

# Accuracy-aware Interference Modeling and Measurement in Wireless Sensor Networks

Jun Huang<sup>1\*</sup>; Shucheng Liu<sup>2,4\*</sup>; Guoliang Xing<sup>1</sup>; Hongwei Zhang<sup>3</sup>; Jianping Wang<sup>2</sup>; Liusheng Huang<sup>4</sup>

<sup>1</sup>Michigan State University, USA; <sup>2</sup>City University of Hong Kong, HKSAR;

<sup>3</sup>Wayne State University, USA; <sup>4</sup>University of Science and Technology of China;

**Abstract**—Wireless Sensor Networks (WSNs) are increasingly available for mission-critical applications such as emergency management and health care. To meet the stringent requirements on communication performance, it is crucial to understand the complex wireless interference among sensor nodes. Recent empirical studies suggest that the packet-level interference model, also referred to as the packet reception ratio (PRR) versus SINR model or PRR-SINR model, offers significantly improved realism than other simplistic models such as the disc model. However, as shown in our experimental results, the PRR-SINR model yields considerable spatial and temporal variations in reality, which poses a major challenge for accurate measurement at run time. This paper presents a novel *accuracy-aware* approach to interference modeling and measurement for WSNs. First, we propose a new regression-based PRR-SINR model and analytically characterize its accuracy based on statistics theory. Second, we develop a novel protocol called *accuracy-aware interference measurement* (AIM) for measuring the proposed PRR-SINR model with *assured accuracy* at run time. AIM also adopts new clock calibration and in-network aggregation techniques to reduce the overhead of interference measurement. Our extensive experiments on a 17-node testbed of TelosB motes show that AIM achieves high accuracy of PRR-SINR modeling with significantly lower overhead than state of the art approaches.

## I. INTRODUCTION

Recent years have witnessed the unprecedented proliferation of Wireless Sensor Networks (WSNs) in mission-critical applications such as emergency management, civil infrastructure monitoring, and health care. These applications pose stringent requirements on system communication performance of WSNs. For instance, in the scenario of wireless elderly and patient monitoring, ECG body sensors must report the cardiac rhythm data within bounded delay for real-time diagnosis.

The main factor that limits the communication performance of wireless networks is *interference*. Due to the broadcast medium, wireless transmissions from one node interfere with the reception of surrounding nodes resulting in lower throughput or higher delivery delay. The situation is even worsened for WSNs due to their already limited bandwidth. Understanding interference is thus critical to enable the successful application of WSNs for mission-critical applications. Early work on wireless interference modeling has

widely assumed simple abstract models (e.g., the protocol model [6]). Unfortunately, recent empirical studies on both WSNs [11] [15] and 802.11 mesh networks [13] showed that these models are largely inaccurate. It is suggested that the packet-level physical interference model, also referred to as the packet reception ratio (PRR) versus SINR model or the PRR-SINR model, offers significantly improved realism. Taking advantage of the PRR-SINR model, several recent efforts have been made to improve the performance of link scheduling, topology control, and medium access control (MAC) protocols [5] [11] [14].

Our experimental results based on WSN hardware platforms show that the PRR-SINR model yields significant spatial and temporal variations and hence requires accurate measurement at *run time*. In particular, the packet reception performance of a radio must be carefully profiled under different SINRs within the *transitional region* where the PRR varies from zero to 100%. Recently, several interference measurement methods have been proposed [9] [10] [12] [13]. However, they either require nodes to transmit/receive a large number of measurement packets or must sample extensive statistics of data packets, which leads to high overhead for WSNs. For instance, the methods proposed in [11] [13] [15] need to be periodically seeded by at least  $O(N)$  trials in an  $N$ -node network where each node transmits a large number of measurement packets in turn. Moreover, none of the existing methods is designed to measure interference models with *assured accuracy*. As a result, the errors of these models may cause the upper-layer protocols (e.g., link scheduling MAC protocols [3] [11] [14]) built upon them to yield unpredictable performance.

In this paper, we propose a novel *accuracy-aware* approach to interference modeling and measurement for WSNs. A key advantage of our approach is that it analytically characterizes the accuracy of the PRR-SINR model, and enforces a specified accuracy bound by adaptively controlling the measurement process at run time. As a result, we can achieve the desired accuracy of interference measurement by sampling “just enough” statistics of data packets. Our major contributions are summarized as follows.

- 1) We conduct extensive measurement of the PRR-SINR model on TelosB motes. Our results show that the PRR-SINR model yields significant spatial and tem-

\*Corresponding authors.

poral variations. In particular, the PRR of a node may vary up to 50% under the same SINR, which demonstrates the necessity of accurate online measurement. To the best of our knowledge, this is the first systematic experimental characterization of spatiotemporal behavior of the PRR-SINR model in WSNs.

- 2) We propose a regression based model to characterize the relation between PRR and SINR of WSN nodes. By linearly transforming the theoretical PRR-SINR model, our model significantly simplifies the complexity of experimental measurement. We then apply statistical techniques to rigorously derive the accuracy bound of our model. Our analysis correlates the interference modeling error with key online measurement parameters including the selection of SINR values and size of samples.
- 3) We develop the *accuracy-aware interference measurement* (AIM) protocol for measuring the proposed PRR-SINR model at run time. AIM consists of several novel mechanisms to minimize the overhead of model measurement: lightweight clock calibration to reduce the overhead of synchronous timestamping, localized interference statistic aggregation, and accuracy control that adaptively stops the measurement process once the desired modeling accuracy is achieved.
- 4) We implemented AIM in TinyOS-2.0.2 and conducted extensive experiments on a 17-node testbed of TelosB motes. Our experimental results show that AIM can achieve high accuracy of PRR-SINR modeling with significantly lower overhead compared with the active measurement approach.

The rest of the paper is organized as follows. Section II reviews related work. Section III describes the background and empirical observation of interference models. Section IV discusses modeling accuracy control. Section V presents the design of AIM. Section VI presents experimental results and Section VII concludes the paper.

## II. RELATED WORK

Early work [16] showed that the communication links of WSNs are often lossy and asymmetric. Several recent studies are focused on experimental characterization of the PRR-SINR model. Son et al. [15] studied the PRR-SINR model of CC1000 radios. In [11], several interference models including the PRR-SINR model, disc model, and the thresholded physical interference model, are studied for their accuracies. Interference and packet delivery models are proposed by Reis et al. [13], which can be instantiated by packet transmission traces. Qiu et al. [12] proposed a general interference model to characterize the interference among arbitrary number of 802.11 senders. In [9], a measurement-based approach is proposed to model the interference and link capacity in 802.11 networks. All the above studies employed the active approach to measure the interference

model, which requires nodes to periodically transmit/receive extensive measurement beacons.

Recently, we propose a protocol called passive interference measurement (PIM) [10]. PIM samples the statistics of timestamps and received signal strength (RSS) of data packets and use them to derive the nodes' interference relationship and build their PRR-SINR models. Without generating any measurement packets, PIM reduces the overhead of active approaches [11] [13] [15]. However, it suffers from the following issues. First, PIM adopts a centralized tree-based statistics collection algorithm which incurs high overhead in large networks. Second, PIM requires frequent time synchronization and constant overhearing, which leads to unnecessary energy consumption of idle listening. Third, like all existing solutions, PIM has no accuracy control over the measured interference model. As a result, it may over/under-sample the data traffic leading to long measurement delay, high messaging overhead, or poor measurement accuracy.

## III. UNDERSTANDING THE PRR-SINR MODEL

In this section, we first provide the theoretical background on the PRR-SINR model. We then present several findings from real testbed experiments. Finally, we present a regression-based PRR-SINR model.

### A. Background on the Theoretical PRR-SINR Model

According to communication theory, the bit error rate (BER), i.e., the probability that a transceiver failed to receive an incoming bit is governed by the physical SINR model. Unfortunately, the BER-SINR model cannot be directly measured on commodity radio transceivers [13]. As a result, most recent empirical studies [9] [12] [13] [14] [15] have adopted a *packet-level* interference model that correlates packet reception ratio (PRR) with SINR. This model is also referred to as the PRR-SINR model. Specifically, PRR is the probability that the transceiver  $r$  successfully receives all bits in an incoming packet, which is given by

$$p_r(\omega) = b_r(\omega)^{\lambda(\omega)} \quad (1)$$

where  $b_r(\omega)$  is the probability that  $r$  receives an incoming bit of packet  $\omega$  whose size is  $\lambda(\omega)$ . According to communication theory,  $b_r$  is determined by  $E_b$  and  $N$ , where  $E_b$  is the energy of the signal, and  $N/2$  is the two-sided power spectral density of the noise. For 802.15.4 radios, the bit reception ratio is given in the following formula:

$$b_r(\omega) = 1 - Q\left(\sqrt{\frac{2E_b}{N}}\right) \quad (2)$$

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt = \frac{1}{2}(1 - \text{erf}(x/\sqrt{2})) \quad (3)$$

where  $\text{erf}(\cdot)$  is the Gauss error function, Let  $\varepsilon$  be the SINR at the receiver of  $\omega$ ,  $\varepsilon$  is given by

$$\varepsilon = \frac{E_b}{N} \frac{R}{B_N} \quad (4)$$

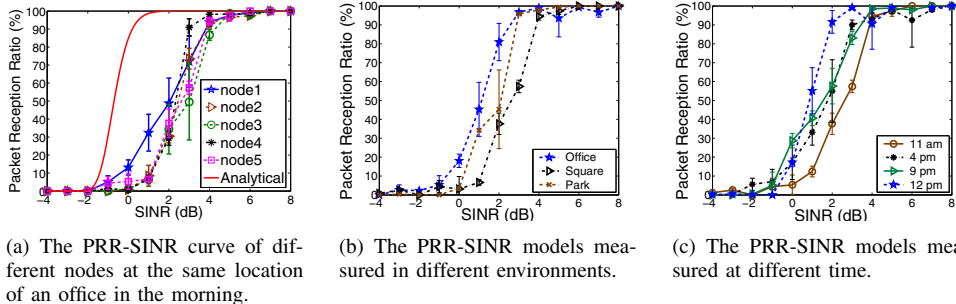


Figure 1. The measured PRR-SINR model at different time in different environments.

where  $R$  is the modulation rate, and  $B_N$  is the noise bandwidth. From Eq. (1) to Eq. (4), we have

$$p_r(\omega) = \left(\frac{1}{2} + \frac{1}{2} \times \text{erf}\left(\sqrt{\frac{B_N \cdot \varepsilon}{R}}\right)\right)^{\lambda(\omega)} \quad (5)$$

### B. Empirical Observation

In this subsection, we provide an empirical study on the PRR-SINR model. Our objective is two-fold. First, we compare the real-world PRR-SINR measurements with the theoretical PRR-SINR model discussed in Section III-A. Second, we investigate the PRR-SINR relationship under different spatial and temporal settings. We show that the model yields significant spatial and temporal variation, which poses a major challenge for interference measurement and modeling on low-power wireless sensor nodes.

Our experiments are conducted on TelosB motes equipped with 802.15.4-compliant CC2420 radios. The PRR-SINR curves are measured using an existing method [11] [14] [15]. Specifically, a large number of packets are sent by two interfering links with different transmission power levels, and the SINR and PRR statistics of these transmissions are collected at the receiver to generate the model. The SINR of a packet reception is computed using the received signal strength (RSS) of the incoming packets, interfering packets, and the noise power. The RSS of a packet can be obtained by reading the RSS indicator register of the radio. The noise power can also be obtained from the RSS indicator when there is no packet transmission. We note that the SINR is always integers as the RSS precision level of CC2420 radio is 1 dBm. More details about the measurement method can be found in [11] [14] [15]. The experiments last about 12 hours and are conducted in three different environments: an office, the square between two academic buildings, and a small park. Each data point (PRR, SINR) is measured with 12000 packets of 128 bytes and a confidence interval of 90% is computed. Fig. 1(a) depicts the PRR-SINR model measured on different nodes in an office in the morning. Fig. 1(b) and 1(c) show the model on the same node measured at different time in different environments. From the results, we have the following observations.

First, we observe that each PRR-SINR curve has a *transitional region* of about 10 dB wide, in which the PRR grows

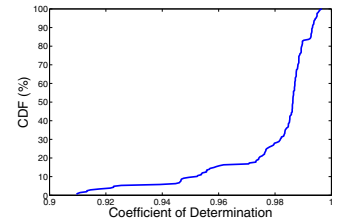


Figure 2. The CDF of coefficient of determinations of 208 regression-based PRR-SINR models

from zero to one. This result is consistent with the findings reported by recent empirical studies [14] except the slight variation in the width of the transitional region.

Second, we observe that the PRR-SINR model is significantly influenced by the spatial and temporal factors. Fig. 1(a) shows that the PRR under the same SINR may vary as much as 50%, when the model is measured on different nodes. From Fig. 1(b) and Fig. 1(c), we also observe significant variations when the model is measured at different time, or the node is placed in different environments. For example, when the SINR is 2 dB at the receiver, the PRR measured in office is 45% higher than the PRR in the square; and the PRR measured at 11am is 55% lower than 12pm.

Third, the theoretical model given in Eq. (5) fails to capture the significant temporal and spatial variations observed in reality. For instance, Fig. 1(a) shows that under the same SINR, the difference between theoretical and measured model can be as much as 85%. This is partially due to the fact that the theoretical model is instantiated by the parameters defined on ideal channel, which do not account for the diversity of environments. Although it is possible to improve the accuracy of the theoretical model by carefully measuring some parameters (e.g., the bandwidth and power spectral density of noise) in a particular environment, doing so is time-consuming and particularly difficult at run time. We also note that the bit-level SINR required by the theoretical model cannot be measured on commodity radios. The results in Fig. 1 clearly demonstrate the need of measuring the PRR-SINR model in an online manner at run time.

### C. The PRR-SINR Regression Model

As discussed in the previous subsection, the accuracy of the theoretical PRR-SINR model is poor due to the spatial and temporal variation of PRR-SINR relation in reality. To obtain high accuracy, we may directly measure the PRR-SINR relation for every SINR point to construct a discrete PRR-SINR model, as shown in Fig. 1. However, all (PRR, SINR) pairs in the transitional region of every node must be continuously measured due to the spatiotemporal variation. Therefore, such an approach poses significant message overhead.

In this paper, we propose to build a regression model for characterizing the PRR-SINR relation. Our approach has

two major advantages. First, the measurement overhead is significantly reduced because only a subset of (PRR, SINR) samples in the transitional region are needed to build the model. Second, the model allows us to leverage the existing theory in statistics to formally characterize the modeling accuracy and hence provide key guidance on minimizing the overhead in interference measurement.

The regression model is built as follows. First, to approximate the actual PRR-SINR model, we scale and shift the theoretical model by transforming the predictor variable  $x$  through a linear transformation as follows:

$$x' = \alpha_1 x + \alpha_2.$$

According to the theoretical model given in Eq. (5), we can use the following parameterized model to characterize the PRR-SINR relation in practice:

$$y = \left(\frac{1}{2} + \frac{1}{2} \times \operatorname{erf}\left(\sqrt{\frac{B_N(\alpha_1 x + \alpha_2)}{R}}\right)\right)^\lambda, \quad (6)$$

where  $\lambda$  is the packet length,  $\alpha_1$  and  $\alpha_2$  are the parameters to be estimated in real-world settings. Since the  $(x, y)$  relation as described by Eq. (6) is non-linear, we cannot directly apply the model in regression analysis. To address this issue, we first approximate the function  $\operatorname{erf}(\cdot)$  according to an existing formula Eq. (7), so that we can derive closed-form solution to the  $(x, y)$  relation:

$$\operatorname{erf}(x) \approx \sqrt{1 - e^{-(2x/\sqrt{\pi})^2}} \quad (7)$$

Then, we derive the following based on Eq. (6):

$$-\ln(1 - (2y^{\frac{1}{\lambda}} - 1)^2) \frac{\pi R}{4B_N} = \alpha_1 x + \alpha_2. \quad (8)$$

Now, we let  $y'$  to represent the left side of Eq. (8),

$$y' = -\ln(1 - (2y^{\frac{1}{\lambda}} - 1)^2) \frac{\pi R}{4B_N}, \quad (9)$$

Equation 8 is now transformed into a linear model,

$$y' = \alpha_1 x + \alpha_2. \quad (10)$$

Based on this linear model, we can conduct regression analysis to derive  $\alpha_1$  and  $\alpha_2$  using on-line measurements. That is, for each sample point  $(x, y)$  where  $x$  is a SINR value and  $y$  is the corresponding PRR, we can derive a transformed sample point  $(x, y')$  using Eq. (9); using the set of transformed sample points and Eq. (10), we can derive the values for parameters  $\alpha_1$  and  $\alpha_2$  through regression analysis; finally, plugging the derived  $\alpha_1$  and  $\alpha_2$  back into Eq. (6) gives the PRR-SINR model.

We now evaluate the accuracy of the above regression-based model. First, Fig. 3 shows the  $(x, y)$  and  $(x, y')$  relations without and with the transformation (as shown in Eq. (9)) for three sets of measurements. The x-axis in Fig. 3(a) is the SINR in the unit of ratio of received powers. Note that it is not converted to decibel. Fig. 3(b) shows

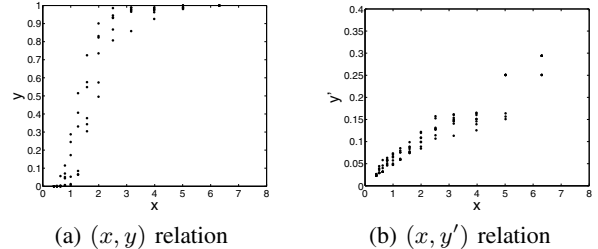


Figure 3. The original and linearly transformed PRR-SINR relations.

that, after transformation, the values of  $y'$  are highly linear with respect to  $x$  for each set of PRR-SINR measurement, which demonstrates the validity of our linear transformation. Second, we evaluate the error of regression with respect to real measurement based on a commonly used metric called the Coefficient of Determination (CoD). When CoD approaches to one, it indicates higher accuracy of regression. Fig. 2 shows the CDF of CoD of 208 sets of PRR-SINR curves measured in the experiments described earlier. We can see that more than 80% of CoD lies above 97%, which shows the accuracy of our regression-based modeling.

#### IV. MODEL-BASED ACCURACY CONTROL

In this section, we analyze the accuracy of regression-based PRR-SINR model and the overhead of measuring it at run time. Given the significant temporal variation of the PRR-SINR relation shown in Section III-B, the PRR-SINR model needs to be updated via regression of runtime measurements to obtain high accuracy. Therefore, it is critical to understand the impact of measurement overhead on the model's accuracy. There are two major factors that affect the overhead in the PRR-SINR regression modeling: 1) the number of SINR points for which we collect sample PRR values and 2) the number of PRR samples we collect for each individual SINR point. Besides affecting modeling overhead, these two factors directly affect the accuracy of regression modeling. Therefore, our approach is to control the overhead in regression modeling while achieving the desired modeling accuracy. Our approach consists of two key techniques: *principal SINR selection* and *sampling size selection*, which are discussed in the following.

Given the parameterized theoretical model Eq. (6) and its transformed linear model Eq. (10), we first identify a set of *principal SINR* points for which we will measure the corresponding PRRs. We refer to the SINR points whose PRRs will not be measured the *secondary SINR* points; the PRR corresponding to a secondary SINR point is predicted using the regression model Eq. (10), where the regression model is built based on measurement data for the principal SINR points. Based on the selected principal SINR points, we then decide the number of PRR samples to collect for each principal SINR point according to the requirements on the measurement accuracy of the principal SINR point and the prediction accuracy of the secondary SINR points. In what follows, we elaborate on our method of selecting

principal SINR points and determining the sample size for each principal SINR point. These results will be used in Section V to develop the accuracy-aware interference measurement (AIM) protocol.

#### A. Selection of Principal SINR Points

The objective of choosing the principal SINR points is such that we can achieve the desired regression modeling accuracy while using the minimum number of principal SINR points. The regression modeling accuracy can be characterized by the mean-squared-errors (MSE) in linear regression [8]. Thus the goal is to choose the minimum number of principal SINR points to satisfy the maximum tolerable MSE. Given that the total number of integer SINR points is not very large (e.g., about 10 for a typical transitional region), we perform exhaustive search to identify the minimum set of principal SINR points, and we have found this approach to be effective in practice and affordable for WSN platforms. To identify the principal SINR points, we need to take a small number of measurements for every SINR point (including the secondary SINR points); once the principal SINR points have been identified, they will be used to build the PRR-SINR model until the accuracy of the model exceeds the required bound due to the temporal variation. We will discuss more details about the implementation of principal SINR points selection in Section V-D.

#### B. Sample Size Selection for Principal SINR Points

Given a selected set of principal SINR points, the overall accuracy of the regression-based modeling can be ensured by controlling the measurement accuracy of the PRRs corresponding to the principal SINR points and by controlling the prediction accuracy for the PRRs corresponding to the secondary SINR points. To minimize the overhead in the regression analysis, we need to control the sampling process to minimize the number of samples taken while ensuring the required measurement accuracy and prediction accuracy. For each principal SINR point  $x_i$ , we first compute the minimum number of PRR samples, denoted by  $m_{x_i}$ , that we need to ensure the measurement accuracy for the PRR corresponding to  $x_i$ ; then we compute the minimum number of PRR samples, denoted by  $n_{x_i}$ , that we need for each principal SINR point to ensure the required prediction accuracy for the PRRs corresponding to secondary SINR points; finally, we compute the number of PRR samples for a principal SINR point  $x_i$  as  $\max\{m_{x_i}, n_{x_i}\}$ . In what follows, we elaborate on our method of sample size computation for ensuring the required measurement accuracy and prediction accuracy.

##### 1) Sample size for ensuring measurement accuracy:

Assume that we need to control the measurement error to be within  $r_m\%$  of the mean PRR at the  $100(1-\alpha)\%$  confidence level for a principal SINR point  $x_i$ , the measured mean PRR is  $y_i$ , and the minimum number of samples to ensure the required accuracy is  $m$ . Note that, in this case, each

sample reflects the status (i.e., success or failure) of a packet transmission whose corresponding receiver-size SINR is  $x_i$ .

Given a set of  $m$  samples, the  $100(1-\alpha)\%$  confidence interval for the mean PRR is

$$y_i \mp z_{1-\alpha/2} \sqrt{\frac{y_i(1-y_i)}{m}}, \quad (11)$$

where  $z_{1-\alpha/2}$  is the  $(1-\alpha/2)$ -quantile of the standard normal variate [8]. To satisfy the required accuracy, the following should hold

$$z_{1-\alpha/2} \sqrt{\frac{y_i(1-y_i)}{m}} \leq y_i \frac{r_m}{100}. \quad (12)$$

From (12), we have the following on the required sample size  $m$ :

$$m \geq \frac{10000 z_{1-\alpha/2}^2 y_i (1-y_i)}{r_m^2}. \quad (13)$$

Therefore,  $\frac{10000 z_{1-\alpha/2}^2 y_i (1-y_i)}{r_m^2}$  samples of packet transmission status is enough to ensure the required accuracy of  $(100-r_m)\%$  for the SINR point  $x_i$ .

2) *Sample size for ensuring prediction accuracy:* Given  $n$  pairs of measurement data on  $(x_i, y_i)$  ( $i = 1 \dots n$ ), where  $x_i$  is a principal SINR point, and  $y_i$  is a sample of the corresponding PRR, we can derive the regression model  $y' = \alpha_1 x + \alpha_2$  (see Eq. 10) with the corresponding standard deviation of errors  $s_e = \sqrt{\frac{SSE}{n-2}}$ . When predicting  $y'_j$  for an secondary SINR point  $x_j$ , the mean value of the predicted  $y'_j$  is

$$\hat{y}'_j = \alpha_1 x_j + \alpha_2, \quad (14)$$

and the standard deviation of  $y'_j$  is

$$s_{\hat{y}'_j} = s_e \left[ \frac{1}{n} + \frac{(x_j - \bar{x})^2}{\sum_{i=1}^n x_i^2 - n\bar{x}^2} \right]^{1/2} \quad (15)$$

where  $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$  [8]. Then the  $100(1-\alpha)\%$  confidence interval for  $y'_j$  is  $\hat{y}'_j \mp s_{\hat{y}'_j} t_{[1-\alpha/2, n-2]}$ , where  $t_{[1-\alpha/2, n-2]}$  is the  $(1-\alpha/2)$ -quantile of a t-variate with  $n-2$  degrees of freedom. Assume that the prediction error is required to be within  $r_p\%$  of the mean value at the  $100(1-\alpha)\%$  confidence level, then the following should hold

$$s_{\hat{y}'_j} t_{[1-\alpha/2, n-1]} \leq \hat{y}'_j \frac{r_p}{100}. \quad (16)$$

From (16) and (15), we have the following on the required sample size  $n$ :

$$n \geq \frac{2x_j \sum_{i=1}^n x_i - Y(\sum_{i=1}^n x_i)^2 - \sum_{i=1}^n x_i^2}{x_j^2 - Y \sum_{i=1}^n x_i^2}, \quad (17)$$

where  $Y = \frac{\hat{y}'_j^2 r_p^2}{10000 s_e^2 t_{[1-\alpha/2, n-2]}^2}$ . Let

$$n_{x_j} = \frac{2x_j \sum_{i=1}^n x_i - Y(\sum_{i=1}^n x_i)^2 - \sum_{i=1}^n x_i^2}{x_j^2 - Y \sum_{i=1}^n x_i^2}.$$

Then the minimum required sample size

$$n = \max_{x_j \in X_a} n_{x_j} \quad (18)$$

where  $X_a$  is the set of secondary SINR points whose corresponding PRRs need to be predicted. In implementation, we can evenly distribute the  $n$  samples to the  $K$  principal SINR points, and we can take  $\lceil \frac{n}{K} \rceil$  pairs of  $(x_i, y_i)$  sample data for each principal SINR point  $x_i$ .

## V. THE DESIGN OF AIM

This section presents the design of AIM. We first give an overview of our approach, and then discuss each component in details.

### A. Overview

AIM is a novel protocol for accurately measuring the PRR-SINR model with minimum overhead. The system architecture of AIM is illustrated in Fig. 4. AIM consists of several novel mechanisms. (1) *Interference statistics collection*. AIM collects interference statistics from normal network traffic, and piggybacks the statistics in data packets that are forwarded to the base station of the network. For each node whose interference model is to be measured, an aggregator is chosen on the routing tree to generate the PRR-SINR model using collected interference statistics. (2) *Regression-based model generation*. With collected statistics, each aggregator infers the packet-level interference, correlates PRR and SINR values, and generates the regression-based PRR-SINR model in real time. (3) *Accuracy control*. AIM controls the accuracy of measured models to a specified bound based on the rigorous error analysis described in Section V-D, while minimizing the overall measurement overhead. AIM also tracks the temporal variations of measured models and dynamically adjusts measurement parameters like principal SINR values. (4) *Lightweight clock calibration*. AIM calibrates local clocks on each nodes for accurate interference detection. The clock calibration mechanism is developed based on the empirical observation of clock drift on real sensor hardware.

### B. Interference Statistics Collection

We now discuss how AIM collects data packet statistics that are needed to build the regression PRR-SINR model. We assume that network has a tree-based topology in which all nodes send their data to the base station. Tree-based topologies have been commonly used in WSN applications. The design of AIM is not dependent on any particular MAC, although it relies on collecting the statistics of packet events (transmissions, receptions and collisions) for generating PRR-SINR models. We note that significant contention and packet collisions may exist even when a network adopts interference mitigation techniques such as CSMA and TDMA. In particular, the TDMA schedule constructed based on simplistic interference models often cannot effectively

avoid packet collisions because these models do not account for the spatiotemporal dynamics of interference.

In contrast to most existing methods that rely on extensive measurement packets [14] [15], AIM adopts a passive scheme [10] that infers the PRR-SINR model purely from the statistics of data packets. We now briefly discuss the basic idea of passive interference measurement. We denote the nodes whose PRR-SINR models are to be measured as *m-nodes*. For a given m-node, a set of *reference nodes*, referred as *r-nodes*, is selected to help the measurement of the PRR-SINR model. The transmission of *r-nodes* must interfere with the packet reception of m-node, so that the transitional region of the PRR-SINR model can be fully captured. The selection of r-nodes can be optimized using an existing algorithm [10]. The m-nodes/r-nodes timestamp packet reception/transmission events. The precise timestamping is crucial, since the timestamps recorded by different nodes are correlated to infer the packet interference at m-nodes. Each m-node works in promiscuous mode, and records the RSS values and reception/loss statistics of overheard packets. The statistics are then piggybacked in data packets, and transmitted to the base station of the network. Along the routing tree, an *aggregator* node is selected for each m-node for PRR-SINR model generation. With the collected timestamps and RSS statistics, the aggregator can infer the SINR of each packet reception or loss for the m-node. Combined with the packet reception statistics, the PRR-SINR model of the m-node can be accurately generated. The aggregator can send the parameters of generated models to the m-node or base station to support the performance optimization of upper layer protocols.

Due to the piggybacking of interference statistics in data packets, AIM induces additional overhead in multi-hop forwarding. To address this problem, AIM employs an efficient distributed aggregator selection algorithm. Let  $T_u$  be the minimum sub-routing tree that includes an m-node  $u$  and all the r-nodes of  $u$ . The root of  $T_u$  should be selected as the aggregator, as only the interference statistics collected by the m-node and r-nodes will be needed for model generation. The details are omitted here due to space limitation and can be found in [7]. After the aggregator is determined, it starts model generation for m-node  $u$  using the collected interference statistics.

### C. Regression based Model Generation

An aggregator in AIM generates the PRR-SINR model by applying regression on a set of (PRR,SINR) samples. That is, the parameters of the linear model Eq. (10) are computed using curve fitting. We now discuss how an aggregator computes (PRR,SINR) samples for an m-node using collected statistics. By analyzing the timing information, the aggregator can infer that a collision occurred at m-node if the air-time of r-nodes' transmissions overlap with each other. We note that this condition is necessary but not sufficient

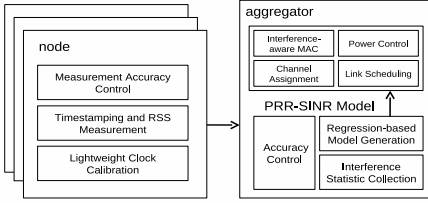


Figure 4. Architecture of the AIM protocol.

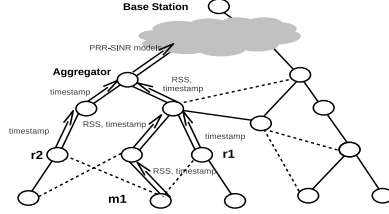


Figure 5. An example of network topology where node  $m_1$ 's model is generated at the aggregator.

timestamp(P12) = timestamp(P22) : collision but received  
 SINR (P12) = RSS(P11) - RSS(P21) - RSS(noise) = 2 dB

node id	Packet id	Tx/Rx	Timestamp (ms)	RSS (dBm)
m1	P11	Rx	0.052	-55
m1	P21	Rx	5.058	-57
m1	P12	Rx	10.112	-51
r1	P11	Tx	0.038	-55
r1	P12	Tx	10.038	-55
r2	P21	Tx	5.045	-57
r2	P22	Tx	10.045	-57

Figure 6. The collected statistics at aggregator.

for a packet collision because two non-interfering nodes may transmit simultaneously. AIM employs an interferer detection algorithm [10] to avoid such false positives. The aggregator then derives the resulting SINR using RSS measurements. Assuming that packet  $p_{u,v}$  transmitted by r-node  $u$  is received by m-node  $v$ . The SINR of  $p_{u,v}$ , denoted as  $\varepsilon_{\eta_{u,v}}$  can be calculated as

$$\varepsilon_{\eta_{u,v}} = \frac{RSS(u,v)}{\sum_{x \in J} RSS(x,v) + \bar{I}} \quad (19)$$

where  $RSS(u,v)$  is RSS of the packet transmitted from  $u$  to  $v$ , and  $\bar{I}$  is the noise power. And  $J$  is the set of r-nodes whose transmissions collide with  $\eta_{u,v}$ . The interference caused packet loss can be detected, if packet transmissions of r-nodes are overlapped in time, but the receiver does not record any packet receptions. Finally the aggregator uses the statistics collected at the m-node to compute the reception ratio of all packets that have the same SINR, and finally generates a set of (PRR, SINR) samples for each m-node.

We now use a simple example to illustrate the basic idea of generating a (PRR, SINR) sample. Fig. 5 shows the topology of the network. Communication and interference links are marked as solid and dashed lines, respectively. Two r-nodes,  $r_1$  and  $r_2$  are selected to measure the PRR-SINR model of m-node  $m_1$ . The m-node and r-nodes are required to timestamp the transmissions/receptions. In addition,  $m_1$  measures the RSS for each overheard packet. The collected statistics, which are given in Fig. 6, are then forwarded to the aggregator. By analyzing the timestamps, the aggregator infers that packets  $p_{12}$  and  $p_{22}$  collided at  $m_1$ , as they have close timestamps. To compute the SINR, the aggregator uses the measured RSS of non-interfered packets, i.e.,  $p_{11}$  and  $p_{21}$ , to infer the signal power and interference power. In this example, the resulted SINR will be 2 dB. This example illustrates how a single (PRR, SINR) sample is computed. A similar process can obtain a set of measured (PRR, SINR) pairs used to generate the regression based model.

#### D. Accuracy Control

The accuracy of the PRR-SINR model is crucial for the performance of upper-layer protocols. We now discuss how AIM implements the accuracy control scheme described in Section IV. Specifically, AIM employs the following three mechanisms for ensuring an accuracy upper bound specified

by users: principal SINR identification, adaptive sampling, and temporal variation correction.

First, we identify the principal SINR points in the transitional region of the PRR-SINR model for regression modeling. To begin with, we take a small number of PRR measurements for every SINR points during the initialization phase of AIM. Then we conduct an optimal exhaustive search to find the required number of principal points. The objective is to use the minimum principal points, while achieving the required modeling accuracy, quantified by the mean-squared-errors (MSE) in linear regression. For each m-node whose model needs to be measured, a principal SINR set is found and then stored on aggregator. Our experiments show that a small number of principal SINR points are enough to achieve satisfactory accuracy (see Section VI).

Second, at run time, an adaptive sampling algorithm is run by each aggregator to count the number of (PRR, SINR) samples received so far, and notifies the m/r-nodes to stop measurement once the desired number of samples is received. As discussed in Section IV, the overall accuracy of the regression model is decided by 1) the measurement accuracy of PRRs at each principal SINR points, 2) and the prediction accuracy at each secondary SINR points. Specifically, in the accuracy control component of AIM, we use Eq. (13) to control the measurement accuracy, and Eq. (18) for prediction accuracy. The aggregator will notify its m/r-nodes once enough samples have been collected for model generation. Then the m/r-node will stop local measurement and statistic sampling, if they are not involved in model generation of other m-nodes.

The third mechanism used by AIM for accuracy control copes with the temporal variation of the PRR-SINR model. As shown in Section III-B, the PRR-SINR model may vary with time considerably, which leads to accuracy degradation of measured models. AIM deals with this issue by two solutions: (1) An aggregator tracks the starting and ending SINR points of the measured transitional region and regenerates the principal SINR points once the transitional region has shifted. In such a case, the original principal SINR points no longer guarantee the accuracy of the measured model. As shown by our empirical results in Section III-B, the shape of PRR-SINR curve remains relatively stable over time. Therefore, a new set of principal SINR points



can be generated based on the offline-measured PRR-SINR model. (2) When the shape of the fitted PRR-SINR curve varies substantially over time (e.g., due to the environmental dynamics), the aggregator asks the m/r-nodes to measure all (PRR, SINR) pairs in the transitional region, and then generates a new set of principal SINR points. Note that an aggregator incurs higher storage overhead in such a case due to the need of buffering more statistics for computing the discrete PRR-SINR model. However, once the new set of principal SINR points is generated, it resumes to normal operation in which only the PRRs corresponding to principal SINR points need to be computed.

### E. Lightweight Clock Calibration

In AIM, the precise timestamping is crucial, since the timestamps recorded by different nodes are correlated to infer the packet interference at m-nodes. Specifically, two packets are deemed to collide at the receiver if the difference between their timestamps is within a threshold  $\Delta$ . We set the threshold in our implementation to 1 ms. For the TelosB platform, the time of transmitting a packet of size 100 bytes is about 4 ms. In order to correctly infer the packet interference at m-node, AIM synchronizes the local clocks on m-nodes and r-nodes for accurate timestamping by running a network-wide time synchronization protocol (e.g., TPSN [4]) and a novel local clock calibration algorithm. Although a network-wide time synchronization protocol can achieve high timing accuracy, it must be run frequently due to the clock shifts of nodes. AIM reduces the overhead by running the time synchronization protocol infrequently while allowing each node to calibrate its clock locally between two consecutive synchronizations.

The design of our algorithm is motivated by the characteristics of clock drift of sensor hardware. Fig. 7 plots clock drifts of 3 TelosB motes during a 9-hour experiment. Two key observations can be made. First, the clock drift differs on different nodes. Second, the clock drift increases nearly linearly with time. To the best of our knowledge, such characteristics have not been explicitly exploited by existing time synchronization protocols. The clock calibration algorithm of AIM consists of two steps. First, the clock drift is measured at each node, when a network-wide time synchronization is performed. Second, the timestamps of m-nodes and r-nodes are calibrated locally according to their drift characteristics. The details are omitted here due to space limitation and can be found in [7]. We conducted an experiment of 12 hours to evaluate the performance of our clock calibration algorithm. We set the period of network-wide time synchronization to 1 hour, which only incurs low control overhead. During the time interval between consecutive synchronizations, we use our clock calibration algorithm to calibrate the clocks of nodes. Fig. 8 shows the drift errors of 8 nodes over the period of 12 hours. It can be seen that the error is within 1 ms most of the time.

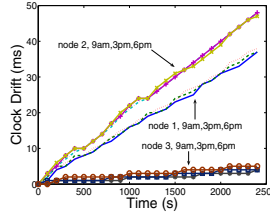


Figure 7. Empirical observation on clock drifts vs time.

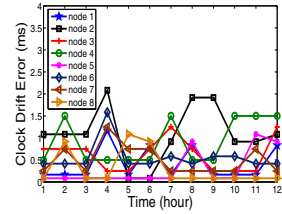


Figure 8. Error of time synchronization.

As a packet transmission takes several milliseconds, such a granularity is accurate enough for the aggregator to infer the packet-level interference correctly.

## VI. EXPERIMENTATION

We experimentally analyze the performance of AIM, and compare AIM with other state-of-the-art interference measurement methods. We first discuss the experimental methodology and then the experimental results.

### A. Experimental Methodology

We have implemented AIM in TinyOS-2.0.2. We use a testbed of 17 TelosB motes [1] for the evaluation. The motes are organized such that 16 of them generate data packets that will be delivered to the 17th mote which serves as the sink. The interference models of all the nodes except the sink are measured. As discussed in Section V, an aggregator node is chosen for measuring the PRR-SINR model of each node. Note that a m-node may serve as the r-node and/or aggregator for other m-nodes. The collection-tree-protocol (CTP) [2] is adopted as the routing protocol. The 16 source nodes are deployed in a  $4 \times 4$  grid in an office, with the distance between two closest grid-points being 10 feet. Each source continuously generates packets of 128 bytes long at an average frequency of 10 packets per second. We tune transmission power levels of nodes to generate different multi-hop topologies up to 4 hops. We conduct an experiment of 2 hours for each topology.

In the following, we first evaluate the performance of accuracy control in AIM, then comparatively study AIM with respect to two baseline interference measurement methods: 1) ACTIVE: a representative *active* method [14] [15] which generates the PRR-SINR models using measurement packets. The number of measurement packets is varied in different experiments; 2) PIM [10]: a state-of-the-art *passive* method which builds the PRR-SINR model using interference statistics collected in normal network traffic. In contrast to AIM, both methods are designed with no accuracy control over the generated models.

### B. Performance of Accuracy Control

We now evaluate the performance of accuracy control, which includes the control of measurement accuracy, regression accuracy and prediction accuracy. The (PRR, SINR) measurements were collected from aggregators and then



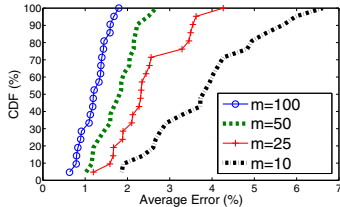


Figure 9. The CDF of average measurement errors

used to generate PRR-SINR models according to various settings. At the same time, the active method [11] [14] [15] is periodically executed to measure the PRR-SINR models. Each model is measured with about 8000 measurement packets. It was shown [11] that such an approach can achieve high accuracy at the price of high measurement overhead. These models are used as “ground truth” to evaluate the accuracy of the models generated by AIM.

We first analyze measurement accuracy control, whose purpose is to ensure the accuracy of measuring PRR for each single principle SINR value. We generate a different PRR-SINR model using every  $m$  samples where each sample is the status (i.e., success or failure) of a packet transmission. Fig. 9 shows the CDF of average measurement errors of the principal points of all models of 16 nodes. It can be seen that the number of transmission status samples required is small. For instance, the average error is less than 7% even with a sample size of 10. As the number of samples collected increases, the measurement error further decreases. For instance, with 100 samples, the error is usually less than 1.5%. From this, we see that accuracy-aware adaptive sampling can save significant sampling overhead since it only takes very few samples in general.

The regression accuracy depends on the number of principal SINR points and the principal selection strategy. Fig. 10 shows the impact of the number of principal-SINR-points on the accuracy of the regression modeling, which is measured by the mean-squared-errors (MSE) in linear regression as defined in Section V-D. The MSE quantifies the error, when the regression PRR-SINR model generated from principal SINR points is used to predict the PRRs of secondary SINR points. We see that a small number of principal points is enough to ensure small MSE. For instance, the MSE is less than 10% even if we only use 3 principal points. As the number of principal points increases, the MSE quickly decreases. For instance, the MSE is less than 3% when the number of principal points is 8. Fig. 11 shows the CDF of the average prediction errors for different number of principal points. We see that, similar to the regression error MSE, the prediction error tends to be small and decreases quickly as the number of principal points increases.

We now evaluate the impact of principal selection strategy on the regression accuracy of PRR-SINR modeling. Fig. 12 shows the CDF of MSEs of all measured models when the

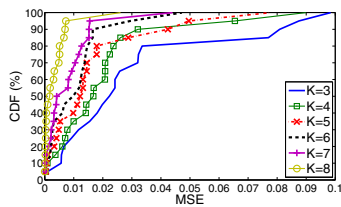


Figure 10. The CDF of MSE for different number  $K$  of principal SINR points.

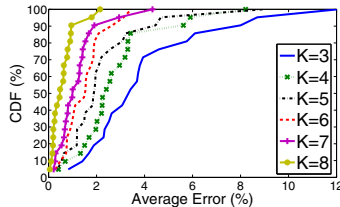


Figure 11. The CDF of prediction errors for different number  $K$  of principal SINR points.

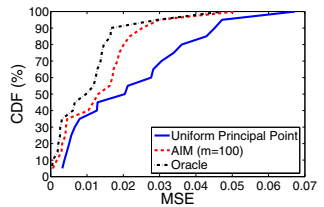


Figure 12. The CDF of MSE for different strategies of principal-SINR selection.

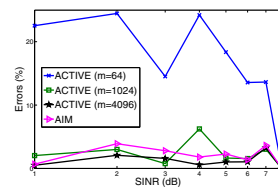


Figure 13. PRR-SINR models generated by ACTIVE and AIM.

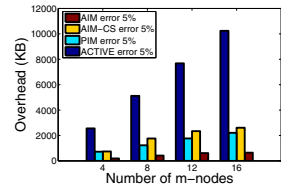


Figure 14. Overhead in AIM, AIM-CS, PIM, and ACTIVE.

number of principal SINR points is 6. Three principal selection strategies are evaluated. The “Uniform-Principal-Point” strategy selects the principal SINR points for an arbitrary node in the network using the AIM method and then uses these principal SINR points for every node in the network. The “Oracle” strategy continuously recomputes the optimal principal points based on the current network condition. Specifically, it selects the principal SINR points by analyzing all SINR measurements during a given period of time while a practical method like AIM only has access to the historical data before starting a new round of measurement and model generation. The Oracle strategy serves as a reference on the optimal performance that could be achieved. We see that the uniform-principal-point strategy performs the worst as it neglects the diversity among nodes. The gap between AIM and the oracle strategy is small, and the MSE is less than 4% most of the time, which is accurate enough for model generation. This result demonstrates the effectiveness of principal selection in AIM.

### C. Performance Comparison

Here we comparatively study AIM, PIM, and ACTIVE in terms of their accuracy and overhead. We first study the accuracy of AIM. For AIM, we use 6 measured (PRR,SINR) pairs for regression model generation, and collect 100 samples for each (PRR,SINR) pair of principal SINR point. That is, the measurement overhead of each model is 600 samples. Fig. 13 shows the PRR modeling error in AIM and ACTIVE, where the error is defined as the absolute difference between the actual PRR and the modeled PRR for each SINR point. For ACTIVE, we vary the number of samples (e.g., measurement packets) used for each model to evaluate its effect on the performance. We see that the accuracy of ACTIVE increases with number of samples used. We also observe that the AIM with 600 samples

achieves an accuracy similar to that in ACTIVE with 4096 samples. We observed that this is caused by the diversity of radio performance at different SINRs. Due to unpredictable environmental factors and hardware biases, the PRR of radio at each SINR has different variance. However, ACTIVE cannot choose the right number of samples for measuring each individual PRR. In contrast, based on the accuracy control mechanism, AIM controls the number of samples for each SINR point in the model measurement, which leads to significantly lower overhead.

We now evaluate the communication overhead of different methods. For a fair comparison, we measure the overhead when they achieve the same error of 5%, where the overhead is measured as the additional bytes used to generate the PRR-SINR models. For AIM, we set the accuracy bound to 5%. Each node in the network runs the accuracy control component to guide the PRR-SINR model generation. To understand the impact of adaptive sampling in AIM, we also study AIM-CS that is a variant of AIM without the adaptive sampling mechanism described in Section IV-B, (i.e., continuously sample the principal SINR points). We see that ACTIVE is the most costly approach with the overhead growing linearly with the number of m-nodes. AIM-CS, PIM and AIM perform significantly better than ACTIVE, and their overhead do not increase quickly with the number of m-nodes. This is because they employ a passive approach for model generation where interference samples are collected from normal network traffic and forwarded to the aggregator by piggybacking data packets. We also observe that AIM introduces the lowest overhead. Compared with AIM-CS and PIM, AIM stops sampling when the required modeling accuracy is achieved. This result demonstrates the effectiveness of accuracy-aware sampling in AIM.

## VII. CONCLUSION

This paper presents an *accuracy-aware* approach to interference modeling and measurement for WSNs. First, we propose a new regression-based PRR-SINR model whose accuracy is analytically characterized based on statistics theory. Second, we develop a novel protocol called *accuracy-aware interference measurement* (AIM) for measuring the proposed PRR-SINR model with *assured accuracy*. AIM enforces a specified accuracy bound by adaptively controlling the measurement process at run time. AIM also adopts new clock calibration and in-network aggregation techniques to reduce the overhead of interference measurement. We conduct extensive experiments on a 17-node testbed of TelosB motes to evaluate the performance of AIM. Our results show that AIM achieves high accuracy of PRR-SINR modeling with significantly lower overhead than state of the art approaches.

## ACKNOWLEDGMENT

This work is supported, in part, by the National Science Foundation under grants CNS-0916576 and CNS-0954039 (CAREER).

## REFERENCES

- [1] TelosB sensor node. <http://www.crossbow.com/>.
- [2] TinyOS collection tree protocol (CTP). <http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html>.
- [3] G. Brar, D. M. Blough, and P. Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *MobiCom*, 2006.
- [4] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *ACM SenSys*, 2003.
- [5] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric sinr. In *MobiHoc*, 2007.
- [6] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46:388–404, 2000.
- [7] J. Huang, S. Liu, G. Xing, H. Zhang, J. Wang, and L. Huang. Accuracy-aware interference modeling and measurement in wireless sensor networks. Technical Report MSU-CSE-11-3, Michigan State University, 2011.
- [8] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, Inc., 1991.
- [9] A. Kashyap, S. Ganguly, and S. R. Das. A measurement-based approach to modeling link capacity in 802.11-based wireless networks. In *MobiCom*, 2007.
- [10] S. Liu, G. Xing, H. Zhang, and J. Wang. Passive interference measurement in wireless sensor networks. In *IEEE ICNP*, 2010.
- [11] R. Maheshwari, S. Jain, and S. R. Das. A measurement study of interference modeling and scheduling in low-power wireless networks. In *SenSys*, 2008.
- [12] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A general model of wireless interference. In *MobiCom*, 2007.
- [13] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference in static wireless networks. In *SIGCOMM*, 2006.
- [14] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang. C-mac: Model-driven concurrent medium access control for wireless sensor networks. In *Infocom*, 2009.
- [15] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of concurrent transmission in wireless sensor networks. In *Sensys*, 2006.
- [16] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Sensys*, Los Angeles, CA, November 2003.