Breath: a Self-Adapting Protocol for Wireless Sensor Networks in Control and Automation

P. G. Park, C. Fischione, A. Bonivento, K. H. Johansson, A. Sangiovanni-Vincentelli

***Abstract*—The novel cross-layer protocol Breath for wireless sensor networks is designed, implemented, and experimentally evaluated. The Breath protocol is based on randomized routing, MAC and duty-cycling, which allow it to minimize the energy consumption of the network while ensuring a desired packet delivery end-to-end reliability and delay. The system model includes a set of source nodes that transmit packets via multi-hop communication to the destination. A constrained optimization problem, for which the objective function is the network energy consumption and the constraints are the packet latency and reliability, is posed and solved. It is shown that the communication layers can be jointly optimized for energy efficiency. The optimal working point of the network is achieved with a simple algorithm, which adapts to traffic variations with negligible overhead. The protocol was implemented on a test-bed with off-the-shelf wireless sensor nodes. It is compared with a standard IEEE 802.15.4 solution. Experimental results show that Breath meets the latency and reliability requirements, and that it exhibits a good distribution of the working load, thus ensuring a long lifetime of the network.**

Keywords: Wireless Sensor Network, Power control, MAC, duty cycle, optimization.

I. Introduction

The deployment of Wireless Sensor Networks (WSNs) for monitoring and control applications heavily depends on the possibility to provide an efficient communication infrastructure. The design of such networked systems has to take into account a large number of factors that ensure the correct implementation: the constraints imposed by the applications running on the network (e.g., end-to-end latency, error probability), the limited energy resources of WSNs, and the available implementation hardware platform.

The network design task can be formulated as an optimization problem. However, as it was noted in [1], a complex interdependence of the decision variables with the network properties often lead to difficult problems even in simple network topologies. As a result, it is not possible to provide a unique solution to the problem of WSNs design, but rather a set of components not too general (to avoid inefficiencies) and not too ad hoc (to allow for reusability).

In this paper, we present Breath, an efficient protocol solution for WSNs for common control and automation application: a source of information has to send packets to a destination using a multi-hop WSN under end-to-end latency and error probability constraints. The solution is based on a randomized routing, a randomized Medium Access Control (MAC) and a randomized sleeping discipline that are jointly optimized for energy consumption. We introduce an adaptation algorithm that allows the network to adapt to traffic variations and reach the optimal working point without communication or state overhead. Randomized routing allows us to reduce overhead due to node coordination, state maintenance and increase robustness on neighboring nodes failures. Random access to the wireless channel avoids packet collisions, while the random sleep discipline permits the nodes to minimize their energy consumption. Since the protocol proposed in this paper adapts to the network variations enlarging or shrinking next-hop distance and sleep time of the nodes, we named it Breath.

There have been many contributions to the problem of protocol design for WSNs, both in academia (e.g., [2], [3]) and industry (e.g., [4], [5]). New protocols have been built around standardized low-power protocols such as IEEE 802.15.4 [6] and Zigbee [7]. To the best of our knowledge, no protocol in the literature presents a comprehensive solution that includes all the relevant characteristics of the physical layer, MAC and routing, which guarantees latency and reliability constraints over multi-hop communication, and optimizes for energy consumption. A first step toward an integrated protocol stack is visible in [8]. In that paper, a randomized routing protocol, a randomized sleeping disciple and a joint optimization are presented. The routing algorithm is called "Region-based Opportunistic Routing" and it is an extension of the geographic routing proposed in [9], where the idea of routing through a random sequence of hops is introduced. Similarly to these approaches, we present in this paper a solution which routes packets through a random sequence of nodes.

An important means of ensuring energy savings and longer network lifetime is enforcing a sleeping discipline, i.e., an algorithm that turns off a node whenever its presence is not required for the correct operation of the network [10], [11]. According to such a discipline, each node goes to sleep for an

P. G. Park and K. H. Johansson are with the ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden. C. Fischione and A. Sangiovanni-Vincentelli are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley. A. Bonivento was with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, when part of this work was done. E-mails: {pgpark,kallej}@ee.kth.se, {fischion,alberto}@eecs.berkeley.edu.

Work done in the framework of the HYCON Network of Excellence, contract number FP6-IST-511368. P. G. Park and K. H. Johansson were also supported by the EU integrated project SOCRADES, the Swedish Research Council, and the Swedish Strategic Research Foundation. C. Fischione and A. Sangiovanni-Vincentelli wish to acknowledge the support of the NSF ITR CHESS and the GSRC.

amount of time that is a random variable dependent on traffic and network conditions. In this paper, we present a duty cycle solution that can be considered as an extension of [12], [13].

In [8] a first level of cross-layer interaction is exploited. In particular, it is shown how the opportunistic routing and the randomized sleeping discipline can be jointly optimized for energy saving while satisfying requirements on end-to-end delay. However, important aspects as the impact of packet collisions have not been considered.

In [13], [14], a relevant design methodology has been presented. Here, we extend and test the design methodologies of that papers by including collision avoidance mechanism, and detailed behavior of the physical layer. Especially, the original contribution is as follows:

1) A comprehensive energy minimization is proposed under reliability and latency constraints. It takes into account the overall energy spent to transmit and receive packets, including an accurate radio power minimization.
2) An algorithm that introduces little communication overhead and allows for rapid deployment and self adaptation of the network to optimal working conditions is presented.
3) The protocol solution is implemented over a complete test bed using Tmote Sky sensor nodes [4].

The rest of the paper is organized as follows: In Section II our protocol is presented. In Section III an optimization problem is introduced to describe the protocol. In Section IV, we show how the protocol can be optimized for power consumption and in Section V we present an algorithm to obtain the optimal working point. An experimental implementation of the protocol is presented in Section VI. Finally, in Section VII concluding remarks and future perspectives are given.

II. The Breath Protocol

In this section we introduce the system model and the protocol proposed in this paper.

We consider a scenario where there is a cluster of source nodes generating data packets associated to a sensed phenomenon. These packets are generated with a rate of λ pckt/s. Between source nodes and the destination, we assume that nodes are uniformly deployed to relay these packets. These intermediate nodes do not generate their own packets, but just relay those coming from the sources using the randomized routing. We adopt a randomized routing, because it is simple to implement, robust and fault tolerant. We assume that each node knows its location. This information can be either hard-coded in the node when they are deployed, or it can be obtained running a positioning algorithm on the network right after deployment. The intermediate nodes are grouped into a number of clusters or forwarding regions. Clustered network topology is supported in networks that require energy efficiency, since transmitting data through intermediate nodes may consume more than routing directly to the destination [15]. Data packets can be transmitted only from a cluster to next cluster closer to the sink. In Fig. 1, the system scenario is depicted. The network is abstracted by $h - 1$ blocks or clusters, as it was proposed in [8], [12]. These blocks represent the forwarding

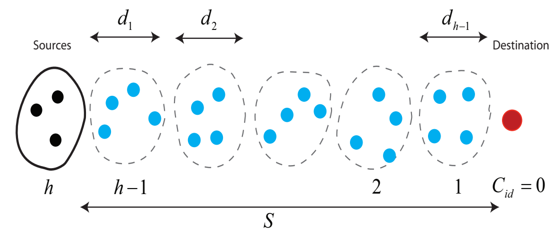

Fig. 1. Network nodes are organized into sources, $h - 1$ blocks, and destination. $C_{id} = 0, \ldots, h$ denotes the group ID of each block.

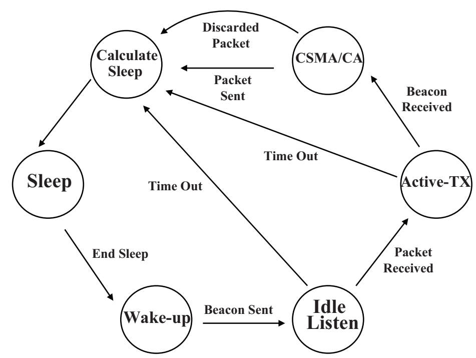

Fig. 2. State machine description of the Breath protocol

regions to which packets can be forwarded. The number of these clusters and wake-up rate change continuously according to the traffic and wireless channel conditions (as we see later). Looking at the network, the continuous enlarging and shrinking of the cluster size makes our protocol to behave like a breath, thereby, we denoted our solution the Breath protocol. We assume that nodes have tunable transmit power. Packets transmitted by the nodes are of two kinds: data packets, which contain information related to the sensed phenomenon, and beacon packets, which carry information related to the control parameters of the protocol. We assume that data packets and beacon packets are transmitted at two fixed disjoint frequencies, so to reduce packet collisions. These assumptions are perfectly compatible with off-the-shelf hardware platforms, as the Tmote Sky [4].

The Breath protocol is a cross-layer solution. The MAC, routing, and duty-cycle algorithms are designed and optimized all together. According to the protocol, a node sends a data packet to a node randomly selected in a forwarding region, which is located in the direction toward the sink node. Nodes in the forwarding region send beacon messages to say that they are available to receive data packets. The MAC is probabilistic and does not implement any acknowledgement or retransmission scheme. 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 conditions. Hence, the duty-cycling algorithm is randomized. The cumulative wake-up rate, i.e., the sum of the wake-up rates that a node sees from all nodes of the next cluster is denoted μ_c, which is the same for each cluster, as we see later.

The detailed behavior of a node is explained in the state machine of Fig. 2, which we describe in the following:

- **Sleep State**: the node k turns off its radio and starts a grenade timer whose duration is an exponentially distributed random variable with average μ_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 the data channel, and it goes to the Idle Listen State.
- **Idle Listen State**: the node starts a grenade timer of a fixed duration that must be long enough to receive completely 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 (Carrier Sense Multiple Access/Collision Avoidance) 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. As soon as a node receives a data packet, it tries to send it to a node in the forwarding region. In CSMA/CA, the node checks whether the channel is clear, i.e., if no other node is transmitting at the time. If the node recognizes a clean channel, then the data packet is sent. If the channel is not clear, the node waits for an exponentially distributed random time, and then checks the channel again. This procedure repeats itself until a maximum number of tries have been done. If the channel is never clear, 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 and goes directly to the Calculate Sleep State.
- **Calculate Sleep State**: the node calculates the parameter μ_k for the next sleeping time and generates an exponential distributed random variable having average $1/\mu_k$. After this the node goes back to the Sleep State. The sleeping parameter μ_k is computed such that the cumulative sleep time of the cluster μ_c can be achieved, according to the adaptation algorithm given in Section V.

According to the protocol given above, the packet delivery depends on the cumulative wake-up time and on the number of forwarding regions. In the next sections, we show how to tune online these parameters to satisfy delay and reliability constraints imposed by the application and optimize them for energy consumption.

III. Breath Optimization Problem

The protocol is tuned by a constrained optimization problem. The objective function is the total energy consumption for transmitting and receiving packets from the source to the destination. The constraints are given by an end-to-end delay probability, and a packet reception probability. The decision variables of the optimization problem are the cumulative wake-up rate μ_c of each cluster and the number of blocks, $h-1$. The optimization problem is

$$\begin{aligned} \mathcal{P}: \quad & \min_{h,\mu_c} && E_{\text{tot}}(h,\mu_c) \\ & \text{s.t.} && \Pr[D(h,\mu_c) \le \tau] \ge \Delta\,, \\ & && \psi(h,\mu_c) \ge \Omega\,. \end{aligned} \tag{1}$$

In Problem $\mathcal{P}$, the objective function is the total energy consumption of the network, denoted as $E_{\text{tot}}(h,\mu_c)$. $D(h,\mu_c)$ is the distribution of the end-to-end delay to transmit a packet from the source to the sink, τ is the desired maximum end-to-end delay, and Δ is the minimum probability with which such a maximum delay should be achieved. The constraint $\psi(h,\mu_c)$ is the probability of successful packet delivery from the source to the sink, and Ω is the minimum desired probability. We remark here such that Δ and Ω are the requirements imposed by the application, and h,μ_c are the protocol parameters that have to be adapted to traffic, channel conditions, and application requirements.

In the following sections, a characterization of Problem $\mathcal{P}$ is given, along with a strategy to achieve the optimal solution, namely the values of h,μ_c that minimize the cost function. As we will see later, some approximations must be done on the constraints, whereas we use an upper bound of the cost function. We intend as optimal solution the solution that solves such an approximated optimization problem. The complex interdependence of the protocol parameter prevents to model with exact accuracy the constraints and cost function, as we discuss next.

A. Latency Constraint

The end-to-end delay between source to destination is given by the sum of the delays at each hop. There are three sources of delay per hop:

- **Time to wait before the first wake-up of a node in the next cluster**: This time is an exponentially distributed random variable α_i whose intensity μ_c is the sum of the wake-up intensities of the nodes in the next cluster i.
- **Time to wait clean channel**: Before sending a packet, a node senses if the channel is busy a CSMA/CA mechanism. If the channel is busy, then the node waits an exponentially distributed random back-off time ε_i with intensity μ_ε. This operation is repeated at most M_b times, after which a packet is discarded. The average number of back-offs is denoted through μ_b. It can be computed as follows: Let P_{hb} be the busy channel probability, then

$$\mu_b = \sum_{j=1}^{M_b} j(1-P_{hb})P_{hb}^{j-1}\,,$$

where, according to the analysis carried out in [16],

$$P_{hb} = \left(\frac{\lambda}{\mu_c+\lambda}\right)^2. \tag{2}$$

where there are h hops to send a packet from source to destination. Finally, the end-to-end delay is a random variable:

$$D(h, \mu_c) = \sum_{i=1}^{h} (\alpha_i + \mu_b \, \varepsilon_i) \, . \quad (3)$$

B. Reliability Constraint

Since we implemented a CSMA/CA scheme, a data packet can be lost at each hop because of a bad wireless channel or collisions. The first such an event is modeled with a Bernoulli model, where the probability of having a good channel during a single transmission is denoted with p. Recalling that M_b is the maximum number of tries to perceive a clean channel, the successful probability to transmit a packet within M_b tries can be well approximated by $1 - [\lambda/(\mu_c + \lambda)]^2 \, \rho_c^{M_b}$ where ρ_c is an upper bound of collision probability in CSMA/CA State. Therefore, the reliability constraint can be expressed by

$$\psi(h, \mu_c) = \prod_{i=1}^{h} p \left[1 - \left(\frac{\lambda}{\mu_c + \lambda} \right)^2 \rho_c^{M_b} \right] \geq \Omega \, . \quad (4)$$

C. Cost Function

The total energy consumption is given by the energy related to the communication of data packets, and the energy to wake-up and beckoning, namely $E_{\text{tot}}(h, \mu_c) = E_{\text{pck}} + E_{wu}$. In the following, we characterize these energies.

Consider the energy spent for transmission and reception of a data packet for a node in the i th cluster. Let us denote the energy consumption for radio transmission with $Q_m(d_i)$, where d_i is the transmission distance to which a data packet has to be transmitted from the node. Such an energy is a function of the radio power used to transmit the packet. The following expression holds:

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

where V is the voltage consumption at the node, t_m is the transmission time of a data packet, and $I(P_t(d_i))$ is the current consumption of the electronic circuit needed to transmit packets of radio power $P_t(d_i)$. The relation between the current consumption and radio power obviously depends on the hardware platform. Using Telos sensors, the following relation holds [17]: $I(P_t(d_i)) \approx -19P_t(d_i)^4 + 53P_t(d_i)^3 - 53P_t(d_i)^2 + 29P_t(d_i) + 8.7$. In Section V-D, the term $P_t(d_i)$ is characterized. Before sending a data packet, the node waits for a beacon coming from the forwarding region, and perceives a clean channel in the CSMA/CA. The expected energy consumption corresponding to this procedure is

$$\mu_b \left(\frac{W_R}{\mu_c} + \frac{W_T}{\mu_\varepsilon} \right) ,$$

where W_R is power consumption at RX mode and W_T is power consumption at TX mode.

For the reception of a data packet, there is a fixed cost R due to the RF circuit at the receiver node. Assuming h hops, and recalling that sources emits λ pckt/s, the total energy consumption for transmission and reception during a time of T s is

$$E_{\text{pck}} = T\lambda \sum_{i=1}^{h} \left[Q_m(d_i) + \mu_b \left(\frac{W_R}{\mu_c} + \frac{W_T}{\mu_\varepsilon} \right) + R \right] . \quad (5)$$

We would like to remark here that the energy consumption in (5) has been derived with the implicit assumption that all packets reach the sink. Obviously, some packet may be lost before reaching the sink, therefore Equation (5) gives a upper bound on the energy consumption.

Consider now the energy for wake-up, listening and beckoning. Each node randomly cycles between an awake state and a sleep state. Each time a node wakes up, it spends a given energy, which is given by the power needed to wake-up W_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}. These two energies give

$$E_{ac} = W_w T_w + W_R (T_{ac} - T_w) \, .$$

After a node wakes up, if transmits a beacon to the next block. Defining the wireless channel loss probability as $1 - p$, nodes have to wake-up on average $1/(1-p)$ 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 block μ_c, the total cost in a time T for wake-ups and beaconing becomes

$$E_{wu} = \frac{T}{1-p} \sum_{i=1}^{h} \left[W_w T_w + W_R (T_{ac} - T_w) + Q_b(d_i) \right] \, , \quad (6)$$

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

IV. Solving Breath Optimization Problem

In this section we propose an approach to the solution of the Problem $\mathcal{P}$ which optimize our protocol. We derive a characterization of the constraints and the cost function. The section is concluded with the solution of the problem.

A. Delay Constraint

Consider the end-to-end delay constraint in Equation (3). Since α_i and ε_i, $i = 1, \ldots, h$ are exponentially distributed, the central limit theorem can be applied. Hence, the delay D can be approximated with a Gaussian random variable. Consequently, the delay is approximated with $D \in N(\mu_D, \sigma_D^2)$, where

$$\mu_D = \mu_b \left[\frac{h}{\mu_c} + \frac{h}{\mu_\varepsilon} \right] ,$$
$$\sigma_D^2 = \mu_b^2 \left[\frac{h}{\mu_c^2} + \frac{h}{\mu_\varepsilon^2} \right] ,$$

Previous equation can be used to express the probability of the end-to-end delay constraint in Problem $\mathcal{P}$:

$$\Pr[D \leq \tau] \approx 1 - Q \left(\frac{\tau - \mu_D}{\sigma_D} \right) \geq \Delta \, , \quad (7)$$

where $Q(x) = 1/\sqrt{2\pi}\int_x^\infty e^{-t^2/2}dt$ is the complementary standard Gaussian distribution. Solving (7) for μ_c, we obtain

$$\begin{aligned} 0 < \mu_c(h) \leq D_{c-}(h) \\ \mu_c(h) \geq D_{c+}(h) \end{aligned} \quad (8)$$

where $D_{c-}(h)$ and $D_{c+}(h)$ are given in (9) and (10). We evidenced that μ_c is a function of h due to these constraints. In that equations, $\mu_{b,\max}$ is an upper bound of the expected number of back-off tries:

$$\mu_b \leq \frac{1}{1 - \max P_{hb}} \leq \frac{\Omega + 1}{\Omega} \triangleq \mu_{b,\max}\,,$$

where we used the fact that the wake-up rate is bounded as $\mu_c \geq \Omega\lambda$.

The fact that the roots in the expression $D_{c-}(h)$ and $D_{c+}(h)$ must be positive gives the constraints $\Gamma_- \leq h \leq \Gamma_+$, where

$$\Gamma_- = \frac{\tau\mu_\varepsilon}{\mu_{b,\max}}$$
$$\Gamma_+ = \frac{\tau\mu_\varepsilon}{\mu_{b,\max}} + 2\Big(1 - \sqrt{\frac{\tau\mu_\varepsilon}{\mu_{b,\max}} + 1}\Big)\,.$$

These constraints are useful for the search of the optimal value of h, as we will see later.

B. Reliability

From the constraint on the reliability (4), it is possible to express a bound on the cumulative wake-up rate, which after some simple algebras becomes:

$$\mu_c(h) \geq \lambda\left(\sqrt{\frac{p\rho_c^{M_b}}{p - \Omega^{1/h}}} - 1\right)\,. \quad (12)$$

Note that from the fact that the squared root of previous inequality must be positive, the constraint $h \leq \frac{\ln(\Omega)}{\ln(p)}$ is obtained.

C. Cost Function

Optimizing the cost function $E_{\text{tot}}(h, \mu_c) = E_{\text{pck}} + E_{wu}$ has some difficulties. The radio power used to transmit packets depends on the distance to which the packet must be sent. Hence, the cost function can be slightly upper bounded by considering the worst distance to which a packet must be sent, which is $S/(h-1)$. From (5) and (6) we obtain

$$\begin{aligned} E_{\text{pck}} = \; & T\lambda\Big[Q_m(\tfrac{S}{h-1}) + Q_m(\tfrac{S}{h-1})(h-1)u(h-1) \\ & + h\tfrac{(\mu_c(h)+\lambda)^2}{(\mu_c(h)+\lambda)^2 - \lambda^2\rho_b}\big(\tfrac{W_R}{\mu_c(h)} + \tfrac{W_T}{\mu_\varepsilon}\big) + hR\Big]\,, \end{aligned} \quad (13)$$

$$\begin{aligned} E_{wu} = \; & \tfrac{T\mu_c(h)}{p}\Big[hW_wT_w + hW_R(T_{ac} - T_w) + 2Q_b(\tfrac{S}{h-1}) \\ & + Q_b(\tfrac{2S}{h-1})(h-2)u(h-2)\Big]\,, \end{aligned} \quad (14)$$

where $u(x) = 1$ if $x \geq 0$, and $u(x) = 0$ otherwise. It is possible to show that the cost function is convex both in h and $\mu_c(h)$ providing that one assumes h as a real number [16]. We find the optimal solution in two steps. For each value of h, the cost function is minimized for μ_c. The pair h and $\mu_c(h)$ which minimize the cost function is the optimal solution.

Specifically, from Equation (8) and (12), the conditions of the optimal wake-up rate are derived from two constraints with the given value of hops. However, it is possible to show that the minimum wake-up rate from the constraints does not guarantee the optimal working point in terms of the total energy consumption. The derivative of the total energy consumption $E_{\text{tot}}(h, \mu_c)$ is used to find the optimal wake-up rate that minimizes the objective function. By getting the derivative of the objective function with respect to the wake-up rate, the optimal wake-up rate is given by Equation (11). We assume that $(\mu_c(h) + \lambda)^2 \gg \lambda^2$ to achieve the derivative of E_{pck}. If (11) is within the bounds given by Equation (8) and (12), then the derivative gives the optimal rate. Otherwise, recalling the convexity of the cost function, the optimal rate is given by one of the constraints.

V. Adaptation Mechanisms

In the previous sections, we showed how to determine the optimal forwarding region and cumulative wake-up rate as the solution of an optimization problem. Here, we present in detail some adaptation algorithms that the destination must run to determine correctly the forwarding region size and wake-up rate as the traffic rate and channel conditions changes. These algorithms allow us to adapt the protocol behavior to the channel condition without high message overhead.

A. Computation of the Protocol Parameters

We assume that all the physical layer abstraction values can be estimated at the destination. Consequently, the destination node solves the optimization problem as described in section IV knowing the traffic rate λ and the average channel condition p. The return value of the algorithm are the protocol parameters, namely the optimal number of hops h and cumulative wake-up rate μ_c, can be piggybacked on beacons toward the intermediate nodes closer to the destination. Then, the protocol parameters are forwarded when the nodes wake-up and send beacons to the next cluster toward the source. During the initial state, nodes set $h = 2$ before receiving a beacon.

B. Estimation of the Node's Wake-up Rate

Assume there are N nodes in a block. We consider the natural solution of distributing the cumulative wake-up rate equally between all nodes. Let μ_k be the wake-up rate of node k. The fair solution is $\mu_k = \frac{\mu_c}{N}$ for $k = 1, ..., N$. However, a node does not know and cannot estimate efficiently the number of nodes in its block. By following the same approach as in [12], an Additive Increase and Multiplicative Decrease (AIMD) algorithm of the wake-up rate of each node leads to a fair distribution of the wake-up duties within a single block. 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 of the forwarding region and it compares it with optimal value of the wake-up rate μ_o. If the estimated value is less than or equal to the optimal value, it communicates to the next hop the

$$D_{c-}(h) = \mu_{b,\max}\,\mu_\varepsilon \frac{h\big(\tau\mu_\varepsilon - \mu_{b,\max}h\big) - 2\sqrt{\big(\tau\mu_\varepsilon - \mu_{b,\max}h\big)^2 h + (\mu_{b,\max}h)^2(h-4)}}{\big(\tau\mu_\varepsilon - \mu_{b,\max}h\big)^2 - 4\mu_{b,\max}^2 h} \quad (9)$$

$$D_{c+}(h) = \mu_{b,\max}\,\mu_\varepsilon \frac{\big(h\big(\tau\mu_\varepsilon - \mu_{b,\max}h\big) + 2\sqrt{\big(\tau\mu_\varepsilon - \mu_{b,\max}h\big)^2 h + (\mu_{b,\max}h)^2(h-4)}}{\big(\tau\mu_\varepsilon - \mu_{b,\max}h\big)^2 - 4\mu_{b,\max}^2 h} \quad (10)$$

$$\mu_c(h) = \sqrt{\frac{p\,\lambda\,h\,W_R}{h\,[W_w T_w + W_R(T_{ac} - T_w)] + 2Q_b(\frac{S}{h-1}) + Q_b(\frac{2S}{h-1})(h-2)u(h-2)}}. \quad (11)$$

information to increase additively its wake-up rate, otherwise it orders the next hop node to decrease multiplicatively its wake-up rate. The command on the wake-up rate variation is piggybacked on the data packet and it does not require any additional message.

However, this approach may generate a load balancing problem because of different wake-ups rate among intermediate nodes 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 intermediate node runs a simple reset mechanism in terms of wake-up rate. If the nodes are uniformly deployed with high density between source and destination, we can assign an upper and lower bounds of wake-up rate for each node. If the wake-up rate of a node is larger than the upper bound, $\mu_k > \frac{\mu_o}{N}(1+\xi)$, or is smaller than the lower bound, $\mu_k < \frac{\mu_o}{N}(1-\xi)$, then a node resets its wake-up rate to $\frac{\mu_o}{N}$, where ξ assumes a small value. Otherwise, the node maintains its own wake-up rate, which oscillates between the upper and lower bound.

C. Adaptation Mechanism

Here the mechanisms described in previous Subsections are put together to adapt the wake-up rate, and the transmit radio power of each node. The mechanism makes use of beacons. Recall that each beacon contains the location information of the beacon node to adapt the change of hops in the network. The beacon is used also for synchronization. The adaptation mechanism that each node runs is described next.

- **Init State**: A node sets the number of hops, the wake-up rate, and the estimation of the packet loss probability to an initial values: $h = h_0$, $\mu_i = \mu_0$, $p = p_0$. When a beacon packet is received, the node goes to the Op State.
- **Op State**: When a beacon is received, the node retrieves information on μ_o, h and location information of beacon node L_b, estimates the wake-up rate μ_c of the forwarding region. If $\mu_c < \mu_o$ the node sends an Additive Increase (AI) command, else it sends a Multiplicative Decrease (MD) command on the data packet. Furthermore, the node sets the data transmission power to $P_t(d_k)$ where d_k is the distance between its own location and beacon node L_b. Go back to Init State.
  If a data packet is received, the node retrieves information on wake-up rate update, if AI then $\mu_i = \mu_i + \theta$, else $\mu_i = \frac{\mu_i}{\phi}$, (from the experimental results, we obtained that $\theta = 3$ and $\phi = 1.05$ achieve good performance) Furthermore, the node runs a reset mechanism for load balancing of wake-up rate. Go back to Init State.

D. Computation of the Radio Power

In this Subsection, the minimum radio power that ensures packets to reach a given distance with a given probability is computed. This ensures the minimization of a component of the energy consumption (5) that plays a relevant role in the energy balance [18].

The radio power minimization is based on that the transmit node receives a beacon from the receiver node. From such a beacon, the transmitter node can easily compute the distance from the receiver node. Consider node k transmitting packets with a radio power level P_{tk}. The power of a received signal at the generic distance d_k from node k can be expressed as follows [19]

$$\mathrm{PL}(d_k)_{\,dB} = \mathrm{PL}(d_0)_{\,dB} + 10\,\beta_k\,\log_{10}\left(\frac{d_k}{d_0}\right) + X_k\,, \quad (15)$$

where $\mathrm{PL}(d_0)$ is the path loss computed at a reference distance d_0, β_k is the path loss exponent $2 \leq \beta_k \leq 6$, and X_k is a Gaussian random variable having zero average and variance σ_k^2. It is easy to see that the Signal to Interference plus Noise Ratio (SINR) in dB can be written as follows:

$$\gamma(d_k)_{\,dB} = P_r(d_k)_{\,dB} - P_{n\,dB}\,, \quad (16)$$

where the received power $P_r(d_k)_{\,dB} = P_{t\,dB} - PL(d_k)_{\,dB}$ and P_n denotes the noise floor plus interference.

Assume that nodes use an offset quadrature phase shift keying modulation (O-QPSK), which is used on Tmote Sky sensors [4]. The bit error probability P_b for O-QPSK with coherent demodulation in a slow Rayleigh fading environment, which exhibits non-selective behavior both in frequency and in time, can be expressed as [19]

$$P_b(d_k) \approx \frac{1}{2}\left(1 - \sqrt{\frac{\overline{\gamma}(d_k)}{1+\overline{\gamma}(d_k)}}\right), \quad (17)$$

where $\overline{\gamma}(d_k)$ is the average SINR at the distance d_k. Recalling that the power attenuation follows an exponential decay with respect to distance, and a Gaussian attenuation, the SINR follows a log normal distribution. Hence, it is easy to see that the average of the SINR is given as follows

$$\overline{\gamma}(d_k) = e^{\mu_{\gamma_k} + \sigma^2_{\gamma_k}/2} \,, \tag{18}$$

where μ_{γ_k} and σ_{γ_k} are, respectively, the average and standard deviation of the SINR in neper unit. By recalling the relation of the neper units with the dB units, and Equation (18), the following relation holds true

$$\mu_{\gamma_k} = \varphi \left[P_{t_k \; dB} - \mathrm{PL}(d_0)_{\; dB} - 10\,\beta_k \, \log_{10}\left(\frac{d_k}{d_0}\right) - P_{n \; dB} \right] ,$$
$$\sigma_{\gamma_k} = \varphi \sigma_k \,. \tag{19}$$

where $\varphi = \ln 10/10$. Given the packet size l, the probability of successful packet reception P_s at a distance d_k is

$$P_s(d_k) = [1 - P_b(d_k)]^l \,. \tag{20}$$

By imposing a constraint P_{con} on the probability of successful packet reception at a distance d_k from node k, we can translate the constraint on the average SINR by (17) and (20), thus obtaining a bound $\overline{\gamma}_c$. From this we can derive the transmit radio power necessary to successfully receive packets at a distance d_k with the probability P_{con}. After simple algebra, we have that the minimum transmit power is

$$P_{t \; dB}(d_k) = \log \overline{\gamma}_c + \mathrm{PL}(d_0)_{\; dB} + 10\,\beta_k \, \log\left(\frac{d_k}{d_0}\right) + \log P_n - \frac{\ln 10}{20}\sigma^2 \,. \tag{21}$$

VI. Experimental Implementation

Breath protocol is validated through an extensive set of experiments. The protocol was implemented on a test bed of Tmote Sky [4] wireless sensor nodes. The experiments enable us to assess Breath in terms of delay, reliable packet transmission and energy consumption of the network. Breath is compared with a standard implementation of IEEE 802.15.4 [6], as we discuss next.

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

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

A single node acted as source and generated packets periodically at different rates ($\lambda = 5$, 10 and 15 pckt/s). 15 intermediate nodes were placed to mimic the topology in Fig. 1. The sources was at a distance of 20 m far from the destination. The destination node collected packets and then forwarded them through the USB port to a computer, where the optimal solution is computed as described in IV-C. The latency requirement was set to ($\tau = 1$s) and the reliability to ($\Omega = 0.9$ and 0.95). These requirements were chosen as representative of some control applications in automation (e.g., latency for the activation of fans, heaters).

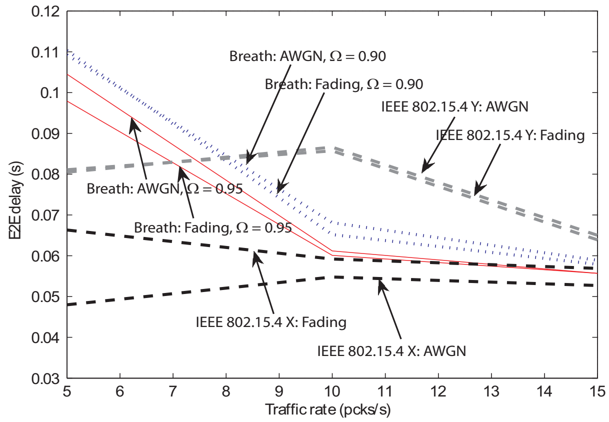

Fig. 3. Temporal average of the end-to-end delay of Breath and IEEE 802.15.4 X, Y with reliability constraint 0.9 and 0.95 over the traffic rate 5, 10 and 15 pckt/s in AWGN and Rayleigh fading environment.

As discussed in the Section I, no comprehensive protocol can be found in the literature which optimizes simultaneously physical layer, MAC and routing in a realistic multi-hop communication scenario with end-to-end reliability and latency constraints. Therefore, we decided to compare Breath against an implementation of the unslotted IEEE 802.15.4 [6], which is similar to the 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 listening time of the nodes in Breath). We defined the case X as the one in which the IEEE 802.15.4 is set with $C = 1$, whereas we defined the case Y setting $C = 4$. Therefore, the case Y represents a fair comparison between Breath and IEEE 802.15.4, while in the case X 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 -5dBm. We set the IEEE 802.15.4 protocol parameters to *macMinBE*$= 3$, *aMaxBE*$= 5$, *macMaxCSMABackoffs*$= 4$. We remark here that other values for such parameters basically give the same trends in the experimental results.

A. End-to-End delay

In this section, we report the experimental results related to the end-to-end delay.

In Fig. 3, the temporal average of the end-to-end delay for both Breath, and IEEE 802.15.4 X and Y are plotted as function of the reliability constraint Ω and traffic rate λ in AWGN and Rayleigh fading environment. The variance of the end-to-end delay exhibits similar behavior as the magnitude, so we do not report it due to lack of space. The end-to-end delay meets perfectly the constrains. Observe that end-to-end delay decreases as the traffic rate increases. This is due to the fact that Breath linearly increases the active time of the

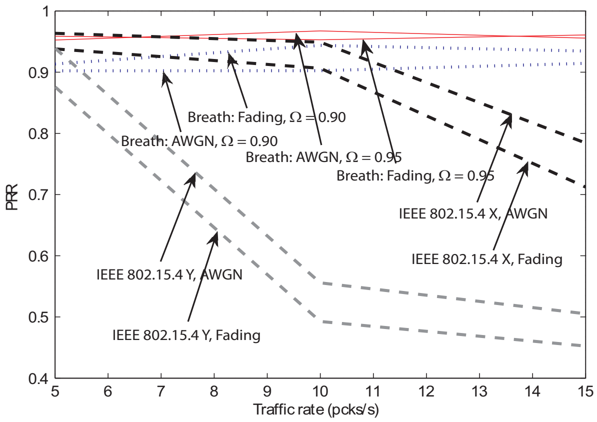

Fig. 4. Packet Reception Rate both in IEEE 802.15.4 X, Y, and Breath protocol with reliability constraint 0.9 and 0.95 over the traffic rate 5, 10 and 15 pckt/s in AWGN and Rayleigh fading environment.

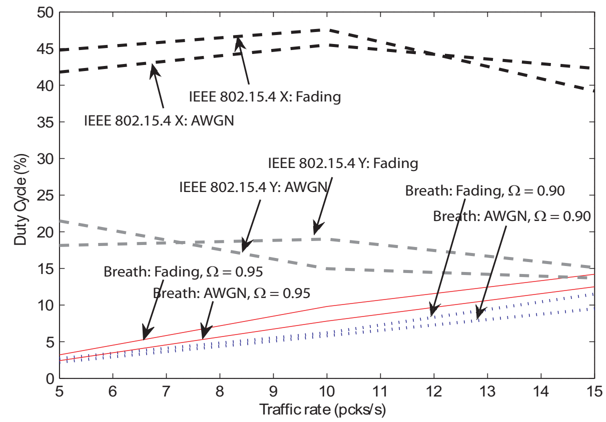

Fig. 5. Temporal average of the duty cycle both in IEEE 802.15.4 X, Y and Breath protocol with reliability constraint 0.9 and 0.95 over the traffic rate 5, 10 and 15 pckt/s in AWGN and Rayleigh fading environment.

nodes as the traffic rate increases (see Equation (12)). The end-to-end delay is larger for worse reliability constraints. The reason of this is found in the fact that Equation (12) increases as the reliability constraint Ω. IEEE 802.15.4 X has lower delay than IEEE 802.15.4 Y because nodes have higher wake-up time. Breath has an intermediate behavior with respect to IEEE 802.15.4 X and Y after $\lambda = 7$. From these experimental results, we conclude that Breath and IEEE 802.15.4 meet perfectly the latency requirements.

B. Packet Reception Rate

In this section we report the packet loss results of Breath.

Fig. 4 shows the packet reception rate (PRR) of Breath and IEEE 802.15.4 X, Y as function of the reliability constraint $\Omega = 0.9$, $\Omega = 0.95$ and traffic rate $\lambda = 5$, 10, 15 pckt/s in AWGN and Rayleigh fading environment. Observe that the PRR is stable around the required reliability for Breath, and in any different traffic rate and environment. However, IEEE 802.15.4 X and Y do not ensure the constraint satisfaction for large traffic rates. Specifically, IEEE 802.15.4 Y show poor PRR in any case, and performance worsen as the the environment moves from the AWGN to the Rayleigh. Furthermore, even though IEEE 802.15.4 X has a higher duty-cycle in Fig. 5, it does not guarantee the better PRR in higher traffic rate. The reason is found in the sleep schedule of the IEEE 802.15.4 case: the wake-up rate of the fixed sleep schedule is independent from traffic rate and wireless channel condition i.e., the fixed sleep schedule is not feasible to support high traffic and unstable wireless channel. Moreover, the fixed sleep schedule does not guarantee the uniform distribution of cumulative wake-up rate within certain time in a cluster, which means that there may be congestion in a cluster. On the contrary, Breath presents an excellent behavior in any situation of channel condition and traffic load.

C. Duty Cycle

In this section we study the energy consumption of the nodes.

As energy performance index, we measured the node's duty cycle, which is the ratio of the active time to the sleep time of a node. Obviously, the lower is the duty cycle, the better is the performance of the protocol in terms of energy consumption.

Fig. 5 shows the temporal average of duty cycle of Breath and IEEE 802.15.4 X, Y with respect to the different traffic rates $\lambda = 5$, 10, 15 pckt/s and $\Omega = 0.9$, $\Omega = 0.95$ in AWGN and Rayleigh fading environment. Note that IEEE 802.15.4 X and Y do not exhibit a clear relationship with respect to traffic rate and have almost flat duty cycle around 42% and 18% because of fixed sleep time. However, recall that the active time is influenced also by the packet transmission attempts. This explain why IEEE 802.15.4 X and Y do not have a fixed duty cycle, even though they have a fixed listen time. Considering Breath, observe that the duty cycle increases linearly with the traffic rate and reliability constraint. As for the end-to-end delay, this is explained recalling Equation (12). Recalling the analysis in Section IV-C, since Breath minimizes the total energy consumption on the base of a trade-off between wake-up rate and waiting time of beacon messages, lower wake-up rates do not guarantee lower duty cycle. Observe that choosing an active time for the nodes of the IEEE 802.15.4 implementation would obviously obtain energy savings comparable with Breath, however, the reliability would be heavily affected (recall Fig. 4). More precisely, ensuring a duty cycle for the IEEE 802.15.4 implementation comparable with Breath would be very detrimental with respect to the reliability.

Fig 6 shows the duty cycle experimental results of each intermediate node for $\lambda = 5$ pckt/s and $\Omega = 0.95$. A fair uniform distribution of the duty cycles among all intermediate nodes is achieved. This is an important result, because 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. 7 reports the case of variable number of intermediate nodes between the source and the destination in an AWGN environment. The figures shows how much Breath extends the network lifetime compared with IEEE 802.15.4 X and Y, as

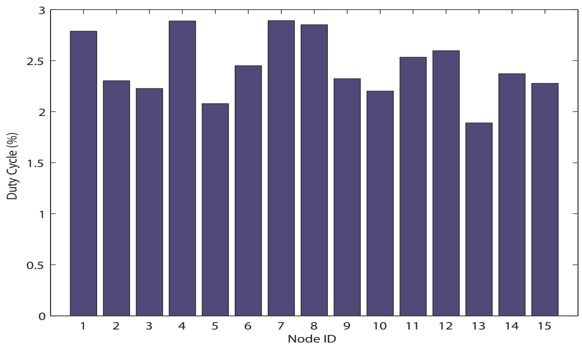

Fig. 6. Distribution of the duty cycle in each node with 15 intermediate nodes. The reliability constraint is 0.95 and traffic rate is 5 pckt/s.

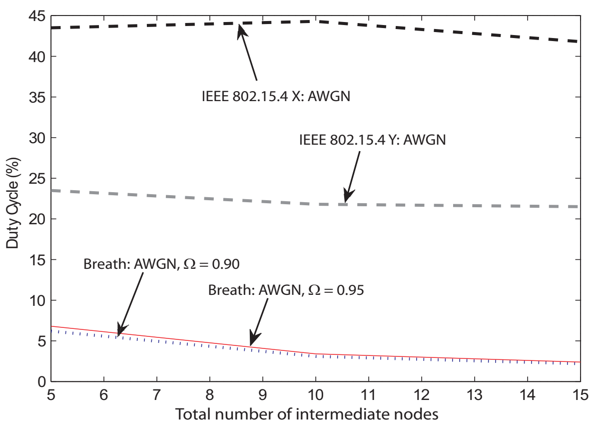

Fig. 7. Temporal average of the duty cycle both in IEEE 802.15.4 X, Y and Breath protocol with reliability constraint 0.9 and 0.95 over the different number of intermediate nodes 5, 10 and 15 pckt/s in AWGN environment.

a function of node density. Observe that the duty cycle has proportional relation with respect to density of nodes. With respect to protocols as [11], this is one of the remarkable strong points of Breath: the network lifetime is extended by adding more nodes.

Finally, we observe that Breath does an accurate radio power control, so that further energy savings are actually obtained with respect to the IEEE 802.15.4 implementation.

VII. Conclusions

We designed and implemented the Breath protocol, a cross-layer protocol for WSNs for real-time control and automation. The protocol considers physical layer aspects (e.g., power control, duty-cycling), randomized MAC and routing. The protocol maximizes the network lifetime under reliability and end-to-end delay constraints.

We provided a test-bed implementation of the protocol, building a WSN with TinyOS and Tmote Sky wireless sensors. An experimental campaign was conducted in order to test the validity of Breath in an indoor environment with both AWGN and Rayleigh fading. Experimental results showed that the protocol achieves the required reliability and the latency constraints, while minimizing the energy consumption. It outperformed significantly a standard IEEE 802.15.4 implementation in terms of both energy consumption and reliability. Breath showed good load balancing performance, and was well scalable with the number of nodes.

Future work includes a performance limit analysis, i.e., we plan to characterize the maximum number of nodes that can be supported by our protocol, and the minimum end-to-end delay achievable. We also plan to test the validity of Breath for outdoor applications. An initial such activity, along with further experimental results, can be found in [16].

References

[1] R. Cristescu, B. Beferull-Lozano, M. Vetterli, and R. Wattenhofer, "Network correlated data gathering with explicit communication: NP-completeness and algorithms," *IEEE/ACM Transactions on Networking*, vol. 14, no. 1, December 2006.

[2] S. Madden, "The design and evaluation of a query processing architecture for sensor networks," *Ph.D. Dissertation, UC Berkeley*, 2003.

[3] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of network density on data aggregation in wireless sensor networks," in *22nd IEEE International Conference on Distributed Computing Systems*, 2002.

[4] *Tmote Sky Data Sheet*, Moteiv, San Francisco, CA, 2006. [Online]. Available: http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf

[5] P. Buonadonna, D. Gay, J. Hellerstein, W. Hong, and S. Madden, "Task: Sensor network in a box," Intel Research Lab Report, Tech. Rep., 2007.

[6] *IEEE 802.15.4*, IEEE, 3 Park Avenue, New York, USA, 2006. [Online]. Available: http://www.ieee802.org/15/pub/TG4.html

[7] The ZigBee Alliance , "http://www.zigbee.org."

[8] R. Shah, A. Bonivento, D. Petrovic, E. Lin, J. V. Greuen, and J. Rabaey, "Joint optimization of a protocol stack for sensor networks," *MILCOM*, 2004.

[9] M.Zorzi and R.R.Rao, "Energy and latency performance of geographic random forwarding for ad hoc and sensor networks," *IEEE WCNC 2003*, March 2003.

[10] Y. Xu, J. Heidemann, and D. Estrin, Eds., *Geography-Informed Energy Conservation for Ad Hoc Routing*, vol. pp. 70-84. Proc. MobiCom 2001, July 2001.

[11] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *MobiCom*, 2001.

[12] A. B. J. Van Greuen, D. Petrovi C, J. Rabaey, K. Ramchandran, and A. Sangiovanni-Vincentelli, "Adaptive sleep discipline for energy conservation and robustness in dense sensor networks," *ICC*, 2004.

[13] A. Bonivento, C. Fischione, A. Sangiovanni-Vincentelli, "Randomized protocol stack for ubiquitous networks in indoor environment," *In Proc. of IEEE CCNC*, 2006.

[14] A. Bonivento, C. Fischione, L. Necchi, F. Pianegiani, A. Sangiovanni-Vincentelli, "System level design for clustered wireless sensor networks," *IEEE Transactions on Industrial Informatics*, August 2007.

[15] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, October 2002.

[16] P. G. Park, C. Fischione, A. Bonivento, K. H. Johansson, and A. Sangiovanni-Vincentelli, "Breath: a self-adapting protocol for wireless sensor networks," The Royal Institute of Technology, Tech. Rep., 2007.

[17] A. Lymberopoulos, D. Savvides, "XYZ: a motion-enabled, power aware sensor node platform for distributed sensor network applications," *Information Processing in Sensor Networks (IPSN) 2005*, Apr. 2005.

[18] B. Z. Ares, P. G. Park, C. Fischione, A. Speranzon, and K.H.Johansson, "On power control for wireless sensor networks: System model, middleware component and experimental evaluation," *IFAC European Control Conference (ECC'07)*, 2007.

[19] G. L. Stüber, *Priciples of Mobile Communication*. Kluwer Academic Publishers, 1996.