeta = \sum_{i=0}^m b_{i,0} = \frac{2}{W + 3}.$$

From the Markov chain, we derive also the busy channel probability $\nu(n)$, which is

$$\nu(n) = \frac{1 - (1 - \zeta)^{n-1}}{2 - (1 - \zeta)^{n-1}}. \quad (2)$$

We will use this probability in Section 5.3.

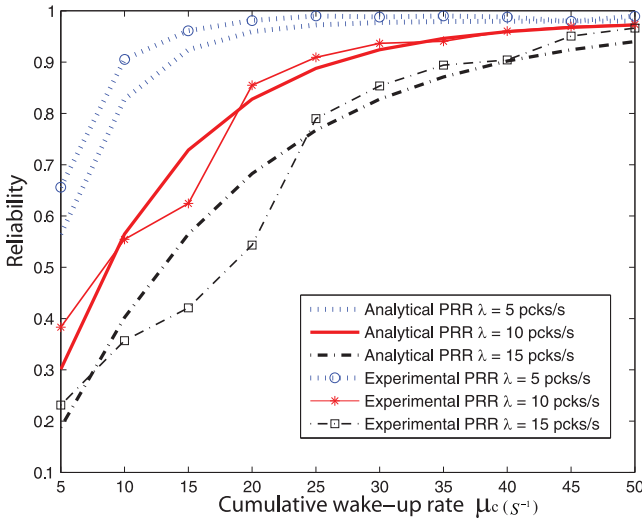


Fig. 4. Reliability as obtained by (4) and experimental results as a function of μ_c . Curves refer to traffic rates $\lambda = 5, 10, 15$ pcks/s for $h = 2$ hops and $N = 15$ nodes.

The probability of successful transmission in CSMA/CA $\psi_{sc}(n)$ depends on the number of nodes n that are contending to transmit packets. We need therefore to compute the probability $\psi_{sb}(\mu_c, n)$ that a generic number n of contending nodes compete within a period $1/\lambda$ to transmit a data packet. By recalling that the cumulative wake-up rate is exponentially distributed random variable with intensity μ_c , and noting that $e^{-\mu_c n/\lambda}$ is the probability to have more than n contending nodes, we conclude that

$$\psi_{sb}(\mu_c, n) = e^{-\mu_c(n-1)/\lambda} - e^{-\mu_c n/\lambda}. \quad (3)$$

Hence, the reliability with $\psi_{sb}(n)$ and $\psi_{sc}(n)$ is

$$R(h, \mu_c) = \prod_{i=1}^h p_i \sum_{n=1}^{\infty} \psi_{sb}(n) \psi_{sc}(n), \quad (4)$$

where p_i denotes the probability of successful packet reception during a single-hop transmission from cluster i to cluster $i-1$.

Since the components of the sum in (4) with $n \geq 2$ give a small contribution, we set $n=2$ and validate (4) by experimental results. Fig. 4 reports the reliability versus μ_c , as obtained by (4) with $n=2$ and experiments for a two-hop network. We see that (4) provides a good approximation of the experimental results because it is always around 5 percent of the experiments for reliability values of practical interest (larger than 0.7). The same behavior is found for h up to 4.

We can rewrite the reliability constraint $R(h, \mu_c) \geq \Omega$ by using (4) with $n=2$, thus obtaining

$$\begin{aligned} \mu_c &\geq f_r(h, \Omega) \triangleq \lambda \ln(2C_r) \\ &\quad - \lambda \ln \left[C_r - 1 + \sqrt{(C_r - 1)^2 - 4C_r(\Omega^{1/h}/p_{\min} - 1)} \right], \end{aligned} \quad (5)$$

where $C_r = \zeta(1 - \zeta)/(1 - (1 - \zeta)^2)$, and $p_{\min} = \min(p_1, \dots, p_h)$. Note that we used the worst channel condition of the network p_{\min} , which is acceptable for optimization purpose because in doing so we consider the minimum of (4). Since

the argument of the square root in (5) must be positive, an additional constraint is introduced:

$$h \leq h_r \triangleq \frac{\ln(\Omega)}{\ln(p_{\min})}. \quad (6)$$

This constraint is a function of the minimum desired reliability requirement Ω and the worst channel condition p_{\min} . As the reliability requirement becomes stricter or the worst channel condition p_{\min} decreases, the constraint h_r decreases. Note that the logarithm function gives a negative numerical value when $0 \leq \Omega \leq 1$ and $0 \leq p_{\min} \leq 1$. We will use (5) and (6) in Section 6 to find the solution of Problem (1). Now, we turn our attention to the delay constraint.

5.2 Delay Constraint

The delay $D(h, \mu_c)$ between edge to sink is given by the sum of the delays experienced by a packet at each hop. There are two sources of delay:

- Time to wait before the first wake-up of a node in the next cluster: Let such a time be denoted with α_i for cluster i .
- Time to wait for clean channel: Since the Breath protocol uses CSMA/CA, a node spends a random time before sensing idle channel. Denote with ε_i such a time for cluster i .

By summing these delays per each hop, we obtain the delay model

$$D(h, \mu_c) = \sum_{i=1}^h (\alpha_i + \varepsilon_i). \quad (7)$$

In this equation, α_i is an exponentially distributed random variable whose intensity μ_c is the sum of the wake-up intensities of the nodes in the next cluster. Characterization of ε_i is more difficult, owing to the backoff mechanism of the CSMA/CA algorithm. However, we assume that the backoff time can be approximated by a Gaussian distribution whose average is matched with the average and standard deviation of a uniformly distributed random variable between 0 and $(M_{ca} + 1)(W - 1)$. Namely,

$$\varepsilon_i \in \mathcal{N} \left(\frac{(M_{ca} + 1)(W - 1)T_{ca}}{2}, \frac{(M_{ca} + 1)(W^2 - 1)T_{ca}^2}{12} \right).$$

We remark that this approximation significantly reduces the computation complexity when compared to an accurate analysis of the probability density function of the packet delay, which we have proposed in [50]. According to such an assumption, the delay $D(h, \mu_c)$ is approximated by a Gaussian random variable $\mathcal{N}(\mu_D, \sigma_D^2)$, where

$$\mu_D = \frac{h}{\mu_c} + \frac{h(M_{ca} + 1)(W - 1)T_{ca}}{2}, \quad (8)$$

$$\sigma_D^2 = \frac{h}{\mu_c^2} + \frac{h(M_{ca} + 1)(W^2 - 1)T_{ca}^2}{12}. \quad (9)$$

We validated these approximations by comparing the analysis to experimental results. Figs. 5a and 5b show the mean and variance of the delay given by (8) and (9) by experimental results, respectively. The analytical model

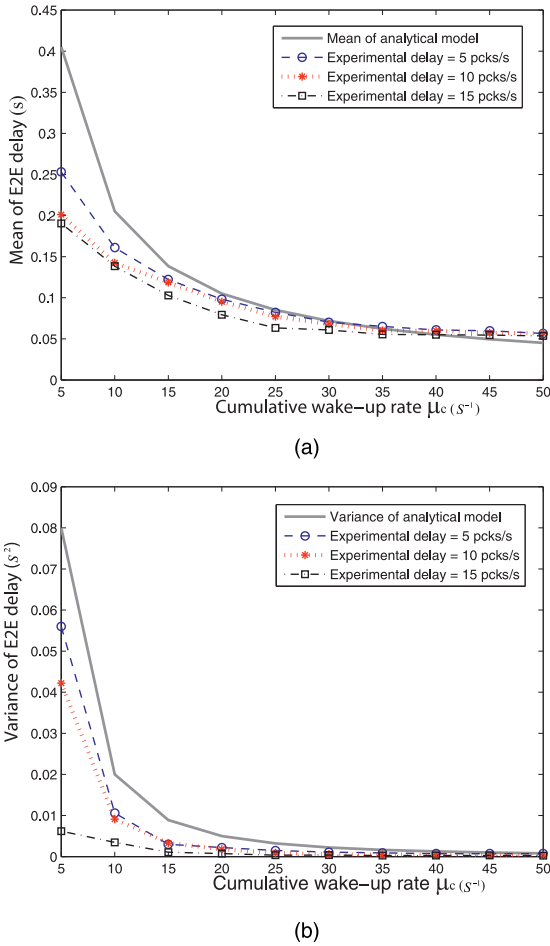


Fig. 5. Validation of average and variance of delay given by (8) and (9) by experimental results, respectively. The traffic rates $\lambda = 5, 10, 15$ pcks/s are considered w.r.t. wake-up rates μ_c from 5 to 50, $h = 2$ hops, and $N = 15$ nodes. (a) Average delay. (b) Variance of delay.

describes well the experimental data because it gives an upper bound for wake-up rates up to $35 s^{-1}$, and then the model underestimates the experimental result for less than 5 percent. These properties are quite useful for optimization purposes. Same dependence is found on h up to 4.

We are now in the position to express the delay constraint in Problem (1) by using (8) and (9) that we just derived:

$$\Pr[D \leq \tau] \approx 1 - Q\left(\frac{\tau - \mu_D}{\sigma_D}\right) \geq \Delta, \quad (10)$$

where $Q(x) = 1/\sqrt{2\pi} \int_x^\infty e^{-t^2/2} dt$ is the complementary standard Gaussian distribution. After some manipulations, it follows that (10) can be rewritten as

$$\mu_c \geq \frac{12 C_{d1} h + 2\sqrt{3 C_{d3} h [12 C_{d1}^2 + C_{d2}(h - C_{d3})]}}{12 C_{d1}^2 - C_{d2} C_{d3}},$$

where

$$\begin{aligned} C_{d1} &= \tau - \frac{h(M_{ca} + 1)(W - 1)T_{ca}}{2}, \\ C_{d2} &= h(M_{ca} + 1)(W^2 - 1)T_{ca}^2, \\ C_{d3} &= (Q^{-1}(1 - \Delta))^2. \end{aligned}$$

Since $T_{ca}^2 = 0.1024 \times 10^{-6}$ [13], and h, M_{ca}, W are positive integers, it follows that $T_{ca}^2 \ll h(M_{ca} + 1)(W^2 - 1)$. Then, $C_{d2} \ll C_{d1}$ and (10) is approximated by

$$\mu_c \geq f_d(h, \tau, \Delta) \triangleq \frac{2[h + Q^{-1}(1 - \Delta)\sqrt{h}]}{2\tau - h(M_{ca} + 1)(W - 1)T_{ca}}. \quad (11)$$

Inequality (11) has been derived under the additional constraint

$$h \leq h_d \triangleq \frac{2\tau}{(M_{ca} + 1)(W - 1)T_{ca}}. \quad (12)$$

This constraint is a function of the maximum allowable delay requirement τ and the parameters of CSMA/CA algorithm such as the maximum number of CSMA/CA transmission tries M_{ca} , the maximum number of random backoff W and the unit of backoff period T_{ca} . As the delay requirement τ becomes stricter or the parameters of CSMA/CA algorithm M_{ca}, W, T_{ca} increase, the constraint h_d decreases. We will use (11) and (12) in Section 6 to find the solution of the optimization problem (1). Now, we investigate the total energy consumption.

5.3 Energy Consumption

The total energy consumption is

$$E_{tot}(h, \mu_c) = E_{pck}(h, \mu_c) + E_{wu}(h, \mu_c), \quad (13)$$

where $E_{pck}(h, \mu_c)$ is the total energy for transmission and reception of data packets and $E_{wu}(h, \mu_c)$ is the energy consumption for wake-up, listening, and beaconing during a time T , which we characterize in Sections 5.3.1 and 5.3.2, respectively.

5.3.1 Data Packet Communication Energy

Assuming h hops, and recalling that edge node emits λ pcks/s,

$$E_{pck}(h, \mu_c) = T\lambda \sum_{i=1}^h \left[Q_m(d_i) + \frac{A_{rx}}{\mu_c} + E_{ca}(\mu_c) + E_r \right], \quad (14)$$

where E_r accounts for the fixed cost of the RF circuit for the reception of a data packet. The term $Q_m(d_i)$ is the energy consumption for radio transmission, where d_i is the transmission distance to which a data packet has to be sent. The term $E_{ca}(\mu_c)$ is the energy spent during the CSMA/CA state.

The energy model given by (14) is derived under the assumption that all packets generated at the edge nodes reach the sink. Obviously, some packet may be lost before reaching the sink, therefore (14) gives an upper bound on the energy consumption. This is reasonable, since our goal is the minimization of the cost function.

The energy spent for radio transmission is a function of the radio power used to transmit packets:

$$Q_m(d_i) = V I(P_t(d_i)) t_m, \quad (15)$$

where V is the voltage consumption of the RF circuit at the node, t_m is the transmission time of a data packet, $I(P_t(d_i))$ is the current consumption of the electronic circuit needed to transmit packets at radio power $P_t(d_i)$, and d_i is the distance from the transmitter which a packet must reach to

with some desired probability. The relation between the current consumption and radio power depends on the hardware platform. For Tmote sensors, it holds that [51]

$$I(P_t(d_i)) \approx -19P_t(d_i)^4 + 53P_t(d_i)^3 - 53P_t(d_i)^2 + 29P_t(d_i) + 8.7.$$

Given this approximation, minimization of $Q_m(d_i)$ is achieved by minimizing $P_t(d_i)$. $P_t(d_i)$ can be minimized by computing the minimum radio power that ensures packets to reach a given distance with a given probability, as we see next.

The optimal transmit power is derived by considering the distribution of the Signal to Interference plus Noise Ratio (SINR). By imposing a requirement p_{con} on the probability of successful packet reception at a distance d_k from node k , we can translate the requirement on the average SINR, thus obtaining a bound $\bar{\gamma}_c$ on such an average SINR. From this, we can then derive the transmit radio power necessary to successfully receive packets at a distance d_k with probability p_{con} . It follows that the minimum transmit power is [1]

$$P_t(d_k)_{\text{dB}} = \bar{\gamma}_c_{\text{dB}} + \text{PL}(d_0)_{\text{dB}} + 10 \beta_k \log_{10} \frac{d_k}{d_0} + P_n_{\text{dB}} - \frac{\ln 10}{20} \sigma_{\gamma_k}^2_{\text{dB}},$$

where $\text{PL}(d_0)_{\text{dB}}$ is the path loss at a reference distance d_0 , β_k is the path loss decay constant, P_n is the noise floor, and σ_{γ_k} is the variance of the SINR (see [1] for details). We remark that the power $P_t(d_k)_{\text{dB}}$ minimizes the energy spent for radio transmission in (15). Notice that the actual packet reception probability p_i may fluctuate around p_{con} due to the delay of this power control and the limited maximum transmit power.

In the following, we characterize E_{ca} . Consider the energy spent for transmission of a data packet in the i th cluster. Let E_{ca} is the energy spent by a node to check the channel status by the CSMA/CA algorithm upon the reception of a beacon. This energy, which is due to CCA, is dependent on the maximum number of tries M_{ca} . We have two situations: the number of contending nodes n attempting to transmit a data packet is less than M_{ca} , or the number of contending nodes is larger than M_{ca} . If $n < M_{\text{ca}} + 1$, all nodes will succeed to sense a clean channel with the energy $E_{\text{ca}1}(n)$, otherwise we need to consider the transmission success and failure probabilities to perform CCA with the energy $E_{\text{ca}2}(n)$, which is the function of the busy channel probability $\nu(k)$ conditioned on k contending nodes defined by (2), see the details in [1].

By summing the two energy components $E_{\text{ca}1}(n)$, $E_{\text{ca}2}(n)$, the average energy consumption spent by the CSMA/CA is

$$E_{\text{ca}}(\mu_c) = \sum_{n=1}^{\infty} \psi_{\text{sb}}(\mu_c, n) [E_{\text{ca}1}(n) u(M_{\text{ca}} - n) + E_{\text{ca}2}(n) u(n - M_{\text{ca}} - 1)], \quad (16)$$

where $\psi_{\text{sb}}(\mu_c, n)$ is the probability to have n contending nodes in a cluster given by (3) and $u(x) = 1$ if $x \geq 0$, whereas $u(x) = 0$ otherwise.

5.3.2 Control Signaling Energy

A node randomly cycles between an awake state and a sleep state. Each time a node wakes up, it spends an energy given by the power needed to wake-up A_w during the wake-up time T_w , plus the energy to listen for the reception of a data packet within a maximum time T_{ac} . After a node wakes up, it transmits a beacon to the next cluster. Let the wireless channel loss probability be $1 - p_i$ of cluster i , then nodes of cluster $i - 1$ have to wake-up on average $1/p_i$ times to create the effect of a single wake-up so that a transmitter node successfully receives a beacon. Recalling that there are h hops and a cumulative wake-up rate per cluster μ_c , the total cost in a time T for wake-ups and beaconing is

$$E_{\text{wu}}(h, \mu_c) = T \mu_c \sum_{i=1}^h \frac{1}{p_i} [Q_b(d_i) + A_w T_w + A_{\text{rx}}(T_{\text{ac}} - T_w)], \quad (17)$$

where $Q_b(d_i)$ is the expected energy consumption to transmit a beacon message at the distance d_i .

5.3.3 Total Energy Consumption

Here, we put together the energy analysis developed in the previous two sections. The total energy consumption is

$$E_{\text{tot}}(h, \mu_c) = T \lambda \left[Q_m \left(\frac{S}{h-1} \right) + Q_m \left(\frac{S}{h-1} \right) (h-1) \times u(h-1) + h \left(\frac{A_{\text{rx}}}{\mu_c} + E_{\text{ca}}(\mu_c) + E_r \right) \right] + \frac{T \mu_c}{p_{\text{min}}} \left[2Q_b \left(\frac{S}{h-1} \right) + Q_b \left(\frac{2S}{h-1} \right) (h-2) \times u(h-2) + h(A_w T_w + A_{\text{rx}}(T_{\text{ac}} - T_w)) \right], \quad (18)$$

where we upper bounded (14) and (17) by considering the worst distance to which data and beacon packets must be sent, which are $S/(h-1)$ and $2S/(h-1)$, and the worst reception probability p_{min} .

Fig. 6 shows the energy given by (18) as a function of the number of hops h over different wake-up rates μ_c . The total energy consumption increases with h given μ_c because increasing h implies higher wake-up rates per node for a given number of total nodes present in the network. In other words, increasing the number of hops is energy-inefficient. Observe also that a low wake-up rate does not minimize the total energy consumption, because of the longer waiting time to receive a beacon message that such a rate causes. Hence, there is a trade-off between the energy consumption for wake-up and waiting to get a beacon message. We explore this trade-off for optimization problem in the following section.

6 OPTIMAL PROTOCOL PARAMETERS

In this section, we give the optimal protocol parameters used by Breath. Consider the reliability and delay constraints, and the total energy consumption as investigated in Sections 5.1, 5.2, and 5.3. The optimization problem (1) becomes

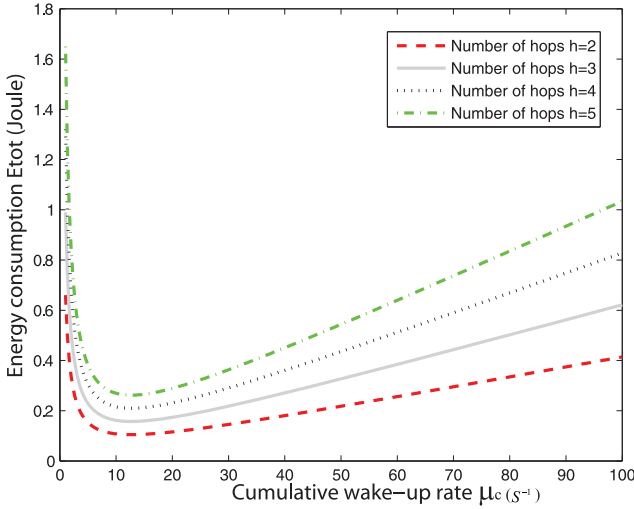


Fig. 6. Total energy consumption given by (18) for a different number of hops ($h = 2, 3, 4, 5$) over wake-up rates μ_c from 1 to 100 in traffic rate ($\lambda = 5$ pcks/s) and $N = 15$ nodes.

$$\begin{aligned} \min_{h, \mu_c} \quad & E_{\text{tot}}(h, \mu_c) \\ \text{s.t.} \quad & \mu_c \geq \max(f_r(h, \Omega), f_d(h, \tau, \Delta)), \\ & 2 \leq h \leq \min(h_r, h_d), \\ & \mu_{\min} \leq \mu_c \leq \mu_{\max}, \end{aligned} \quad (19)$$

where the first constraint comes from (5) and (6), and the second from (11) and (12). We assume that this problem is feasible. Infeasibility means that for any $h = 2, \dots, \min(h_r, h_d)$, then $\mu_c \geq \max(f_r(h, \Omega), f_d(h, \tau, \Delta)) > \mu_{\max}$, namely it is not possible to guarantee the satisfaction of the reliability and delay constraint given the application requirements. This means that the application requirements must be relaxed, so that feasibility is ensured and the problem can be solved. The solution of this optimization problem, h^* and μ_c^* , is derived in the following.

By using the numerical values given for the Tmote sensors [13] for all the constants in the optimization problem, we see that the cost function of Problem (19) is increasing in h and convex in μ_c . This allows us to derive the optimal solution in two steps: for each value of $h = 2, \dots, \min(h_r, h_d)$, the cost function is minimized for μ_c , achieving $\mu_c^*(h)$. Then, the optimal solution is found in the pair $h, \mu_c^*(h)$ that gives the minimum energy consumption. We describe this procedure next.

Let h be fixed. From the properties the cost function of Problem (19), the optimal solution $\mu_c^*(h)$ is attained either at the minimum of the cost function or at the boundaries of the feasibility region given by the requirements on μ_c . The minimum of the cost function can be achieved by taking its derivative with respect to μ_c . To obtain this derivative in an explicit form, we assume that CSMA/CA energy consumption can be approximated by a constant value since the numerical value is smaller than other factors. Under this assumption, the minimization by the derivative is approximated by

$$\begin{aligned} \mu_e(h) = & (p_{\min} \lambda A_{\text{tx}})^{\frac{1}{2}} \left[\frac{h-2}{h} Q_b \left(\frac{2S}{h-1} \right) u(h-2) \right. \\ & \left. + \frac{2}{h} Q_b \left(\frac{S}{h-1} \right) + A_w T_w + A_{\text{rx}} (T_{\text{ac}} - T_w) \right]^{-\frac{1}{2}}. \end{aligned} \quad (20)$$

In Fig. 7, we check the validity of this approximation. The figure reports the approximated minimum of the cost function as obtained by (20) compared to the wake-up rate that minimizes the actual energy consumption as obtained by a numerical minimization algorithm. The approximation is tight because the error is less than 2 percent.

Equation (20) tells us that, provided h , an optimal solution $\mu_c^*(h)$ is given by $\mu_e(h)$ if $\mu_e(h)$ is in the feasible range $\mu_{\min} \leq \mu_e(h) \leq \mu_{\max}$ and $\mu_e(h)$ satisfies the two constraints of both reliability and delay requirement $\mu_e(h) \geq \max(f_r(h, \Omega), f_d(h, \tau, \Delta))$. If $\mu_e(h)$ is not in the feasible range ($\mu_e(h) \geq \mu_{\max}$) \cup ($\mu_e(h) \leq \mu_{\min}$) or $\mu_e(h)$ does not satisfy two the constraints $\mu_e(h) \leq \max(f_r(h, \Omega), f_d(h, \tau, \Delta))$ but $\max(f_r(h, \Omega), f_d(h, \tau, \Delta))$ is in the feasible range $\mu_{\min} \leq \max(f_r(h, \Omega), f_d(h, \tau, \Delta)) \leq \mu_{\max}$, then an optimal solution is given by $\max(f_r(h, \Omega), f_d(h, \tau, \Delta))$. Otherwise, an optimal solution is given by μ_{\max} as the best-effort mode since the network does not satisfy the constraints. Therefore, for any $h = 2, \dots, \min(h_r, h_d)$, we compute $\mu_c^*(h)$. Then, the optimal solution h^* and μ_c^* is given by the pair $\mu_c^*(h), h$ that minimizes the cost function. This procedure to compute the optimal solution is illustrated by Algorithm 1, which summarizes one of the main contributions of this paper.

Algorithm 1. Algorithm for the computation of the optimal solution of Problem (19)

```

Input: Requirements  $\Omega, \tau, \Delta$ , feasible range  $\mu_{\min}, \mu_{\max}$ 
Output:  $h^*, \mu_c^*$ 
begin
     $h^* \leftarrow 2$ ;
    if ( $\mu_e(h^*) \geq \max(f_r(h^*, \Omega), f_d(h^*, \tau, \Delta))$ )  $\cap$  ( $\mu_{\min} \leq \mu_e(h^*) \leq \mu_{\max}$ ) then
         $\mu_c^* \leftarrow \mu_e(h^*)$ ;
        SAT  $\leftarrow$  true;
    else if [ $(\mu_e(h^*) \geq \mu_{\max}) \cup (\mu_e(h^*) \leq \mu_{\min}) \cup (\mu_e(h^*) \leq \max(f_r(h^*, \Omega), f_d(h^*, \tau, \Delta)))$ ]  $\cap$  ( $\mu_{\min} \leq \max(f_r(h^*, \Omega), f_d(h^*, \tau, \Delta)) \leq \mu_{\max}$ ) then
         $\mu_c^* \leftarrow \max(f_r(h^*, \Omega), f_d(h^*, \tau, \Delta))$ ;
        SAT  $\leftarrow$  true;
    else
         $\mu_c^* \leftarrow \mu_{\max}$ ;
        SAT  $\leftarrow$  fail;
     $E \leftarrow E_{\text{tot}}(h^*, \mu_c^*)$ ;
    for  $h \leftarrow 3$  to  $\min(h_r, h_d)$  do
        if ( $\mu_e(h) \geq \max(f_r(h, \Omega), f_d(h, \tau, \Delta))$ )  $\cap$  ( $\mu_{\min} \leq \mu_e(h) \leq \mu_{\max}$ ) then
             $\mu_{\text{tmp}} \leftarrow \mu_e(h)$ ;
            SAT  $\leftarrow$  true;
        else if [ $(\mu_e(h) \geq \mu_{\max}) \cup (\mu_e(h) \leq \mu_{\min}) \cup (\mu_e(h) \leq \max(f_r(h, \Omega), f_d(h, \tau, \Delta)))$ ]  $\cap$  ( $\mu_{\min} \leq \max(f_r(h, \Omega), f_d(h, \tau, \Delta)) \leq \mu_{\max}$ ) then
             $\mu_{\text{tmp}} \leftarrow \max(f_r(h, \Omega), f_d(h, \tau, \Delta))$ ;
            SAT  $\leftarrow$  true;
        else
             $\mu_{\text{tmp}} \leftarrow \mu_{\max}$ ;
            SAT  $\leftarrow$  fail;
         $E_{\text{tmp}} \leftarrow E_{\text{tot}}(h, \mu_{\text{tmp}})$ ;
        if ( $E > E_{\text{tmp}}$ )  $\cap$  SAT then
             $h^* \leftarrow h$ ;
             $\mu_c^* \leftarrow \mu_{\text{tmp}}$ ;
             $E \leftarrow E_{\text{tmp}}$ ;
    end
    
```

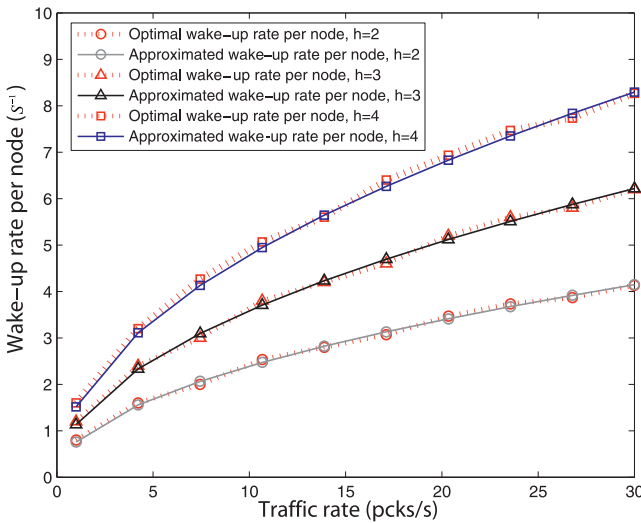


Fig. 7. Wake-up rate that minimizes the total energy consumption and approximated wake-up rate as obtained by (20) for different number of hops ($h = 2, 3, 4$), traffic rates λ from 1 to 30 pkcs/s, and $N = 15$. The y-axis was normalized by 15.

7 ADAPTATION MECHANISMS

In the previous sections, we showed how to determine the optimal number of clusters and cumulative wake-up rate by solving an optimization problem. Here, we present in detail some adaptation algorithms that the sink must run to determine correctly h^* and μ_c^* as the traffic rate and channel conditions changes. These algorithms allow us to adapt the protocol parameters to the traffic rate and channel condition without high message overhead.

7.1 Traffic Rate and Channel Estimation

The sink node estimates the traffic rate λ and the worst channel probability p_{\min} of the network. To estimate the global minimum of the worst channel condition, each p_i should be estimated at a local node and sent to the sink for each link of the path $i = 1, \dots, h$. This might increase considerably the packet size. To avoid this, we propose the following strategy. Consider a relay node of the i th cluster. It estimates p_i by the signal of the beacon packet. Then, the nodes compares p_i with the channel condition information carried by the received data packet and selects the minimum. This minimum is then encoded in the data packet and sent with it to the next-hop node. After the sink node retrieves the channel condition of the route by receiving a data packet, it computes an average of the worst channel conditions among the last received data packets. Using this estimate, the sink solves the optimization problem running Algorithm 1. Afterwards, the return value of the algorithm, h^* and μ_c^* , can be piggybacked on beacons that the sink sends toward the relays closer to the sink. Then, these protocol parameters are forwarded when the nodes wake up and send beacons to the next cluster toward the edge nodes. During the initial state, nodes set $h = 2$ before receiving a beacon.

7.2 Wake-Up Rate and Radio Power Adaptation

Once a cluster received μ_c^* , each node in the cluster must adapt its wake-up rate so that the cluster generates such a

cumulative wake-up rate. We consider the natural solution of distributing μ_c^* equally between all nodes of the cluster. Let μ_k be the wake-up rate of node k , and suppose that there are l nodes in a cluster. The fair solution is $\mu_k = \mu_c^*/l$ for any node. However, a node does not know and cannot estimate efficiently the number of nodes in its cluster.

To overcome this problem, we follow the same approach proposed in [36], where an Additive Increase and Multiplicative Decrease (AIMD) algorithm leads to a fair distribution of the wake-up duties within a single cluster. Specifically, each node that is waiting to forward a data packet observes the time before the first wake-up in the forwarding region. Starting from this observation, it estimates the cumulative wake-up rate $\tilde{\mu}_c$ of the forwarding region and it compares it with the optimal value of the wake-up rate μ_c^* when a node receives a beacon. Note that the node retrieves information on h^*, μ_c^* , and location information of the beacon node. If $\tilde{\mu}_c < \mu_c^*$, the node sends by the data packet an Additive Increase (AI) command for the wake-up rate of next-hop cluster, else it sends a Multiplicative Decrease (MD) command. Furthermore, the node updates the probability of successful transmission p_i based on the channel information using the RSSI and distance information d_k between its own location and beacon node. After the node updates the channel condition estimation, it sets the data packet transmission power to $P_i(d_k)$, and encodes the channel estimation in the packet as described in Section 7.1. If a data packet is received, the node retrieves information on wake-up rate update: if AI, then $\mu_k = \mu_k + \theta$, else $\mu_k = \mu_k/\phi$, where θ and ϕ are control parameters. From experimental results, we obtained that $\theta = 3$ and $\phi = 1.05$ achieve good performance. The command on the wake-up rate variation is piggybacked on data packets and does not require any additional message.

However, this approach may generate a load balancing problem because of different wake-up rates among relays within a short period. Load balancing is a critical issue, since some nodes may wake up at higher rate than desired rate of other nodes, thus wasting energy. To overcome this situation, each relay node runs a simple reset mechanism. We assign an upper and lower bound to the wake-up rate for each node. If the wake-up rate of a node is larger than the upper bound $(1 + \xi)\mu_c^*(h^* - 1)/N$ or is smaller than the lower bound $(1 - \xi)\mu_c^*(h^* - 1)/N$, then a node resets its wake-up rate to $\mu_c^*(h^* - 1)/N$, where ξ assumes a small value and $(h^* - 1)/N$ is an estimation of the number of nodes per cluster.

8 FUNDAMENTAL LIMITS

Understanding the fundamental limits of Breath is critical for its appropriate use. This section focuses on the minimum number of relays required to support the protocol, and the minimum delay that can be set by the application.

8.1 Minimum Number of Nodes per Cluster

The minimum number n_{\min} of nodes per cluster to support the protocol with given reliability and delay requirements is

$$n_{\min} \geq \frac{\mu_c^*}{\mu_{k,\max}}$$

where $\mu_{k,\max}$ is the maximum wake-up rate of node k . By considering the worst active time for the duty cycle, we have

$$\mu_{k,\max} = (T_{ac} + T_{be})^{-1},$$

where $T_{ac} = 30$ ms is the maximum listening time to receive a data packet and $T_{be} = 500$ ms is the maximum waiting time before receiving a beacon [13].

8.2 Minimum Delay

The minimum delay that the application can set is achieved by considering a very high wake-up rate per cluster. This minimizes the waiting time before receiving a beacon. Hence, by summing the delays of the CSMA/CA state and physical limits of the wireless channel, the minimum delay is

$$\tau \geq h [2(M_{ca} + 1)(W - 1)T_{ca} + 2T_{prop} + t_m + t_b],$$

where the first term is the maximum delay of CSMA/CA state, T_{prop} is propagation delay, and t_m and t_b are, respectively, the transmission delay of data and beacon packets. Since $T_{prop} = 0.875$ ms, $t_m = 1.5$ ms, and $t_b = 0.64$ ms [13], they can be basically ignored because they are negligible with respect to other delays.

9 EXPERIMENTAL IMPLEMENTATION

In this section, we provide an extensive set of experiments to validate the Breath protocol. The experiments enable us to assess Breath in terms of reliability and delay in the packet transmission, and energy consumption of the network both in stationary and transitional condition. The protocol was implemented on a testbed of Tmote sensors [13], and was compared with a standard implementation of the IEEE 802.15.4 [5], as we discuss next.

We consider a typical indoor environment with concrete walls. The experiments were performed in a static propagation (AWGN) and time-varying fading environment (Rayleigh), respectively:

- AWGN environment: nodes and surrounding objects were static, with minimal time-varying changes in the wireless channel. In this case, the wireless channel is well described by an Additive White Gaussian Noise (AWGN) model.
- Rayleigh fading environment: obstacles were moved within the network, along a line of 20 m. Furthermore, a metal object was put in front of the edge node, so the edge node and the relays were not in line-of-sight. The edge node was moved on a distance of few tens of centimeters.

A node acted as edge node and generated packets periodically at different rates ($\lambda = 5, 10,$ and 15 pcks/s). Fifteen relays were placed to mimic the topology in Fig. 1. The edge node was at a distance of 20 m far from the sink. The sink node collected packets and then computed the optimal solution using Algorithm 1. The delay requirement was set to $\tau = 1$ s and the reliability to $\Omega = 0.9$ and 0.95 . In other words, we imposed that packet must reach destination within 1 s with a probability of Ω . These requirements were chosen as representative for control applications.

We compared Breath against an implementation of the unslotted IEEE 802.15.4 [5] standard, which is similar to the

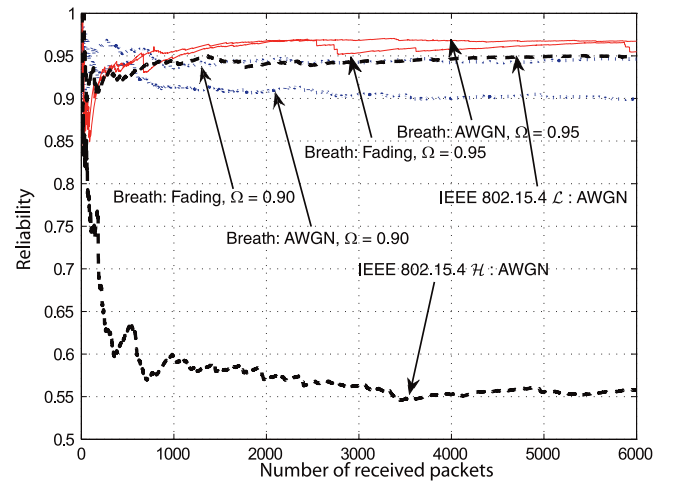


Fig. 8. Convergence over the number of received packets of the reliability for IEEE 802.15.4 \mathcal{L}, \mathcal{H} in AWGN, and Breath with reliability requirements $\Omega = 0.9, 0.95$ and traffic rates $\lambda = 10$ pcks/s in AWGN and Rayleigh fading environments.

randomized MAC that we use in this paper. In such an IEEE 802.15.4 implementation, we set nodes to a fixed sleep schedule, defined by CT_{ac} where C is integer number (recall that T_{ac} is the maximum node listening time in Breath). We defined the case \mathcal{L} (low sleep), where the IEEE 802.15.4 implementation is set with $C = 1$, whereas we defined the case \mathcal{H} (high sleep) by setting $C = 4$. The case \mathcal{H} represents a fair comparison between Breath and the IEEE 802.15.4, while in the case \mathcal{L} nodes are let to listen much longer time than nodes in Breath. The power level in the IEEE 802.15.4 implementation where set to -5 dBm. We set the IEEE 802.15.4 protocol parameters to default values $macMinBE = 3, aMaxBE = 5, macMaxCSMABackoffs = 4$. Details follow in the sequel.

9.1 Protocol Behavior for Stationary Requirements

In this section, we investigate the performance of Breath about the reliability, average delay, and energy consumption that can be achieved in a stationary configuration of the requirements, i.e., during the experiment there was not change of application requirements. Data were collected out of 10 experiments, each lasting 1 hour.

9.1.1 Reliability

Fig. 8 indicates that the network converges by Breath to a stable error rate lower than $1 - \Omega$ and hence satisfies the required reliability with traffic rate $\lambda = 10$ pcks/s, the delay requirement $\tau = 1$ s, and $\Omega = 0.9, 0.95$. IEEE 802.15.4 \mathcal{H} in AWGN channel provides the worse performance than the other protocols because of lower wake-up rate. Observe that $\Omega = 0.9$ in Rayleigh fading environment gives the better reliability than $\Omega = 0.95$ in AWGN channel due to higher wake-up rate to compensate the fading channel condition. Notice that the higher fluctuation of reliability between the number of received packets 2,500 and 2,800 for Rayleigh fading environment with $\Omega = 0.95$ is due to deep attenuations in the wireless channel.

Fig. 9 shows the reliability of Breath and IEEE 802.15.4 \mathcal{L}, \mathcal{H} as a function of the reliability requirement $\Omega = 0.9, 0.95$ and traffic rate $\lambda = 5, 10, 15$ pcks/s in AWGN and Rayleigh fading

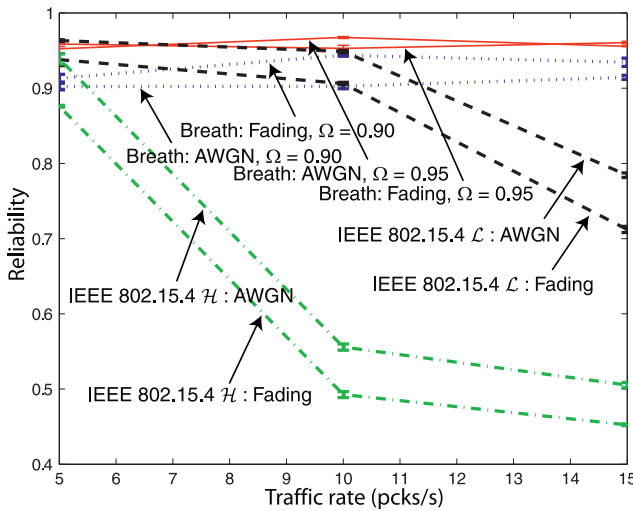


Fig. 9. Reliability in IEEE 802.15.4 \mathcal{L} , \mathcal{H} , and Breath with requirement $\Omega = 0.9, 0.95$ for traffic rates $\lambda = 5, 10, 15$ pcks/s in AWGN and Rayleigh fading environments. The vertical bars indicate the standard deviation as obtained out of 10 experimental runs of 1 hour each.

environments, with the vertical bars indicating the standard deviation as obtained out of 10 experimental runs of 1 hour each. Observe that the reliability is stable around the required value for Breath, and this holds for different traffic rates and environments. However, the IEEE 802.15.4 \mathcal{L} and \mathcal{H} do not ensure the reliability satisfaction for large traffic rates. Specifically, the IEEE 802.15.4 \mathcal{H} shows poor reliability in any case, and performance worsen as the environment moves from the AWGN to the Rayleigh fading. Furthermore, even though the IEEE 802.15.4 \mathcal{L} imposes that nodes wakes up more often, it does not guarantee a good reliability in higher traffic rates. The reason is found in the sleep schedule of the IEEE 802.15.4 case, which is independent of traffic rate and wireless channel conditions. The result is that the fixed sleep schedule is not feasible to support high traffic and time-varying wireless channels. Moreover, the fixed sleep schedule does not guarantee a uniform distribution of cumulative

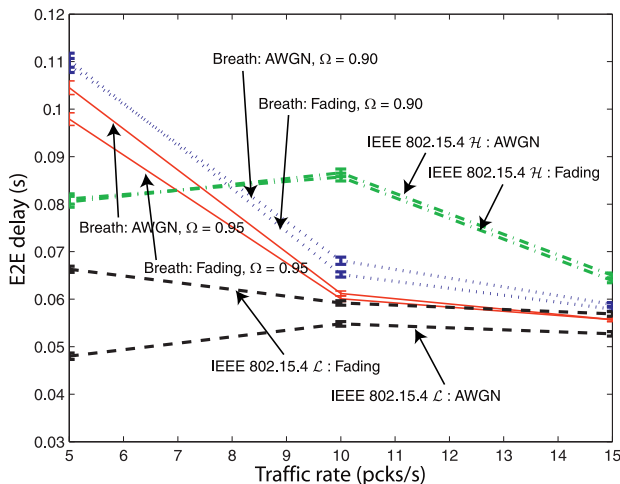


Fig. 10. Temporal average of the delay of Breath and IEEE 802.15.4 \mathcal{L} , \mathcal{H} with reliability requirement $\Omega = 0.9, 0.95$ and delay requirement $\tau = 1$ s over traffic rates $\lambda = 5, 10, 15$ pcks/s in AWGN and Rayleigh fading environment. The vertical bars indicate the standard deviation as obtained out of 10 experimental runs of 1 hour each.

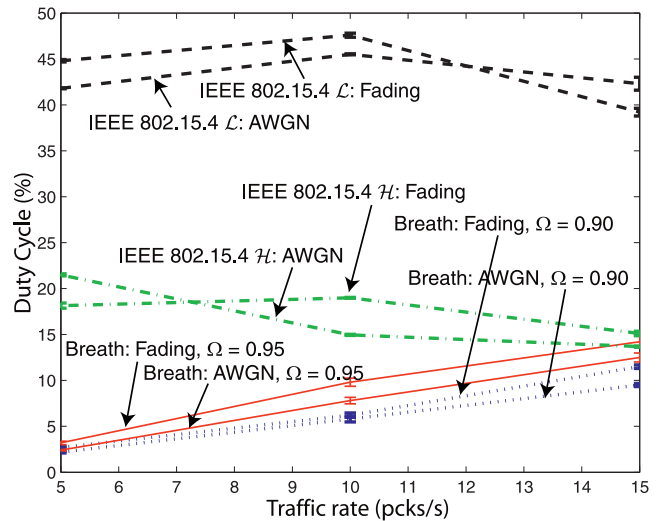


Fig. 11. Sample average of the node's duty cycle in IEEE 802.15.4 \mathcal{L} , \mathcal{H} , and Breath with reliability requirement $\Omega = 0.9, 0.95$ for traffic rates $\lambda = 5, 10, 15$ pcks/s in AWGN and Rayleigh fading environments.

wake-up rate within certain time in a cluster, which means that there may be congestions. On the contrary, Breath presents an excellent behavior in any situations of traffic load and channel condition.

9.1.2 Delay

In Fig. 10, the sample average of the delay for packet delivery of Breath, IEEE 802.15.4 \mathcal{L} and \mathcal{H} are plotted as a function of the reliability requirement Ω and traffic rate λ in AWGN and Rayleigh fading environments, with the vertical bars indicating the standard deviation of the samples around the average. The sample variance of the delay exhibits similar behavior as the average. The delay meets quite well the constrains. Observe that delay decreases as the traffic rate rises. Because Breath increases linearly the wake-up rate of nodes when the traffic rate increases (see (5)). The delay is larger for worse reliability requirements. Note that (5) increases as the reliability requirement Ω increases. IEEE 802.15.4 \mathcal{L} has lower delay than IEEE 802.15.4 \mathcal{H} because nodes have higher wake-up time. Breath has an intermediate behavior with respect to IEEE 802.15.4 \mathcal{L} and \mathcal{H} after $\lambda = 7$. From these experimental results, we conclude that both Breath and the IEEE 802.15.4 meet the delay requirement. However, notice that the delay for the IEEE 802.15.4 is related to only packet successfully received, which may be quite few.

9.1.3 Duty Cycle

In this section, we study the energy consumption of the nodes. As energy performance indicator, we measured the node's duty cycle, which is the ratio of the active time of the node to the total experimental time. Obviously, the lower is the duty cycle, the better is the performance of the protocol on energy consumption.

Fig. 11 shows the sample average of duty cycle of Breath, IEEE 802.15.4 \mathcal{L} and \mathcal{H} with respect to the traffic rates $\lambda = 5, 10, 15$ pcks/s and $\Omega = 0.9, 0.95$, both in AWGN and Rayleigh fading environments, with the vertical bars indicating the standard deviation of the samples. Note that the IEEE 802.15.4 \mathcal{L} and \mathcal{H} do not exhibit a clear relationship

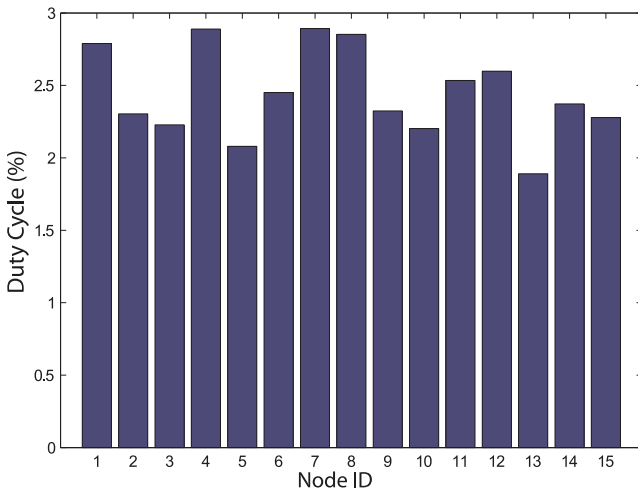


Fig. 12. Distribution of the duty cycle in each node with $N = 15$ relays. The reliability requirement is $\Omega = 0.95$ and traffic rate is $\lambda = 5$ pcks/s.

with respect to traffic rate and have almost flat duty cycle around 42 and 18 percent, respectively, because of fixed sleep time. Considering Breath, observe that the duty cycle increases linearly with the traffic rate for a given reliability requirement, which is explained by (5). Since Breath minimizes the total energy consumption on the base of a trade-off between wake-up rate and waiting time of beacon messages (recall the analysis in Section 5.3), lower wake-up rates do not guarantee lower duty cycle. Observe that choosing a lower active time for the nodes of the IEEE 802.15.4 implementation would obviously obtain energy savings comparable with Breath, however, the reliability of the IEEE 802.15.4 implementation would be heavily affected (recall Fig. 9). In other words, ensuring a duty cycle for the IEEE 802.15.4 implementation comparable with Breath would be very detrimental with respect to the reliability.

Fig. 12 shows the experimental results for the duty cycle of each relay node for $\lambda = 5$ pcks/s and $\Omega = 0.95$. A fair uniform distribution of the duty cycles among all nodes of the network is achieved. This is an important result, because the small variance of the wake-up rate among nodes signifies that duty cycle and load are uniformly distributed, with obvious advantages for the network lifetime.

Fig. 13 reports the case of several networks, where each network corresponds to a number of relays between the edge and the sink in an AWGN environment. From the figure, it is possible to evaluate how much Breath extends the network lifetime compared to the IEEE 802.15.4 \mathcal{L} and \mathcal{H} . Observe that the duty cycle is proportional to the density of nodes. Hence, the network lifetime is extended fairly by adding more nodes without creating load balancing problems.

Finally, recall that Breath uses a radio power control (Section 5.3.1), so that further energy savings are actually obtained with respect to the IEEE 802.15.4 implementation.

9.2 Protocol Behavior for Time-Varying Requirements

Performance of Breath protocol is based on the application requirements and estimation of the channel condition. In this section, we investigate the dynamic adaptation of Breath when the reliability Ω and delay τ requirements change. Fig. 14 shows the dynamic adaptability of reliability, packet

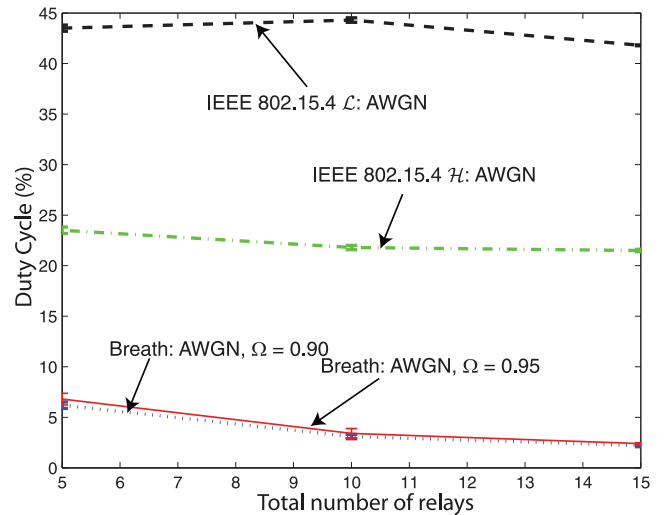


Fig. 13. Sample average of the duty cycle in IEEE 802.15.4 \mathcal{L} , \mathcal{H} , and Breath with reliability requirement $\Omega = 0.9, 0.95$ and traffic rates $\lambda = 5, 10, 15$ pcks/s in AWGN environment for different networks, each with a different number of relaying nodes. The vertical bars indicate the standard deviation as obtained out of 10 experimental runs of 1 hour each.

delay, and energy consumption when the requirements are changed for given traffic rate and number of nodes. Figs. 14a, 14c, 14e and 14b, 14d, 14f present the behavior of the reliability, packet delay, and average active time when the reliability and delay requirements change, respectively. More specifically, the average active time is defined as the average time nodes are active. We observe performance in terms of reliability, delay, and average active time in Figs. 14a, 14c, and 14e for a reliability requirement variation. When Ω increases from 0.9 to 0.95 at a time corresponding to the number of received packets is 2,000 the reliability converges to 0.95. At the same time, packet delay decreases and average active time increase since optimal wake-up rate increases to guarantee the higher reliability requirement. Analogously, Breath adapts the network by considering the delay requirement variation in Figs. 14b, 14d, and 14f. It is clear that the average delay is under 60 ms since we consider the distribution of the delay probability. The average active time increases when the delay requirement changes due to a higher optimal wake-up rate. Hence, in the optimization problem, the delay requirement 60 ms gives a stricter constraint than reliability constraint computed at a requirement of 0.9. From this analysis, we can conclude that Breath adaptively achieves its target (i.e., minimization of power consumption) while guaranteeing the reliability and delay requirements. Furthermore, we observe clearly the trade-off between the application requirements and energy consumption, i.e., as application requirements become strict, energy consumption increases.

10 CONCLUSIONS

We designed and implemented Breath, a protocol that is based on a system-level approach to guarantee explicitly reliability and delay requirements in wireless sensor networks for control and actuation applications. The protocol considers duty cycle, routing, MAC, and physical layers all together to maximize the network lifetime by

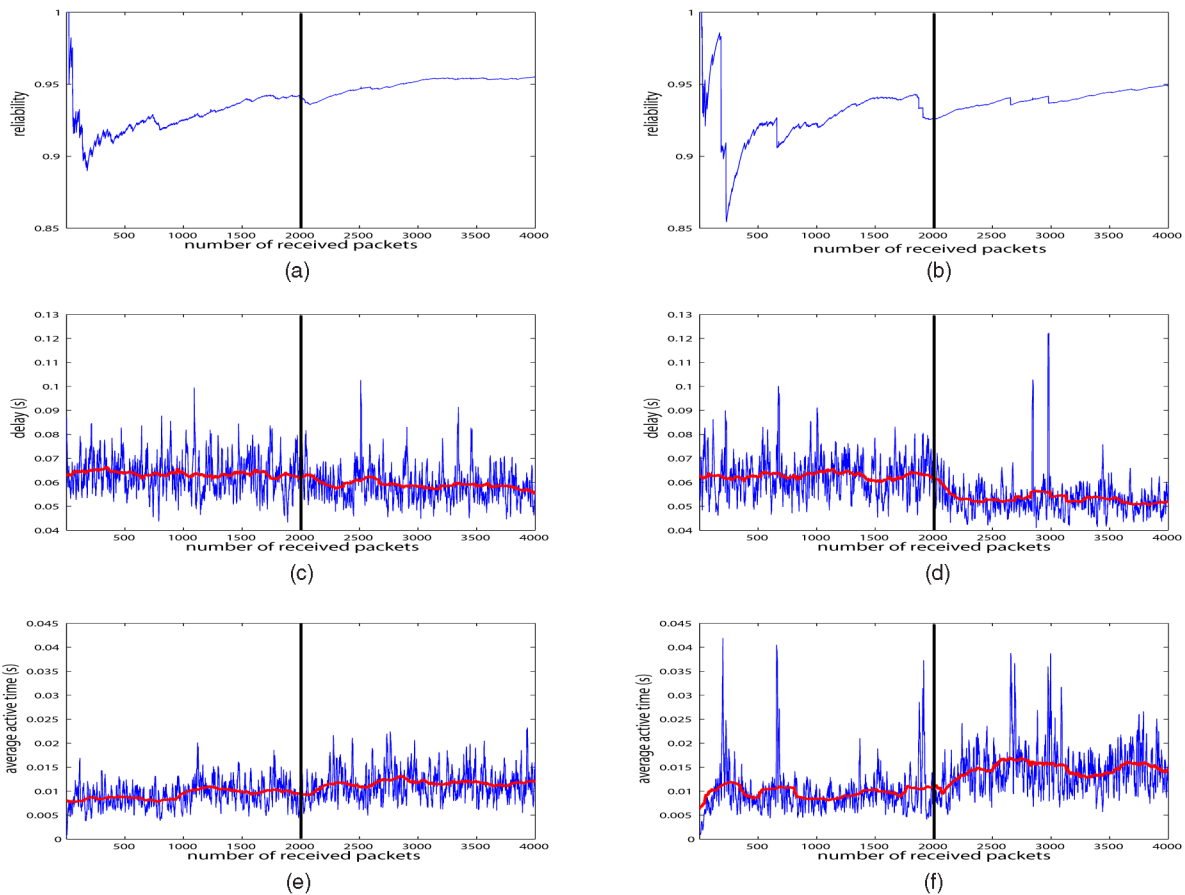


Fig. 14. Reliability, packet delay, and active time behavior for $\lambda = 10$ pcks/s, $N = 15$ when reliability and delay requirements vary from $\Omega = 0.9$ to $\Omega = 0.95$ and from $\tau = 1$ s to $\tau = 60$ ms at a time instant corresponding to the number of received packets is 2,000 in AWGN environment. (a) Reliability behavior when Ω changes from 0.9 to 0.95. (b) Reliability behavior when τ changes from 1 s to 60 ms. (c) Delay behavior when Ω changes from 0.9 to 0.95. (d) Delay behavior when τ changes from 1 s to 60 ms. (e) Active time behavior when Ω changes from 0.9 to 0.95. (f) Active time behavior when τ changes from 1 s to 60 ms.

taking into account the trade-off between energy consumption and application requirements for control applications.

We developed an analytical expression of the total energy consumption of the network, as well as reliability and delay for the packet delivery. These relations allowed us to pose a mixed real-integer constrained optimization problem to optimize the number of hops in the multihop routing, the wake-up rates of the nodes, and the transmit radio power as a function of the routing, MAC, physical layer, traffic, and hardware platform. An algorithm for the dynamic and continuous adaptation of the network operations to the traffic and channel conditions, and application requirements, was proposed.

We provided a complete testbed implementation of the protocol, building a wireless sensor network with TinyOS and Tmote sensors. An experimental campaign was conducted to test the validity of Breath in an indoor environment with both AWGN and Rayleigh fading channels. Experimental results showed that the protocol achieves the reliability and delay requirements, while minimizing the energy consumption. It outperformed a standard IEEE 802.15.4 implementation in terms of both energy efficiency and reliability. In addition, Breath showed good load balancing performance, and is scalable with the number of nodes. Given its good performance, Breath is a

good candidate for many control and industrial applications, since these applications ask for both reliability and delay requirements in the packet delivery. A practical application of the protocol was illustrated in [11].

We are currently investigating the extension of the design methodology to consider mesh networks such as coexisting ad hoc and wireless sensor networks.

ACKNOWLEDGMENTS

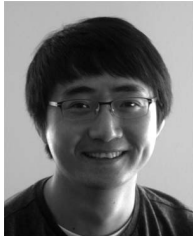
A preliminary version of this work was presented at IEEE SECON 2008 [1]. Alvisio Bonivento was with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, when contributing to this work. Pangun Park, Carlo Fischione, and Karl Henrik Johansson were supported by the Swedish Research Council, Strategic Research Foundation, and Swedish Governmental Agency for Innovation Systems.

REFERENCES

- [1] P. Park, C. Fischione, A. Bonivento, K.H. Johansson, and A. Sangiovanni-Vincentelli, "Breath: A Self-Adapting Protocol for Wireless Sensor Networks in Control and Automation," *Proc. Fifth Ann. IEEE Comm. Soc. Conf. Sensor, Mesh and Ad Hoc Comm. and Networks (SECON '08)*, 2008.

- [2] A. Willig, K. Matheus, and A. Wolisz, "Wireless Technology in Industrial Networks," *Proc. IEEE*, vol. 93, no. 6, pp. 1130-1151, June 2005.
- [3] A. Willig, "Recent and Emerging Topics in Wireless Industrial Communication," *IEEE Trans. Industrial Informatics*, vol. 4, no. 2, pp. 102-124, May 2008.
- [4] V.C. Gungor and G.P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *IEEE Trans. Industrial Electronics*, vol. 56, no. 10, pp. 4258-4265, Oct. 2009.
- [5] *IEEE 802.15.4 Std: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE, <http://www.ieee802.org/15/pub/TG4.html>, 2006.
- [6] A. Bonivento, L. Carloni, and A. Sangiovanni-Vincentelli, "Platform-Based Design for Wireless Sensor Networks," *Mobile Networks and Applications*, vol. 11, no. 4, pp. 469-485, 2006.
- [7] A. Sangiovanni-Vincentelli, "Quo Vadis, SLD? Reasoning about the Trends and Challenges of System Level Design," *Proc. IEEE*, vol. 95, no. 3, pp. 467-506, Mar. 2007.
- [8] J.P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Recent Results in Networked Control Systems," *Proc. IEEE*, vol. 95, no. 1, pp. 138-162, Jan. 2007.
- [9] W. Zhang, M.S. Branicky, and S.M. Phillips, "Stability of Networked Control Systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84-99, Feb. 2001.
- [10] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poola, and S. Sastry, "Foundations of Control and Estimation over Lossy Networks," *Proc. IEEE*, vol. 95, no. 1, pp. 163-187, Jan. 2007.
- [11] E. Witrant, P. Park, M. Johansson, C. Fischione, and K.H. Johansson, "Predictive Control over Wireless Multi-Hop Networks," *Proc. IEEE Int'l Conf. Control Applications (CCA '07)*, 2007.
- [12] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," *Proc. ACM MobiCom*, 2000.
- [13] Tmote Sky Data Sheet, Moteiv, <http://www.moteiv.com/products/docs/tmote-sky-datashheet.pdf>, 2006.
- [14] P. Buonadonna, D. Gay, J. Hellerstein, W. Hong, and S. Madden, "TASK: Sensor Network in a Box," technical report, Intel Research Lab Report, 2007.
- [15] R. Steigman and J. Endresen, "Introduction to WISA and WPS, WISA—Wireless Interface for Sensors and Actuators and WPS—Wireless Proximity Switches," white paper, <http://www.eit.uni-kl.de/litz/WISA.pdf>, 2004.
- [16] The ZigBee Alliance, <http://www.zigbee.org>, 2011.
- [17] WirelessHart Data Sheet, <http://www.hartcomm2.org/hartprotocol/wirelesshart/wirelesshartmain.html>, 2007.
- [18] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '06)*, 2006.
- [19] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '04)*, 2004.
- [20] Y. Sun, O. Gurewitz, and D. Johnson, "RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks," *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '08)*, 2008.
- [21] W. Ye, F. Silva, and J. Heidemann, "Ultra-Low Duty Cycle MAC with Scheduled Channel Polling," *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '06)*, 2006.
- [22] G. Halkes and K. Langendoen, "Crankshaft: An Energy-Efficient MAC Protocol for Dense Wireless Sensor Networks," *Proc. European Conf. Wireless Sensor Networks (EWSN '07)*, 2007.
- [23] G. Ahn, S.G. Hong, E. Miluzzo, A.T. Campbell, and F. Cuomo, "Funneling-MAC: A Localized, Sink-Oriented MAC for Boosting Fidelity in Sensor Networks," *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '06)*, 2006.
- [24] Y. Xu, J. Heidemann, and D. Estrin, "Geography-Informed Energy Conservation for Ad Hoc Routing," *Proc. ACM MobiCom*, 2001.
- [25] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "SPAN: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Proc. ACM MobiCom*, 2001.
- [26] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and Yield in a Volcano Monitoring Sensor Network," *Proc. USENIX Symp. Operating Systems Design and Implementation (OSDI '06)*, 2006.
- [27] N. Burri, P. von Rickenbach, and R. Wattenhofer, "Dozer: Ultra-Low Power Data Gathering in Sensor Networks," *Proc. Sixth IEEE/ACM Int'l Symp. Information Processing in Sensor Networks (IPSN '07)*, 2007.
- [28] R. Musaloiu-E, C.M. Liang, and A. Terzis, "Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks," *Proc. Seventh IEEE/ACM Int'l Symp. Information Processing in Sensor Networks (IPSN '08)*, 2008.
- [29] M. Zorzi and R.R. Rao, "Energy and Latency Performance of Geographic Random Forwarding for Ad Hoc and Sensor Networks," *Proc. IEEE Wireless Comm. and Networking (WCNC '03)*, 2003.
- [30] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," *Proc. ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys '09)*, 2009.
- [31] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without Routes: The Backpressure Collection Protocol," *Proc. Ninth IEEE/ACM Int'l Conf. Information Processing in Sensor Networks (IPSN '10)*, 2010.
- [32] E. Felemban, C.G. Lee, and E. Eylem, "MMSPEED: Multipath Multi-Speed Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 5, no. 6, pp. 738-754, June 2006.
- [33] A. Bonivento, C. Fischione, L. Necchi, F. Pianegiani, and A. Sangiovanni-Vincentelli, "System Level Design for Clustered Wireless Sensor Networks," *IEEE Trans. Industrial Informatics*, vol. 3, no. 3, pp. 202-214, Aug. 2007.
- [34] J. Misić, S. Shafi, and V. Misić, "Cross-Layer Activity Management in an 802.15.4 Sensor Network," *IEEE Comm. Magazine*, vol. 44, no. 1, pp. 131-136, Jan. 2006.
- [35] L. van Hoesel, T. Nieberg, J. Wu, and P. Havinga, "Prolonging the Lifetime of Wireless Sensor Networks by Cross-Layer Interaction," *IEEE Trans. Wireless Comm.*, vol. 11, no. 6, pp. 78-86, Dec. 2004.
- [36] J. Van Greunen, D. Petrovic, A. Bonivento, J. Rabaey, K. Ramchandran, and A. Sangiovanni-Vincentelli, "Adaptive Sleep Discipline for Energy Conservation and Robustness in Dense Sensor Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '04)*, 2004.
- [37] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, "Network Correlated Data Gathering with Explicit Communication: NP-Completeness and Algorithms," *IEEE/ACM Trans. Networking*, vol. 14, no. 1, pp. 41-54, Feb. 2006.
- [38] P. Chen and S. Sastry, "Latency and Connectivity Analysis Tools for Wireless Mesh Networks," *Proc. First IEEE/ACM Int'l Conf. Robot Comm. and Coordination (ROBOCOMM '07)*, 2007.
- [39] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Trans. Wireless Comm.*, vol. 1, no. 4, pp. 660-670, Oct. 2002.
- [40] Y. Ma and J.H. Aylor, "System Lifetime Optimization for Heterogeneous Sensor Networks with a Hub-Spoke Topology," *IEEE Trans. Mobile Computing*, vol. 3, no. 3, pp. 286-294, July-Aug. 2004.
- [41] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Trans. Mobile Computing*, vol. 3, no. 4, pp. 366-379, Oct.-Dec. 2004.
- [42] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Trans. Networking*, vol. 12, no. 3, pp. 493-506, June 2004.
- [43] CC2420 Data Sheet, Chipcon, <http://www.chipcon.com/files/CC2420-Data-Sheet-1-3.pdf>, 2005.
- [44] CC2431 Data Sheet, Texas Instruments, <http://focus.tij.co.jp/jp/lit/ds/symlink/cc2431.pdf>, 2011.
- [45] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Selected Areas in Comm.*, vol. 18, no. 3, pp. 535-547, Mar. 2000.
- [46] S. Pollin, M. Ergen, S.C. Ergen, B. Bougard, L.V.D. Perre, F. Cathoor, I. Moerman, A. Bahai, and P. Varaiya, "Performance Analysis of Slotted Carrier Sense IEEE 802.15.4 Medium Access Layer," *Proc. IEEE Global Telecomm. Conf. (GLOBECOM '06)*, 2006.
- [47] Y. Xiao and J. Rosdahl, "Throughput and Delay Limits of IEEE 802.11," *IEEE Comm. Letters*, vol. 6, no. 8, pp. 355-357, Aug. 2002.
- [48] H. Zhai, Y. Kwon, and Y. Fang, "Performance Analysis of IEEE 802.11 MAC Protocols in Wireless LANs: Research Articles," *Wireless Comm. and Mobile Computing*, vol. 4, no. 8, pp. 917-931, 2004.

- [49] J. Misić, S. Shafi, and V.B. Misić, "Performance of a Beacon Enabled IEEE 802.15.4 Cluster with Downlink and Uplink Traffic," *IEEE Trans. Parallel and Distributed Systems*, vol. 17, no. 4, pp. 361-376, Apr. 2006.
- [50] P. Park, P.D. Marco, C. Fischione, and K.H. Johansson, "Accurate Delay Analysis of Slotted IEEE 802.15.4 for Control Applications," technical report, KTH, http://www.ee.kth.se/~pgpark/papers/delay_wpan.pdf, 2010.
- [51] A. Lymberopoulos and D. Savvides, "XYZ: A Motion-Enabled, Power Aware Sensor Node Platform for Distributed Sensor Network Applications," *Proc. Fourth IEEE/ACM Int'l Conf. Information Processing in Sensor Networks (IPSN)*, 2005.



Pangun Park received the BS degree in electrical engineering from Aju University, Suwon, Korea, in 2005 and the MS degree in electrical engineering from the Royal Institute of Technology, Stockholm, Sweden, in 2007. Since 2007, he has been working toward the PhD degree in the School of Electrical Engineering at the Royal Institute of Technology (KTH), Sweden. He received the Best Paper award at the IEEE International Conference on Mobile Ad-Hoc and Sensor Systems 2009. His research interests include ad hoc and sensor networks, and networked control systems.



Carlo Fischione received the DrEng degree in electronic engineering (Laurea Summa Cum Laude) and the PhD degree in electrical and information engineering from the University of L'Aquila, Italy, in April 2001 and May 2005, respectively. He is currently an assistant professor in the Department of Electrical Engineering, ACCESS Linnaeus Centre, Royal Institute of Technology (KTH), Stockholm, Sweden. His history includes holding research positions at the University of California at Berkeley, California (2004-2005 and 2007-2008) and Royal Institute of Technology, Stockholm, Sweden (2005-2007). He is supervising a number of doctorate and graduate students at the Royal Institute of Technology (KTH), Sweden, and the University of L'Aquila, Italy. He received the Best Paper award from the *IEEE Transactions on Industrial Informatics* in 2007, the best paper awards at the IEEE International Conference on Mobile Ad-Hoc and Sensor System 2005 and 2009 (IEEE MASS 2005 and IEEE MASS 2009), and is a recipient of the "Ferdinando Filaurio" award from the University of L'Aquila, Italy, and the "Alta Formazione" Award from the Abruzzo Region Government, Italy. He has chaired or served as a technical member of program committees of several international conferences and is serving as referee for technical journals. Meanwhile, he has also offered his advice as a consultant to numerous technology companies such as the Berkeley Wireless Sensor Network Lab, Ericsson AB, Synopsys, and the United Technology Research Center. His research interests include optimization, wireless sensor networks, and system-level design of wireless networks. He is a member of the IEEE.



Alvise Bonivento received the Laurea degree in telecommunication engineering from the University of Padova, Italy, in 2002 and the MS and PhD degrees in electrical engineering and computer science from the University of California, Berkeley, in 2004 and 2007, respectively. His research interest is in communication protocols for embedded systems, with emphasis on system-level design and protocol synthesis for wireless embedded networks and wireless industrial networks. He is a coauthor of several technical papers in major communication and embedded systems conferences and he received the best paper award from the *IEEE Transactions on Industrial Informatics* in 2007. He was a management consultant at McKinsey & Co. from 2007 and 2009 and joined a venture capital firm in 2009 where he currently is an investment manager.



Karl Henrik Johansson received the MSc and PhD degrees in electrical engineering from Lund University, Sweden, in 1992 and 1997, respectively. He is the director of the ACCESS Linnaeus Centre and professor at the School of Electrical Engineering, Royal Institute of Technology, Sweden. He is a Wallenberg scholar and holds a senior researcher position with the Swedish Research Council. He has held visiting positions at the University of California at Berkeley, California (1998-2000) and the California Institute of Technology (2006-2007). His research interests are in networked control systems, hybrid and embedded control, and control applications in automotive, automation, and communication systems. He has been the chair of the International Federation of Automatic Control (IFAC) Technical Committee on Networked Systems since 2008. He has served on the executive committees of several European research projects in the area of networked embedded systems. He is on the editorial boards of the *IEEE Transactions on Automatic Control* and *IET Control Theory & Applications*, and previously of the IFAC journal *Automatica*. He was awarded an Individual Grant for the Advancement of Research Leaders from the Swedish Foundation for Strategic Research in 2005. He received the triennial Young Author prize from IFAC in 1996 and the Peccei award from the International Institute of System Analysis, Austria, in 1993. He received Young Researcher awards from Scania in 1996 and from Ericsson in 1998 and 1999.



Alberto L. Sangiovanni-Vincentelli received the degree in electrical engineering and computer science ("Dottore in Ingegneria") summa cum laude from the Politecnico di Milano, Italy, in 1971. He holds the Edgar L. and Harold H. Buttner chair of electrical engineering and computer sciences at the University of California at Berkeley. In 1987, he was a visiting professor at the Massachusetts Institute of Technology. He was a cofounder of Cadence and Synopsys, the two leading companies in the area of Electronic Design Automation. He is a member of the board of directors of Cadence and the chair of its Technology Committee, UPEK, Sonics, and Accent. He was a member of the HP Strategic Technology Advisory Board and is a member of the Science and Technology Advisory Board of General Motors. He is a member of the High-Level Group, the steering committee, the governing board, and of the Public Authorities Board of the EU Artemis Joint Technology Undertaking. He is a member of the scientific council of the Italian National Science Foundation (CNR) and a member of the executive committee of the Italian Institute of Technology. He received the Kaufman award of the Electronic Design Automation Council for "pioneering contributions to EDA" and the IEEE/RSE Wolfson James Clerk Maxwell Medal "for groundbreaking contributions that have had an exceptional impact on the development of electronics and electrical engineering or related fields." He is an author of more than 800 papers, 15 books, and three patents. He is a fellow of the IEEE and a member of the National Academy of Engineering.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.