

On Power Control for Wireless Sensor Networks: System Model, Middleware Component and Experimental Evaluation

B. Zurita Ares, P. G. Park, C. Fischione, A. Speranzon, K. H. Johansson

Abstract—In this paper, we investigate strategies for radio power control for wireless sensor networks that guarantee a desired packet error probability. Efficient power control algorithms are of major concern for these networks, not only because the power consumption can be significantly decreased but also because the interference can be reduced, allowing for higher throughput. An analytical model of the Received Signal Strength Indicator (RSSI), which is link quality metric, is proposed. The model relates the RSSI to the Signal to Interference plus Noise Ratio (SINR), and thus provides a connection between the powers and the packet error probability. Two power control mechanisms are studied: a Multiplicative-Increase Additive-Decrease (MIAD) power control described by a Markov chain, and a power control based on the average packet error rate. A component-based software implementation using the Contiki operating system is provided for both the power control mechanisms. Experimental results are reported for a test-bed with Telos motes.

Index Terms: Power Control, Wireless Sensor Networks, Networked Embedded Systems.

I. INTRODUCTION

The design of efficient power control algorithms for wireless sensor networks (WSNs) is crucial to reduce energy consumption to a level suitable for many applications [1]. This is for instance the case in the disaster relief scenario considered in this paper, where wireless sensor nodes are used to provide real-time information on the physical conditions within a road tunnel. In such a scenario, where recharging is typically not possible, radio power control is very important in order to increase the network lifetime. Transmission power is responsible for up to 70% of the total energy consumption for off-the-shelf sensor nodes, e.g., [2], [3]. Keeping the transmit power under control is furthermore beneficial to reduce the packet collision probability, which otherwise leads to more retransmitted packets wasting even more energy.

When facing the problem of power control for WSNs, traditional control strategies cannot be applied due to low signal bandwidths in the control loop, highly nonlinear and uncertain system models, and limited memory and processing resources. To ensure an adequate lifetime, WSN protocols should exploit platform-based design techniques to minimize the energy consumption in a coordinated way [4], [5]. In fact, the conventional layered OSI protocol model

is not adequate for efficient WSN protocols. The protocol stack need to be complemented with a middleware that provides the control application with the appropriate interface to the network layers. A middleware approach offers a suitable abstraction of the platform and it allows an efficient software implementation of the controller in a multi-application environment. An appropriate middleware should effectively hide the complexity and heterogeneity of the hardware platform and software architecture from the application programmer [5]. An early example of this design methodology applied to WSNs is given in [6].

The main contribution of this paper is in component-based design and implementation of power control for WSNs. The component is integrated as part of a disaster relief demonstration with mobile robots and WSNs within the European Integrated Project Reconfigurable Ubiquitous Networked Embedded Systems (RUNES) [7]. The RUNES middleware is described in [8], and an overview of the design of control components is given in [9]. The component described in this paper is aimed at controlling the power consumption of the radio transmission, while guaranteeing a given packet error probability. The sensors in the RUNES demonstration platform are the Telos motes [2], so we characterize the typical measurements of quality of communication used in these sensors, namely, the uncertain Received Signal Strength Indicator (RSSI), and we relate it to the Signal to Noise plus Interference Ratio (SINR). Based on these findings, we develop two power control algorithms: one Multiplicative-Increase Additive-Decrease Power Control (MIAD PC) and one power control based on the estimation of the packet error rate (PER PC). Furthermore, we provide a component-based description for the abstraction of our algorithms and specify the key interfaces for an implementation on the RUNES platform.

In recent years, power control has received intense attention for cellular radio systems and ad-hoc wireless networks. In [10], a power control method is described as a Markov chain. An interesting characterization of power control algorithms from a control theoretic perspective can be found in [11]. With reference to WSNs, the authors of [1] investigate power control in a cooperative diversity scheme, which includes sleeping disciplines for the nodes. Compared to these contributions, and references therein, our original approach consists in modeling the physical layer characteristics that are necessary to implement power control algorithms for off-the-shelf WSNs. Since our studies are related to the Telos motes, we include a detailed link layer model, which includes path loss and slow fading of the RSSI,

Work done in the framework of the RUNES Integrated Project, contract number FP6-IST-004536, and the HYCON Network of Excellence, contract number FP6-IST-511368. The research was partially funded also by the Swedish Foundation for Strategic Research and by the Swedish Research Council. All the authors were with the School of Electrical Engineering of the Royal Institute of Technology, Stockholm, Sweden, when this work was done. E-mail:{carlofi|albspe|kallej}@ee.kth.se

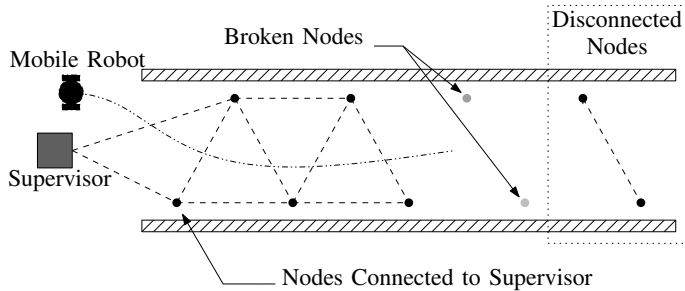


Fig. 1. Physical network reconfiguration scenario: a cluster of network nodes is disconnected from the supervisor in the tunnel control room due to some broken nodes. A mobile robot carries a new node into the broken area to act as a bridge between the two clusters. The node carried by the robot controls the radio power of the edge nodes.

for these sensors. This model enables us to propose a novel power control strategy based on the packet error probability.

The rest of the paper is organized as follows: in Section II, the system scenario and problem formulation is introduced; in Section III the relation between the RSSI and SINR is reported. The MIAD PC and PER PC are summarized in Section IV. A component-based implementation of these power control algorithms is outlined in Section V; in Section VI experimental results are presented, and finally, in Section VII conclusions and future developments are summarized.

II. PROBLEM FORMULATION

We consider the disaster relief tunnel scenario of the RUNES project, in which a cluster of nodes of a WSN deployed in a tunnel is disconnected from the supervisor in the tunnel control room as a consequence of a car accident, see Figure 1. Since all nodes of the network may gather important information, the gap between the connected and the disconnected clusters of sensor nodes should be bridged. This physical network reconfiguration or reconnection is implemented in two steps: first, the nodes on the edge of the connected cluster increase their transmit power. If connection is not re-established, then a mobile robot carries a new node into an appropriate position within the disconnected region to work as a bridge. The node carried by the robot controls the radio power of the edge nodes. The data packets from the edge nodes of the disconnected cluster are transmitted with a power level assigned by the power control algorithm residing in the mobile node. Thereby the data packets from the disconnected cluster are successfully received with a minimum waste of energy and collision probability.

Consider a generic link between transmitter node i on the edge and the controller in the corresponding receiver node. The power control loop can be modeled as in the Figure 2. The power to transmit data packets is denoted P_i . When the signals associated to the data packets are transmitted through the wireless link, they are attenuated by a channel coefficient $\Omega_i(t)$, and then received at the controller corrupted by additive interference, with power denoted $\Phi_i(t)$, and thermal noise, with power spectral density η_0 . The interference is caused by the co-channel transmitters (other nodes in the scenario). If the interference is too high, due,

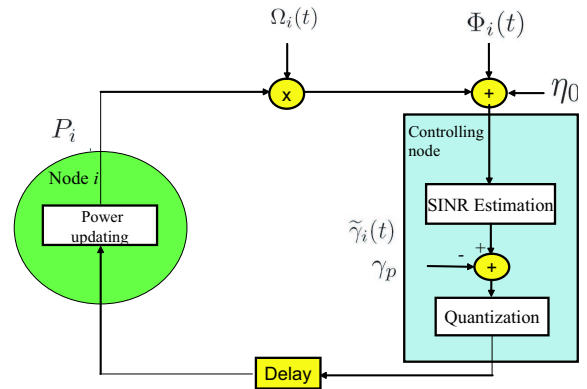


Fig. 2. Power control loop that consists of the transmitter node (on the left) and the receiver node (on the right). The receiver node, which is mounted on the robot, executes the control algorithm.

for example, to collisions with other transmissions, then it is not possible to successfully receive the packets sent by the node i . Thus, a proper transmit power level should be used by each node in order to ensure low interference and radio power consumption.

In the power control loop, the quality of the received signal sent by node i is quantified through its SINR. The value of this statistical variable has to be estimated. It is compared with its reference value γ_p , which is chosen such that there is a low packet error probability from node i to the receiver. According to the comparison between the received SINR and γ_p , a power control level is computed at the receiver. Its quantized value is sent to the transmitter node by a signaling packet. Such a packet reaches the transmitter node after a small delay, which accounts for processing operations and communication. The receiver actuates the power control level, and transmits packets toward the receiver node with the chosen level of power. The power control problem consists consequently in regulating the radio power at the transmitters such that a given communication quality is experienced by the receiver. The power control mechanism tries to react to the SINR fluctuations by controlling the transmitter power to ensure a desired packet error probability.

III. RADIO MODEL

To design the power control mechanism, we need to model the SINR of the communication link for each transmitter–receiver pair. The SINR depends, among the other things, on the implementation platform. Here we consider the Telos mote sensor nodes [2], which are the ones used in the RUNES tunnel scenario. The transceiver of these nodes uses a Direct Sequence Spread Spectrum (DSSS) technique. Data are coded according to a DSSS operation, and then transmitted through a Carrier Sensing Multiple Access/ Collision Avoidance (CSMA/CA) technique. The radio power level used by (edge) node i is denoted P_i , $i = 1, \dots, K$. The same fixed bandwidth W , and hence the same chip interval T_c , is allocated for each communication link. The processing gain is denoted $G = W/R_b = T_b/T_c$, where R_b is the bit rate

and T_b the bit interval. The transmitted signal, after being attenuated by the wireless channel, is received corrupted by additive Gaussian noise and multi access interference (MAI) caused by other transmitting nodes and co-channel interferences, as indicated in Figure 2. At the receiver, the signal is despread, demodulated, and decoded in order to get the source data.

Following an approach similar to [12], we can express the received signal corresponding to the signal transmitted by node i as

$$Z_i(t) = D_i(t) + I_i(t) + N_g(t), \quad (1)$$

where $D_i(t)$ is the desired signal, $I_i(t)$ is the interference due to the presence of multiple transmitting nodes (causing MAI), and $N_g(t)$ is a Gaussian random variable with zero mean and variance η_0 modeling thermal noise. Specifically, it can be shown that

$$D_i(t) = \sqrt{P_i \Omega_i(t)}, \quad (2)$$

and that the variance of the MAI term is

$$E\{I_i^2(t)\} = \frac{1}{3G} \sum_{j=1, j \neq i}^K \nu_j(t) P_j \Omega_j(t) + I_n(t), \quad (3)$$

where $I_n(t)$ includes the co-channel interferences affected by the considered power control loop, see [13].

The term $\nu_i(t)$ is a binary random variable that models the CSMA/CA transmission attempts of the nodes. We have $\nu_i(t) = 1$ if a node has the permission to transmit, and $\nu_i(t) = 0$ otherwise. We assume $\Pr(\nu_i(t) = 1) = \alpha_i$ and $\Pr(\nu_i(t) = 0) = 1 - \alpha_i$. The wireless channel coefficient associated to the considered path is defined as

$$\Omega_i(t) = \text{PL}_i(t) e^{\xi_i(t)}. \quad (4)$$

where $\text{PL}_i(t)$ is the path loss, and $e^{\xi_i(t)}$ is the shadow fading component over the path, with $\xi_i(t)$ being a Gaussian random variable with zero average and standard deviation σ_{ξ_i} . The path loss (in dB) can be described as

$$\text{PL}_i(t)|_{\text{dB}} = -\text{PL}_i(d_r)|_{\text{dB}} - 10n \log_{10}(d_i(t)/d_r), \quad (5)$$

where $\text{PL}_i(d_r)$ is the path loss computed at the reference distance d_r , d_i is the transmitter–receiver distance, and n is the path-loss decay constant.

The quality of the received signal $Z_i(t)$ is measured by its SINR. For the generic transmitter–receiver pair i , the SINR is derived as follows [13]:

$$\gamma_i(t) = \frac{P_i \Omega_i(t)}{\frac{N_0}{2T_b} + \frac{1}{3G} \sum_{j=1, j \neq i}^K \nu_j(t) P_j \Omega_j(t) + I_n(t)}. \quad (6)$$

Note that the SINR is a stochastic process, since it depends on the wireless channel coefficient $\Omega_i(t)$ and on the binary variable $\nu_i(t)$. Obviously, the SINR can be directly influenced by the transmit powers P_i .

IV. SINR ESTIMATION

Power control algorithms based on the radio model described in previous section require estimates of the SINR.

The Telos motes do not provide the SINR as a performance measure of the communication link, but instead the RSSI is given. In this section, we therefore derive a relation between the RSSI and the SINR, which indicates how the SINR is estimated in the implementation.

The Chipcon CC2420 on the Telos motes gives RSSI measurements in dBm defined as an average of the received signal power calculated over 8 symbol periods (i.e., $128\mu\text{s}$). We propose the following model for the RSSI at the receiver from transmitter i :

$$\begin{aligned} \text{RSSI}_i(t) = & P_i|_{\text{dB}} - P_l(d_r)|_{\text{dB}} - 10n \log_{10}(d_i(t)/d_r) \\ & + \varphi \xi_i(t) + \Phi_i(t) + C + \delta(t) + 30, \end{aligned} \quad (7)$$

where the first term is the transmitted power, the second and third are due to the path loss, and the fourth term is the shadowing component times the constant $\varphi = 10 \log_{10} e$ (with e being the base of the natural logarithm). The term $\Phi_i(t)$ is a zero-average stochastic process taking into account the interfering powers coming from co-channel transmitters:

$$\Phi_i(t) = f \left(10 \log_{10} \left(\frac{1}{3G} \sum_{j=1, j \neq i}^K \alpha P_j \Omega_j(t) + I_n(t) \right) \right),$$

where $f(\cdot)$ is a linear function, which comes from sample average used to compute the RSSI. The constant C is the measurement offset, which is empirically measured, and equal to $C = 45$ dB in our experimental setup. The term $\delta(t)$ represents measurement uncertainty, which is specified to be bounded by ± 6 dB for the Telos motes. Finally, the term 30 accounts for the conversion of dBm to dB.

To obtain the SINR (6) from the RSSI (7), we need to do some simplifications. By the hypothesis that there are few interferers and that the probability α to access the channel is low, the interfering term is small. Furthermore, the function $f(\cdot)$ is a sample average since the RSSI is defined as an average over 8 symbols, thus the interference powers collected by the receiver node is further smoothed. Hence, we assume that $\Phi_i(t)$ is negligible. We can therefore derive the SINR from the RSSI as follows

$$10 \log_{10} \gamma_i(t) \approx \text{RSSI}_i(t) - \eta_0 - C - 30. \quad (8)$$

An estimate of the SINR can thus be derived as

$$\tilde{\gamma}_i(t) = 10^{(\text{RSSI}_i(t) - \eta_0 - C - 30)/10}. \quad (9)$$

V. POWER CONTROL ALGORITHMS

In this section, we develop two power control algorithms: one Multiplicative-Increase Additive-Decrease Power Control (MIAD PC) and one power control based on the estimation of the packet error rate (PER PC).

A. MIAD Power Control

A simple adjustment rule for the transmit power can be based on the following mechanism [10]: when an erroneous packet is detected, the power P_i is increased by $d\Delta$, where d is an integer and Δ the step size, whereas each correctly received packet imposes a decrease of the transmit power by

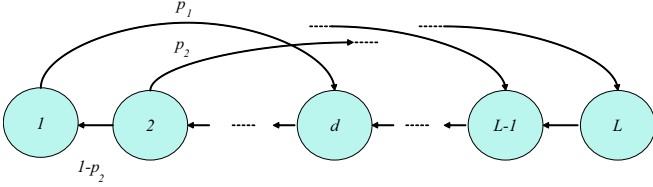


Fig. 3. Markov chain model of the MIAD power control algorithm.

Δ . The parameters d and Δ obviously influence performance of the packet error rate process, and the power consumption.

The dynamics of power transmission can be modeled through a discrete Markov chain, see Figure 3. Transmitted power is quantized into L values and each value is associated to a state of the chain. Detection and power control force jumps from one state to another. According to this scheme, there is a Δ -length jump forward (to the right in the figure) if a packet error occurs, and a single jump backward if a packet is correctly received. If p is the desired packet loss probability, then it should hold that $\Delta = p/(1 - p)$. The transition probabilities from one state to another depend on the packet error probability, whereas the steady-state probabilities π_k , $k = 1, \dots, L$, can be easily computed through the equilibrium equations

$$\begin{aligned} \pi_k &= (1 - F_{k+1})\pi_{k+1}, & 1 \leq k < d \\ \pi_k &= F_{k-d}\pi_{k-d} + (1 - F_{k+1})\pi_{k+1}, & d \leq k \leq L - 1 \\ \pi_P &= \pi_{P-d}F_{P-d}, \end{aligned}$$

together with $\sum_{k=1}^P \pi_k = 1$. Here F_k denotes the packet error probability associated to level k of the transmit power. An analytical model of the transition probabilities enables the analysis of the packet loss process. This is for example useful in situations when an ARQ protocol is used for the retransmission of erroneously received packets. Indeed, the Markov model allows for an accurate characterization of the delay of packet delivery [14]. This characterization is beyond the scope of the paper.

The implementation of the MIAD PC requires knowledge of the packet error probability, once a power level is fixed. Such a probability can be estimated either using an analytical model of the SINR, or simply observing the sequence of the packets erroneously received. The first solution is the most accurate, but requires an estimation of the channel statistics. This estimation may be difficult on computationally constrained sensor nodes, since some accurate signal processing is required to estimate the average and standard deviation of the transmitted and interfering signals. The second solution is simpler, as it requires just the computation of the number of erroneous packets. The disadvantage of this solution is that it takes some time to collect an accurate estimation of the loss process. Hence, the packet error probability could be over- or under-estimated, leading to an too high power consumption or to a too high packet error rate, respectively. In the experimental result section, we present an implementation of the MIAD PC with a real-time estimation of the packet

error rate.

B. PER Power Control

When a model of the wireless channel is available, the packet error probability can be analytically computed for each communication link. By setting a constraint on the probability, the transmit power can be derived. Let us denote the packet error probability of the node i by $F(P_i)$, indicating the dependence on the transmit power P_i . Under the assumption of bit-to-bit error independence, we have [15]:

$$F(P_i) = 1 - [1 - F_b(P_i)]^l, \quad (10)$$

where l is the number of bits of a packet, and $F_b(P_i)$ is the average bit error probability. Such a probability has to be computed according to the modulation scheme and the wireless propagation statistics. The Telos motes use the O-QPSK (offset quadrature phase shift keying) modulation. In a AWGN wireless channel, it is easy to see that the bit error probability is given as follows:

$$F_b(P_i) = \frac{1}{2}Q(\sqrt{\bar{\gamma}(P_i)}), \quad (11)$$

where $Q(x) = 1/\sqrt{2\pi} \int_x^\infty e^{-t^2/2} dt$ is the complementary standard Gaussian distribution. In a slow fading environment, which exhibits non-selective behavior in frequency and time, the bit error rate can be expressed as

$$F_b(P_i) = \frac{1}{2} \left(1 - \sqrt{\frac{\bar{\gamma}(P_i)}{1 + \bar{\gamma}(P_i)}} \right), \quad (12)$$

where $\bar{\gamma}(P_i)$ is the average SINR.

Equation (9) gives that the SINR has a log normal distribution. Hence, its average is given by [16]:

$$\bar{\gamma}(P_i) = e^{\mu_\gamma + \sigma_\gamma^2/2}, \quad (13)$$

where

$$\mu_\gamma = 2 \ln M_i^{(1)}(P_i) - \frac{1}{2} \ln M_i^{(2)}(P_i) \quad (14)$$

$$\sigma_\gamma^2 = \ln M_i^{(2)}(P_i) - 2 \ln M_i^{(1)}(P_i). \quad (15)$$

The terms $M_i^{(1)}(P_i)$ and $M_i^{(2)}(P_i)$ are statistical expectations of the first two moments of the SINR.

By setting a constraint on the packet error probability, say p , the corresponding average SINR $\bar{\gamma}_p$ can be derived by (10) and (11) for the AWGN environment, and from (10) and (12) for the Rayleigh case. At the receiver, the transmit power can be computed in the controller such that the average SINR is met. The algorithm is as follows: The average SINR for transmitter i is computed using (13)–(15), where the statistical expectations (14) and (15) are estimated using sample averages. Denote the period time of the power control by T and denote the number of samples of the RSSI

collected during one period by M . We then have

$$\widetilde{M}_i^{(1)}(nT) = \frac{1}{M} \sum_{j=1}^M \widetilde{\gamma}_i(nT + j) \quad (16)$$

$$\widetilde{M}_i^{(2)}(nT) = \frac{1}{M} \sum_{j=1}^M \widetilde{\gamma}_i^2(nT + j). \quad (17)$$

Let the average SINR computed using (13) and (16)–(17) at time nT be denoted $\bar{\gamma}(nT)$ (with a slight abuse of notation). The power that the transmitter has to use at time $nT + T$ is calculated in the receiver as

$$P_i(nT + T) = \bar{\gamma}_p \frac{P_i(nT)}{\bar{\gamma}(nT)}. \quad (18)$$

This value (or its quantization) is communicated to the transmitter.

VI. POWER CONTROL COMPONENTS

A. The Middleware Component Framework

The implementation of the power control mechanism has been carried out using the software architecture developed in the RUNES project [17] and the operating system Contiki [18], which is developed for low-memory low-computation devices such as the Telos motes.

The basic building block of the middleware developed in RUNES is a software component. From an abstract point of view, a component is an autonomous software module with well-defined functionalities that can interact with other components only through interfaces and receptacles. Interfaces are a sets of functions, variables and associated data types that are accessible by other components. Receptacles are required interfaces by a component and thus they make explicit the inter-component dependencies. The connection of two components occurs between a single interface and a single receptacle. Such an association is called a binding.

The implementation of the component model for Contiki is called a component runtime kernel (CRTK). It allows the instantiation of a variety of components and the dynamic run-time binding of them. It provides a set of functions that allows a component to be substituted with another component which has the same interface.

B. The Power Control Components

Power control components (PCCs) have been developed based on the Contiki CRTK. Although the implementations are tailored for the RUNES scenario, they are easy to modify to other situations and applications. The PCCs described in this section implements both the MIAD PC and the PER PC.

The PCCs reside in the network node carried by the mobile robot. The components consist of infinite loops that wait for events to take place. The PCC for the MIAD PC monitors the event `PROCESS_EVENT_PCC`, which causes the power control algorithm to be executed. When it executes, a PER estimator tracks the number of received packets for each link and periodically generates an event `PROCESS_EVENT_PER` to notify the estimated PER. When `PROCESS_EVENT_PER` takes place, the transmission powers of the nodes are adjusted

based on the estimated PER. The update of the power is done on the basis of the MIAD PC presented in Section V-A. The PCC for the PER PC monitors the same event as in the MIAD PC, the `PROCESS_EVENT_PCC`, which causes the power control algorithm to be executed as described in section V-B.

The PCCs offer interfaces with three functions for executing the power control algorithm, setting the transmission power, and for retrieving the current transmission power. An excerpt from the interface file is presented below:

```
DECLARE_INTERFACE(iexample,
{
    void (* run_pwr_alg)(
        uint8_t minpwrlvl,
        uint8_t maxpwrlvl,
        uint8_t run);
    void (* set_pwr)(
        uint8_t pwrlvl);
    void (* get_pwr)(
        uint8_t *pwrlvl);
});
```

The Contiki CRTK provides the necessary kernel-level functions that allow other components to use the functions declared in the interface of the PCC. A generic component that needs to be bound to the PCC requires the declaration of a receptacle for the power control component, and invokes a kernel function that creates the binding. An example of how to use the PCC component is as follows:

- 1) Declare the receptacle:

```
static Receptacle myrec_r;
```
- 2) Bind to the PCC component:

```
INVOKE("crtk", crt_k,
    connect("pcc", &myrec_r));
```
- 3) Call the function:

```
INVOKE(myrec_r, interface_name,
    run_pwr_alg(MAX, MIN, YES));
```

When the function `run_pwr_alg` is invoked, the event `PROCESS_EVENT_PCC` is triggered and the power control algorithm starts running.

The computation involved in the PER PC may be quite expensive for the resource limited hardware platform. In particular the computation of (16) and (17), and then (13), requires significant resources in terms of processing time of the microcontroller. In order to reduce the computational complexity of the PER PC, we adopted the strategies described in the sequel.

In the PER PC for the AWGN environment, the variance of the SINR can be considered very low. Therefore, only (16) has to be estimated, while $M_i^{(1)}(P_i) \approx \sqrt{M_i^{(2)}(P_i)}$. Hence (13) reduces to the simple expression

$$\bar{\gamma}(P_i) = M_i^{(1)}(P_i). \quad (19)$$

The computation of the inverse of the Q function in (11) can be simplified using an array of which puts in correspondence the desired bit error rate, with the $\bar{\gamma}(P_i)$. Furthermore, the power updating is not always necessary, but only when (19)

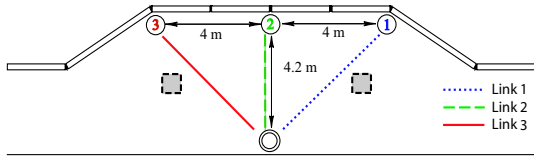


Fig. 4. Experimental layout with three transmitter nodes (circles) and one receiver node (double circle). The transmitter node corresponds to three sensors and the receiver node to the node attached to the mobile robot.

changes over periods of time. Since significant variations of (19) are not frequent, the component sends power updating commands with low rate.

In the case of PER PC in the Rayleigh environment, computations cannot be simplified, as the variance of the wireless channel may be quite high and time varying. For the same argument, adopting the idea of sending power control commands only when the sample average changes, does not help. In the numerical section, we will discuss the effects of the computational processing.

VII. EXPERIMENTAL RESULTS

In this section, we present experiments performed with the power control components and Telos motes in an indoor environment. Figure 4 shows a setup that mimics a possible situation in the physical network reconfiguration scenario described in Section II. In the figure, three stationary sensor/transmitter nodes are displayed as circles with numbers identifying the nodes. The double circle indicates the mobile robot with the receiver node. The links between the transmitter nodes and the receiver are correspondingly denoted links 1–3. The three grey rectangular objects are metal obstacles.

In order to evaluate the power control performance, we did experiments with both the MIAD PC and the PER PC. The parameters of the experiments are as follows. A transmission rate of 10 data packets per second is used. The signaling packets sent by the receiver node to the transmitter nodes are beacons with a periodicity of one beacon every 2 s. Knowing the sensor’s transmission rate, the receiver node can detect packet losses. Specifically, the `PROCESS_EVENT_PER` event of the PER module is generated every 2 s. The expected number of packets to be received for each link is 10 packets per second. The difference between this value and the received number of packets gives the estimation of the PER. The experiments were ran for 350 s, a time long enough so that all the variations of the wireless channel were averaged out. We set $d = 1$ for the MIAD PC. For each experiment, we considered static and time-varying conditions. The static case corresponds to a fixed position of the robot, which is put in line of sight with the transmitters. In this case, the wireless channel is well described by an AWGN model, so that we can use (11) to estimate the bit error probability. In the time-varying case, the robot is moved around its initial position, along a line of 25 cm, and a metal object is put in front of it, so the transmitter and the receiver are not in line-of-sight. The motion of the robot on a short distance

allows to neglect the variation of the wireless channel due to shadow fading and path loss, so the channel can be well described by a Rayleigh fading distribution, and we can use (12) for the bit error rate.

To characterize the energy consumption, we define the gain of using power control as

$$\rho_i = \frac{P_{\max}}{P_{\text{avg}, i}}, \quad (20)$$

where $P_{\max} = 0$ dBm is the radio power corresponding to the maximum power level, and $P_{\text{avg}, i}$ is the average power consumed by node i . A high ρ_i and a low packet error rate indicate a good behavior of the power control algorithm.

TABLE I
GAIN AND PACKET ERROR RATE OF MIAD IN A AWGN ENVIRONMENT
FOR TWO DIFFERENT VALUES OF THE STEP SIZE Δ .

Link	$\Delta = 2$		$\Delta = 4$	
	ρ_i	PER %	ρ_i	PER %
1	7.23	0.16	5.27	4.10
2	7.33	2.88	7.51	0
3	7.46	0.16	7.51	0

In Tab. I, we report the energy gain and the PER for the MIAD PC in the AWGN channel, for two different values of the step size Δ . The links have similar energy efficiency and PER in the case of $\Delta = 2$. The energy consumption decreases for the second and third link when $\Delta = 4$, whereas the packet error rate is zero for the same links, and it is around 5% for the first link. This can be explained considering that in the case of $\Delta = 4$, lower standard deviations of the AWGN noise were experienced for links 2 and 3 with respect to the case with $\Delta = 2$. By the contrary, considering $\Delta = 4$, link 1 has higher packet error rates and, at the same time, higher energy consumption. Since the channel is AWGN, i.e. with low values of the standard deviation of the RSSI, it is not energy efficient to use large values of the step size on such link.

In Fig. 5, we report the average RSSI evolution over time for $\Delta = 2$. The average is taken over the samples collected during a window of 2 s. The RSSI is basically flat for long periods of time (meaning low standard deviation of the AWGN channel). The spikes correspond to the increase of the transmit power as consequence of the MIAD PC. The initial high values of the RSSI are due to the initialization of the MIAD PC, when the powers are set to the maximum. As can be observed, link 1 experiences the larger number of jumps, thus consuming more power. This is confirmed by Fig. 6, where the temporal evolution of the power is reported for each link. Note that the power levels are expressed by the 29 values index used in the Telos specifications, where 3 corresponds to -25 dBm, and 31 to 0 dBm. The power is increased for each unsuccessful transmission, while decreased for each successful packet reception. The closed-loop behavior is thus a periodic oscillation as illustrated in the figures. Notice the large power consumption between

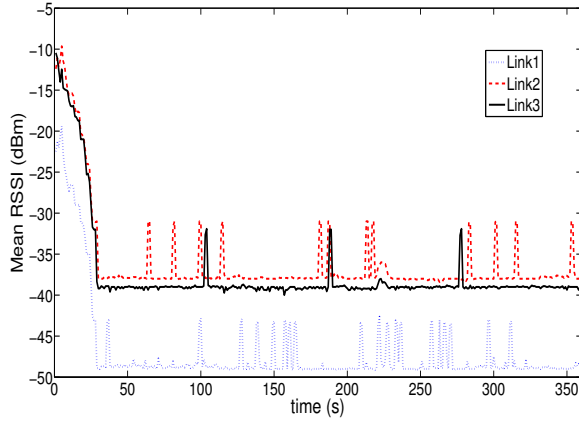


Fig. 5. RSSI for each link as function of time for the AWGN environment, MIAD PC with $d = 2$.

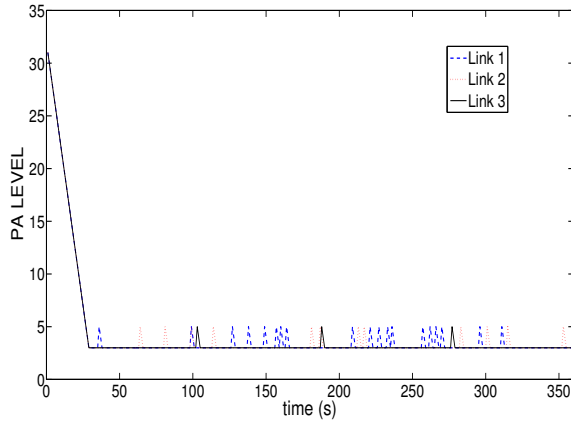


Fig. 6. Power Consumption for each link as function of time for the AWGN environment, MIAD PC with $d = 2$.

130 s and 170 s for link 1, which matches the frequent spikes in Fig. 5.

In Tab. II, the energy gain and the PER is reported for the MIAD PC in a Rayleigh environment. It is interesting to observe that a low value of the step size causes PER around 5% and 10%. As soon as Δ is increased there is a decrease of the PER, which is between 2.88% and 3.65%. Therefore, it is confirmed that higher power consumption (i.e lower energy gain) help to reduce the PER. However, comparing Tabs. I and II, $\Delta = 4$ does not ensure to have very low values of the PER. This is due to the high variability of the RSSI, caused by the Rayleigh fading. In particular, Fig. 7 shows the temporal evolution of the RSSI in the Rayleigh fading environment, for the MIAD PC with $\Delta = 2$. The high variance of the RSSI causes frequent packet losses, and hence the MIAD PC reacts by increasing the transmit power, as shown in Fig. 8. Specifically, link 1 experiences many packet losses between 100 s and 150 s. This explains the low energy efficiency of link 1 in Tab. II.

By considering both Tabs. I and II, it is reasonable to think that a smaller value of Δ would yield a decrease

Link	$\Delta = 2$		$\Delta = 4$	
	ρ_i	PER %	ρ_i	PER %
1	3.33	9.13	2.91	3.65
2	5.68	7.14	4.55	3.33
3	6.20	5.93	5.09	2.88

TABLE II
GAIN AND PACKET ERROR RATE OF MIAD PC IN A RAYLEIGH ENVIRONMENT FOR TWO DIFFERENT VALUES OF THE STEP SIZE Δ .

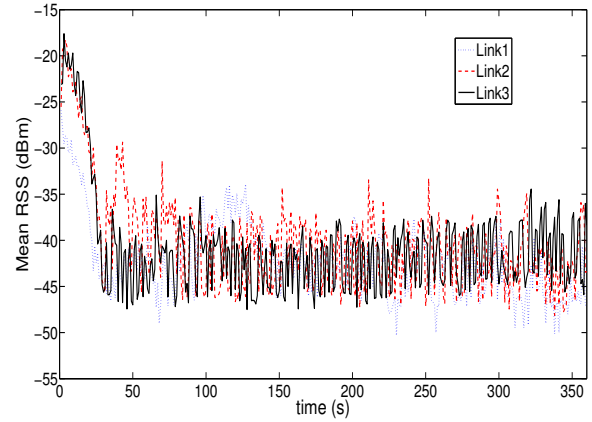


Fig. 7. RSSI for each link as function of time in the Rayleigh environment, MIAD PC with $\Delta = 2$.

of the probability of the high power level states, so that the power gain decreases without degrading communication performance. However, smaller values of Δ provoke more frequent transitions through the lossy states, which demands the necessity to establish a compromise between ρ_i and PER.

In Tab. VII, the energy gain and PER of the PER PC is reported for both the cases of AWGN environment and Rayleigh environment. As first remark, it is interesting to observe that the Rayleigh fading requires more power with respect to the AWGN case (the energy gain is indeed much lower). This is due to the fact that in Rayleigh propagation environments, deep attenuations of the wireless channel coefficients must be compensated with larger transmit power.

In order to asses the MIAD PC with respect to the PER PC, a comparison can be made among I, II, and VII respectively. It is interesting to observe that the PER PC in the AWGN environment obtains very good performance both in term of

Link	AWGN		Rayleigh	
	ρ_i	PER %	ρ_i	PER %
1	7.42	1.06	1.11	7.28
2	10.33	2.17	2.17	6.77
3	10.33	0.45	3.54	7.12

TABLE III
GAIN AND PACKET ERROR RATE OF PER PC IN A AWGN (LEFT) AND RAYLEIGH (RIGHT) ENVIRONMENTS.

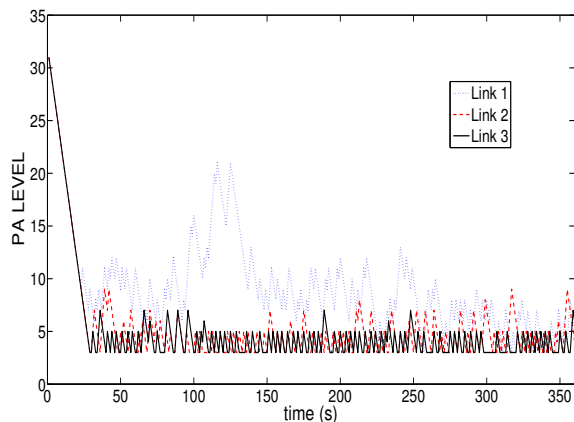


Fig. 8. Power Consumption for each link as function of time in the Rayleigh environment, MIAD PC with $\Delta = 2$.

energy efficiency (which is around 10%) and PER compared to the corresponding cases MIAD PC. This is due to the fact that having a model of the wireless propagation scenario clearly helps to design more effective power levels. Surprisingly, the PER PC shows bad performance if compared to the MIAD PC in the Rayleigh case, both in terms of energy efficiency and in term of packet error rate. The reason of this behavior is found in the fact that the computation of the average SINR, and the corresponding derivation of the power levels using the expression (10), is computationally demanding for the hardware platform, as discussed in section VI. In particular, during the computational operation, the Contiki OS is not able to store more than one packet at a time. Our simulation experiments showed that such a time may be longer than the time between the reception of two consecutive packets. As a consequence, the impossibility to store packets, lead to a bad estimation of the average SINR, with a consequent inaccurate computation of the transmit powers. These problems could be overcome by adding an external processor to the hardware platform, or modifying the operating system in order to enable concurrent execution of computation and packet reception.

VIII. CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented power control algorithms for wireless sensor networks. Specifically, we have described two power control mechanisms: one based on a MIAD mechanism, and another one based on estimated packet loss probability. The latter mechanism is based on a detailed model of the RSSI performance measure provided by the Telos motes. We provided also a middleware description of the power control algorithms using the Contiki operating system. Both the MIAD PC and the PER PC were implemented on Telos motes. The experimental results show that MIAD PC has good performance in terms of energy consumption and packet error rate, and is not computational demanding. PER PC outperforms MIAD PC in an AWGN environment, while in the case of Rayleigh fading, the limitations of the hardware

and software platforms, do not allow to successfully implement the computational demanding PER PC. Ongoing work is focused on the investigation of middleware solutions to simplify the computational complexity of the power control algorithms, so that knowledge of the wireless propagation scenario can be beneficial to the reduction of the power consumption.

IX. ACKNOWLEDGMENTS

The authors acknowledge the support on Contiki by Björn Grönvall.

REFERENCES

- [1] C. Fischione, A. Bonivento, K. H. Johansson, A. Sangiovanni-Vincentelli, "Cooperative diversity with disconnection constraints and sleep discipline for power control in wireless sensor networks," *IEEE VTC Spring 2006*, accepted for publication, May 2006.
- [2] www.moteiv.com/products/docs/telos-revb-datashet.pdf, "Telos: Ultra low power ieee 802.15.4 compliant wireless sensor module," *Moteiv Corporation*, 2004.
- [3] D. Lymberopoulos, Q. Lindsey, A. Savvides, "Characterization of the radio signal strength variability in 3-d ieee 802.15.4 networks using monopole antennas," *EWSN, ETH, Zurich*, 2006.
- [4] X. Liu, A. Goldsmith, "Wireless network design for distributed control," *IEEE Conf. on decision and control*, June 2004.
- [5] M. Sgroi, A. Wolisz, A. Sangiovanni-Vincentelli, J. M. Rabaey, "A service-based universal application interface for ad-hoc wireless sensor networks," *Whitepaper, U.C.Berkeley*, 2004.
- [6] F. Graziosi, F. Santucci and S. Tennina, "Location service design with a platform based approach in ad-hoc wireless sensor networks," *IEEE Globecom*, 2004.
- [7] <http://www.ist-runes.org>, *IP RUNES (Reconfigurable Ubiquitous Networked Embedded System, IST program)*, 2005.
- [8] P. Costa, G. Coulson, C. Mascolo, G. P. Picco, and S. Zachariadis, "The RUNES middleware: A reconfigurable component-based approach to networked embedded systems," *In Proc. of the 16th IEEE PIMRC'05, Berlin, Germany*, 11-14 Sept. 2005.
- [9] K.-E. Årzén, A. Bicchì, G. Dini, S. Hailes, K. H. Johansson, J. Lygeros, and A. Tzes, "A component-based approach to the design of networked control systems," *European Journal of Control*, no. 2–3, 2007, invited paper.
- [10] A. Sampath, P.S. Kumar and J.M. Holtzman, "On setting reverse link target SIR in a CDMA systems," *in Proc. of IEEE VTC*, pp. 929-933, 1997.
- [11] F. Gunnarsson, *Power Control in Wireless Networks - Characteristics and Fundamentals*, ser. M. Guizani, editor, Wireless Communications Systems and Networks. Kluwer Academic Press, 2004.
- [12] M.B. Pursley, "Performance evaluation for phase-coded spread-spectrum multiple-access communication - part i: System analysis," *IEEE transaction on Communications*, Vol. COM. 25, 195–799, 1977.
- [13] F. Santucci, G. Durastante, F. Graziosi, C. Fischione, "Power allocation and control in multimedia cdma wireless systems," *Kluwer Telecommunication Systems*, Vol. 23, pp. 69–94, May–June 2003.
- [14] N. Möller, C. Fischione, K. H. Johansson, F. Santucci, and F. Graziosi, "Modeling and control of ip transport in cellular radio links," *in Proc. of 16 IFAC 2005, Prague, Cze.*, 2005.
- [15] J. G. Proakis, *Digital Communications*, 3rd ed., ser. Electrical Engineering. McGraw Hill International Editions, 1995.
- [16] A. Papoulis, "Probability, Random Variables, and Stochastic Processes". McGraw-Hill, 1984.
- [17] P. Costa, G. Coulson, C. Mascolo, G. P. Picco, and S. Zachariadis, "The RUNES middleware: A reconfigurable component-based approach to networked embedded systems," *in Proc. of the 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, 2005.
- [18] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," *in In Proceedings of the First IEEE Workshop on Embedded Networked Sensors*, 2004.