Passive Interference Measurement in Wireless Sensor Networks

Shucheng Liu[1,2]; Guoliang Xing[3*]; Hongwei Zhang[4]; Jianping Wang[2]; Jun Huang[3]; Mo Sha[5]; Liusheng Huang[1]
[1]University of Science and Technology of China and USTC-CityU Joint Advanced Research Centre;
[2]City University of Hong Kong; [3]Michigan State University, USA;
[4]Wayne State University, USA; [5]Washington University in St. Louis, USA

***Abstract*—Interference modeling is crucial for the performance of numerous WSN protocols such as congestion control, link/channel scheduling, and reliable routing. In particular, understanding and mitigating interference becomes increasingly important for Wireless Sensor Networks (WSNs) as they are being deployed for many data-intensive applications such as structural health monitoring. However, previous works have widely adopted *simplistic* interference models that fail to capture the wireless realities such as probabilistic packet reception performance. Recent studies suggested that the physical interference model (i.e., PRR-SINR model) is significantly more accurate than existing interference models. However, existing approaches to physical interference modeling exclusively rely on the use of active measurement packets, which imposes prohibitively high overhead to bandwidth-limited WSNs. In this paper, we propose the *passive interference measurement (PIM)* approach to tackle the complexity of accurate physical interference characterization. PIM exploits the spatiotemporal diversity of data traffic for radio performance profiling and only needs to gather a small amount of statistics about the network. We evaluate the efficiency of PIM through extensive experiments on both a 13-node and a 40-node testbeds of TelosB motes. Our results show that PIM can achieve high accuracy of PRR-SINR modeling with significantly lower overhead compared with the active measurement approach.**

I. Introduction

Interference is a fundamental issue in wireless networks. Due to the broadcast medium, wireless transmissions from one radio interfere with the receptions of surrounding radios resulting in packet loss and low network throughput. Accurate characterization of the performance of links/nodes under interference is crucial for the efficient operation of many wireless protocols such as congestion control [15], rate allocation [14], and link/channel scheduling [5][7][12]. In particular, Wireless Sensor Networks (WSNs) are being increasingly deployed for data-intensive applications, such as structural health monitoring [21], which constantly suffer from interference caused by heavy traffic flows. It is thus critical to accurately measure and model the interference among wireless sensors in these applications.

Previous works on WSNs have widely adopted *simplistic* interference models for protocol design and evaluation. Examples include the *disc* model (also referred to as the *protocol interference model*) and the hop model [11]. By adopting distance or hop-based metrics, these models greatly simplify the design and evaluation of wireless networks. Unfortunately, they have largely failed to capture the complex realities of low-power wireless radios such as the probabilistic packet reception caused by environmental noise and interfering transmissions [11][19][22].

Recently, several empirical studies[11][17][18][19] have been conducted to investigate the interference on various wireless platforms. Despite the variation of the results, these studies have suggested that the packet-level physical interference model (also referred to as the packet reception ratio (PRR) versus SINR model or PRR-SINR model) can be estimated based on packet statistics and the received signal strength (RSS) available on commodity radios. In contrast to the existing simplistic models, the PRR-SINR model offers significantly improved realism by accounting for the impact of various dynamics (e.g., environmental noise and concurrent transmissions). Recent studies [5][7][11][17][18] showed that numerous protocols on link scheduling, topology control, and medium access control (MAC) can significantly benefit from using the PRR-SINR model.

However, a key challenge of fully exploring the potential of PRR-SINR model in practical protocol design lies in the high complexity of accurately measuring it at run time. In particular, to achieve satisfactory measurement accuracy, the reception performance of a radio must be carefully profiled under different SINRs, which incurs high time complexity and message overhead [11][19]. The existing interference measurement methods are based on the *active* approach where each node must transmit/receive extensive measurement packets in order to generate accurate PRR-SINR models. For instance, in [9][13][16], PRR-SINR models need to be seeded by $O(N)$ trials in an N-node network where each node transmits in turn while receivers measure the channel condition. Such high complexity, although possibly acceptable for 802.11-based networks, can easily counteract the benefit of adopting the PRR-SINR model in bandwidth-limited WSNs.

In this paper, we propose a new approach called *passive interference measurement (PIM)* to tackle the complexity of accurate physical interference characterization. A key advantage of PIM over existing active interference measurement approaches is its extremely low overhead: each node only needs to measure a small amount of statistics such as timestamps and received signal strength (RSS) of data packets. Moreover, it completely avoids frequent time synchronization [19] or

*Correspondence author.

the transmission of a large amount of measurement packets [9][11][16]. The collected statistics are used to infer the interference among nodes by mining the correlation between PRR and RSS measurements, and eventually build the PRR-SINR models. PIM is opportunistic in nature as it exploits the spatial and temporal diversity of data traffic to profile radio performance under different levels of interference. We implemented PIM in TinyOS-2.0.2 and conducted extensive experiments on both a portable 13-node and a fixed 40-node testbeds of TelosB motes[2]. Our experimental results show that PIM 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 problem we study in this paper. Section IV presents the design and implementation of PIM. In Section V, we discuss how to minimize the set of reference nodes used by PIM for model generation. Section VI presents the experimental results and Section VII concludes the paper.

II. Related Work

The previous studies that are related to this paper fall into the following three directions.

Link quality measurement and modeling: Recent empirical studies [6][20][22] revealed the prevalence of lossy and asymmetric links in low-power wireless networks. Moreover, the communication range of nodes exhibits a transitional region where receivers experience highly variable reception performance. In [24], an analytical model is proposed to estimate the transitional region in the communication range of CC1000 radios. The root causes of the variable link-level performance include external noise, random interference, and the transitional region in the PRR-SINR relationship of radio transceivers.

Measurement-based interference modeling: Son et al. [19] studied the PRR-SINR model of CC1000 radios and showed that the SINR threshold changes with the number of interferers. This result, however, is inconsistent with more recent findings on other radio platforms (e.g., CC2420) [11][17]. In [11], a set of interference models, including the PRR-SINR model, disc model, and the thresholded physical interference model are studied based on CC2420 radios for their modeling accuracies and impacts on link scheduling performance. In particular, it is shown that adopting the PRR-SINR model can lead to significant link throughput improvement. Reis et al. [16] presented interference and packet delivery models that can be instantiated by packet transmission traces. Qiu et al. [13] proposed a general interference model to characterize the interference among arbitrary number of 802.11 senders and predict the resultant throughput. 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 interference measurement, which requires nodes to periodically transmit and receive control packets..

Interference-aware link scheduling and MAC protocols: The problem of link scheduling under the physical interference model has recently received significant attention. It is shown in [7] that the problem of finding a minimum-length collision-free schedule is NP-complete. A computationally efficient heuristic with provable performance bound is proposed in [5]. The complexity of scheduling a set of communication requests is also studied in [12]. In our earlier work[17], a new MAC protocol called C-MAC is developed to maximize the aggregate throughput of a wireless cell based on the empirical PRR-SINR model. All the above works require accurate interference models.

III. Problem Statement

The objective of this work is to accurately measure the PRR-SINR interference model with the minimum overhead. In this section, we first discuss the characteristics of PRR-SINR model. We then motivate this work using empirical results on TelosB motes.

A. Understanding the PRR-SINR model

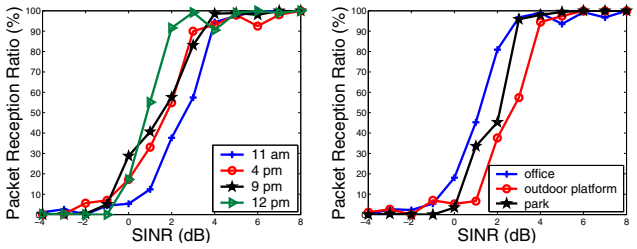
According to communication theory, the bit error rate (BER), i.e., the probability that a transceiver r successfully receives an incoming bit τ, denoted by $p_r(\tau)$, is governed by the following model:

$$p_r(\tau) = Prob\left[\frac{\text{signal power of } \tau}{I_r + n_r} > \beta_r\right] \quad (1)$$

where I_r is the interference experienced at r, which is equal to the power of other nodes' transmissions and electromagnetic signals from the environment. n_r is a random variable that equals the power of ambient noise. β_r is a constant determined by the modulation scheme and the transceiver sensitivity. Unfortunately, the above BER-SINR model cannot be directly measured on commodity radio transceivers [16]. As a result, most recent empirical studies [9][13][16][19] have adopted a *measurement-based* packet-level interference model that correlates packet reception ratio with SINR and is also referred to as the PRR-SINR model. In the PRR-SINR model, the probability that a transceiver r successfully receives an incoming packet ω is given by:

$$p_r(\omega) = f\left(\frac{RSS(\omega)}{RSS(I_r) + \overline{n}_r}\right) \quad (2)$$

where function $f(\cdot)$ can be determined by the measurements of SINR and PRRs. $\overline{n}_r$ is the measured *average* power of ambient noise. $RSS(\omega)$ and $RSS(I_r)$ are the signal power of packet ω and interfering transmissions I_r, respectively. In contrast to the BER-SINR model, the PRR-SINR model can be measured on most commodity radio transceivers. In particular, the RSS values and $\overline{n}_r$ in Eq. (2) can be obtained from a radio hardware register called RSS Indicator (RSSI) that is available on commodity wireless platforms. And $p_r(\omega)$ can be measured as the link-level packet reception ratio (PRR). Moreover, the PRR-SINR model is critical for optimizing wireless protocol performance because it can *predict* the PRR of a link when it experiences interference. Due to the measurability and the



(a) Measurement at different times (b) Measurement at different locations

Fig. 1. The PRR-SINR models measured on the same node at different locations and times.

prediction power, the PRR-SINR model has been widely used in the empirical performance modeling of wireless links [9][13][16][19].

B. Measuring the PRR-SINR model

We now experimentally analyze the PRR-SINR model based on our measurements on TelosB motes equipped with CC2420 radios. The details of the measurement methodology are discussed in the rest of this paper. Fig. 1(a) shows the models measured in an office at different times during a day. Fig. 1(b) shows the models measured in three different environments: an office, an outdoor open platform, and a small park. Two observations can be made from these results. First, the PRR-SINR curve has a *transitional region* of about 5 dB wide in which the PRR gradually grows from zero to one. This result is consistent with the findings reported by recent empirical studies [9][13][16][19] except the slight variation in the width of the transitional region. Second, the PRR-SINR model yields significant spatial and temporal variation. Fig. 1 shows that the PRR of a radio under the same SINR may vary as much as 55%. These results demonstrate the need of measuring the PRR-SINR model in an *online* manner at run time. This requirement poses a major challenge for existing *active* interference modeling methods as they exclusively relied on carefully controlled measurements [17][19] or extensive data trials [9][16].

Several mathematical functions have been proposed to describe the PRR-SNR model in previous studies. For instance, two functions are specified in the 802.15.4 standard [8] and TinyOS 2.1 [4][1]. However, it is challenging (if not impossible) for these models to capture the significant spatial and temporal variation of the PRR-SNR relationship measured in reality (see Fig. 1). Although the passive measurement approach proposed in this paper can be used to build various realistic interference models, we employ a measurement-based PRR-SINR model as follows. For a node v, $\Theta(v)$ represents a set of tuples $(prr, sinr)$ in node v's PRR-SINR model, where $sinr$ is an SINR value (in the unit of dB) within the transitional region $[t_l(v), t_u(v)]_{dB}$ and prr is the PRR of node v when the SINR is equal to $sinr$. $\Theta(v)$ can be formally defined as follows.

$$\Theta(v) = \{(prr, sinr) \mid prr \in [0,1]; sinr \in [t_l(v), t_u(v)]_{dB}\} \quad (3)$$

[1]The model is implemented in tinyos-2.x/tos/lib/tossim/CpmModelC.nc.

The above model is essentially a discrete version of the function defined in Eq. (2). A similar model was used in [17] to design a new MAC protocol in TinyOS 2.x that allows nodes to tune their transmit power for throughput maximization. The key advantage of the model in Eq. (3) is that it can accurately capture the relationship between PRR and SINR based on the RSSI values provided by most commodity radios. We define $sinr$ as integer decibels (dB) because the RSSI of most low-power radios (e.g., CC2420 and CC1000) offer a sensitivity of 1 dBm. However, Eq. (3) can be easily extended for radios with more sensitive RSSIs.

The objective of this work is to accurately measure the PRR-SINR model defined in Eq. (3) with the minimum overhead. Specifically, our problem can be defined as follows. For a given set of nodes Φ in the network, the objective is to measure $\{\Theta(v) \mid v \in \Phi\}$ - the PRR-SINR models of all nodes in Φ. We note that set Φ may include all or a subset of nodes in the network whose interference conditions are critical for system performance. For instance, multi-hop 802.15.4 networks are typically composed of a number of clusters with star topology. Φ may include all cluster heads in such a case. In this paper, we propose a novel approach called *Passive Interference Measurement (PIM)* that can measure the PRR-SINR model at extremely low overhead.

IV. The Design and Implementation of PIM

This section describes the design and implementation of PIM. We first give a brief overview of our approach in Section IV-A. The components of PIM are discussed in details in Sections IV-B to IV-D.

A. Overview

The major design objectives of PIM are *low overhead* and *high accuracy*. In contrast to the existing methods [9][16][17][19] that rely on extensive special measurement packets, PIM uses the statistics of data packets for building the PRR-SINR model, and only generates light extra traffic. This feature is particularly desirable for bandwidth-limited WSNs.

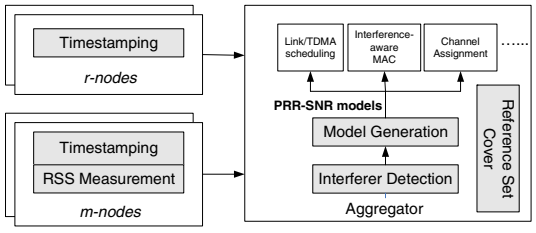


Fig. 2. The system architecture of PIM.

We assume that network has a tree-based topology in which all nodes send their data to the base station or *aggregator* node. We note that such a topology is commonly used in WSN applications. PIM does not depend on any particular MAC protocol. It generates PRR-SINR models by measuring packet-level interference in the network. We note that significant interference 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 as discussed in Section III.

Fig. 2 shows the system architecture of PIM. It is composed of three parts that reside on different nodes in the network. We refer to the nodes whose PRR-SINR models are to be measured as *m-nodes*. PIM employs an asymmetric architecture: ordinary nodes only collect simple packet statistics such as transmission/reception times and RSS, while the aggregator analyzes the statistics and generates the PRR-SINR models of m-nodes. In Section IV-E, we discuss how to extend the design of PIM to the case where the PRR-SINR models can be generated by non-aggregator nodes in a network.

For a given m-node, PIM chooses a set of nodes, referred to as *reference nodes* or *r-nodes*, to help measure the PRR-SINR model of the m-node. An important property of r-nodes is that their transmissions must interfere with the packet reception of the m-node. In other words, an r-node may be a child of the m-node on the data routing tree or an interferer whose data transmissions interfere with the data reception of the m-node. We note that an m-node can also be the r-node of another m-node.

The *timestamping* component records the time when an r-node forwards each packet and the time when an m-node receives each packet. In addition, the *RSS measurement* component on m-nodes records the RSS values of the received packets. Note that the transceiver of an m-node works in the promiscuous mode and records the information of the packets it overhears. The recorded statistics are then transmitted to the aggregator, which generates the PRR-SINR models of m-nodes. The timing information of packet transmission/reception provides important clues about the signal contention at each m-node. Combined with the RSS measurements, they can be used to determine the SINR of each packet reception. The PRR of each m-node can also be computed from the packet timing information. Finally, the *model generation* component of aggregator constructs the PRR-SINR model of each m-node.

A key advantage of the PIM architecture is that it introduces little overhead at resource-constrained nodes while letting the resource-rich aggregator perform more computationally intensive tasks such as data processing and model generation. The generated PRR-SINR models can be directly provided to other components in the system, such as centralized link/TDMA scheduling algorithms [5][7] that are executed by the aggregator. If the PRR-SINR models are needed by a distributed system component, they can be sent to other nodes by the aggregator. As discussed in Section III-B, a PRR-SNR model can be efficiently encoded by a few bytes.

We now use a simple example to illustrate the basic idea of PIM. In Fig. 3, the PRR-SINR model of the m-node m_1 will be measured using two r-nodes, r_1 and r_2. The solid arrows represent the data communication links while the dashed arrow represents the interference link. The time

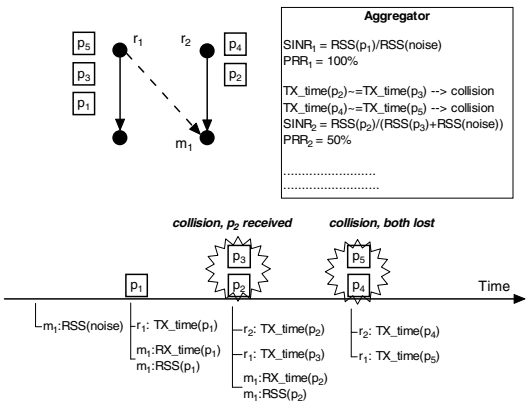


Fig. 3. An example of PRR-SINR model generation by PIM.

sequence of packet transmissions is shown below the topology. After p_1 is transmitted by r_1, it is successfully received by m_1. r_1 records the time when packet p_1 is transmitted while m_1 records the reception time and its RSS. As p_2 and p_3 are transmitted by r_2 and r_1 roughly at the same time, they collide at node m_1. However, p_2 is successfully received by m_1 due to higher signal power. In contrast, when p_4 and p_5 collide at node m_1, both of them are lost. Such a probabilistic reception performance suggests that, when r_1 and r_2 transmit simultaneously, the resulting SINR falls in the transitional region of the PRR-SINR model of m_1. Based on the statistics collected by r_1, r_2, and m_1, the aggregator will find scenarios with different SINRs and then compute the PRR for each of them. Two different SINRs can be found in this example. The SINR of p_1 can be computed by the RSS of p_1 and noise. Moreover, the SINR of two packet collisions can be computed by the RSS of p_2, p_3 and noise, where,for simplicity of this example, the RSS of p_3 is assumed to equal that of p_1 and the noise power is assumed to equal the one measured before the reception of p_1. The PRRs for the above two cases are 100% and 50%, respectively. In realistic settings with more nodes and packet transmissions, the aggregator can find SINRs that are more different and compute the corresponding PRRs, which allow to construct the complete PRR-SINR model of the m-node of interest. We note that the PRR-SINR model constructed in this example has little statistical significance due to the small sample size. However, this issue can be addressed when the aggregator accumulates more statistics.

As illustrated in the above example, PIM is opportunistic in nature as it exploits the spatiotemporal diversity of data traffic for profiling the transceiver's reception performance under different SINRs. The accuracy of the PRR-SINR models measured by PIM depends on the correct identification of packet-level interference and computation of the resulted SINRs at m-nodes. PIM identifies packet interference by analyzing the temporal correlation among packet transmission/reception events. However, considerable false positives may be resulted because packets transmitted at the same time by two nodes far apart do not necessarily cause interference at the receiver. Filtering out such false positives is not trivial as real interferers of the receiver may lie outside of the communication range.

The *interferer detection* component of PIM employs a novel mechanism to accurately detect packet-level interference with a low false positive probability.

The main overhead of PIM is due to the use of r-nodes as they merely help the PRR-SINR measurement of m-nodes. However, we show that minimizing the set of r-nodes while achieving the same level of model accuracy is NP-hard. PIM includes an efficient algorithm (*reference set cover* component in Fig. 2) that greedily chooses the r-nodes based on their utilities, which can significantly reduce the number of total r-nodes needed in model measurement.

B. Timing and RSS measurement

We now describe how PIM measures the timing and RSS of packets. Specifically, an r-node records the time instance when each packet is forwarded by it. An m-node records the time instance when each packet is received or overheard, the RSS of the packet, and the RSS of external (background) noise. The information recorded by each node is periodically aggregated into special packets and sent to the aggregator node. Alternatively, such information may be piggybacked in data packets; but this often incurs considerable overhead due to packet buffer copying, and hence is not adopted by PIM. The timing information (i.e., transmission and reception times) are used to estimate the *air-time* of a packet, which is crucial to infer the interference among packets. Moreover, as the reception time of a packet is recorded at each hop, it allows the aggregator to compute the PRR of a link.

An m-node obtains the RSS of an incoming packet from the meta data in the packet. In addition, each m-node periodically estimates the power of external (background) noise by reading the value from the RSSI register. However, the noise RSS may also contain the signal power of incoming packets transmitted by other nodes. PIM can filter out such interference by using the packet transmission and reception times recorded.

Several issues must be addressed in order to accurately measure the timing information of packets. When CSMA is adopted at the MAC layer, several attempts may be needed before a node seizes the channel. PIM records the time (in milliseconds) when the channel is idle and the packet is passed to the transceiver. However, for a packet-based transceiver (i.e., CC2420), the exact time that the first bit of packet is transmitted to the channel is determined by the transceiver hardware. Our experiments show that the timing error caused by radio hardware is not significant.

C. Interference Detection

To measure the PRR-SINR model of an m-node, PIM needs to know the r-nodes whose transmissions interfere with the packet reception of the m-node. Existing methods of interference detection [23] require active probes among nodes. In contrast, PIM identifies interferers of an m-node by analyzing the timing and RSS measurements collected by both the m-node and the interfering r-nodes. Based on this information, the aggregator "reconstructs" the interference that occurred among the packet transmissions, computes the SINR of each received packet, and finally generates the PRR-SINR model.

A challenge of interference detection is that r-nodes are often out of the communication range of an m-node. As a result, packet timing information may lead to false positives. We now illustrate this using an example. Suppose an m-node a senses interference after receiving a packet with higher RSS. The air-time of the packet overlaps with those of two packets sent by r-nodes b and c, respectively. If both b and c are out of the communication range of a, one cannot determine whether b or c or both are interferers. If both b and c are classified as interferers of a when only one of them is actually an interferer, the SINR of the received packet will be underestimated, thus leading to errors in the computed PRR-SINR model. We now discuss how PIM accurately identify interferers. We first define the following notation.

- $P(v)$ is the set of packets sent to m-node v; $p^i_{u,v} \in P(v)$ is the ith packet sent by node u.
- $RSS(p^i_{u,v})$ is the RSS of packet $p^i_{u,v}$.
- $J^i_{u,v}$ is the set of packets whose air-times overlap with the air-time of packet $p^i_{u,v}$, which is referred to as the *concurrent packet set* of packet $p^i_{u,v}$.
- $J_{u,v} = \{J^i_{u,v} \mid p^i_{u,v} \in R(v)\}$ is the set of concurrent packet sets of all packets that u sent to v, and $R(v)$ is the set of packets sent from u to v.

The concurrent packet set of packet $p^i_{u,v}$ includes all packets, which may potentially interfere with the reception of $p^i_{u,v}$. Several issues must be addressed to construct an accurate concurrent packet set. First, the transmission times of two colliding packets may not perfectly align with each other. The aggregator computes the air-time of each packet from both the transmission and reception times. Two packets are concurrent if their air times overlap. However, this approach is not applicable to the lost packets. As elaborated later, lost packets also need to be included in concurrent packets in order to compute the PRR-SINR model. In such a case, the aggregator compares the difference between transmission times of packets, and if it is within a predefined constant Δ, the packets are classified as concurrent packets. Δ is set to be half of the time it takes to transmit a packet in our implementation.

Note that a packet in concurrent packet set $J^i_{u,v}$ may not interfere with the reception of $p^i_{u,v}$ although their transmissions overlap. Our objective is to detect and remove *fake interfering packets* from $J_{u,v}$ for every m-node v. PIM achieves this based on the key observation that the signal attenuation between two nodes does not change substantially over a short time period (e.g., a few minutes) [10][17]. Accordingly, PIM utilizes the following two rules to eliminate fake interfering packets.

- **Rule 1**: Suppose two elements of $J_{u,v}$, set $J^i_{u,v}$ and $J^j_{u,v}$, satisfy $J^i_{u,v} \subset J^j_{u,v}$ and $RSS(p^i_{u,v}) = RSS(p^j_{u,v})$. Then node w is a fake r-node of node v, if and only if $p^k_{w,t} \in J^j_{u,v} \setminus J^i_{u,v}$.
- **Rule 2**: If node w is a fake r-node of node v, then any packet sent by w does not interfere with any packet received by u.

In Rule 1, if two concurrent packet sets $J^i_{u,v}$ and $J^j_{u,v}$ satisfy $J^i_{u,v} \subset J^j_{u,v}$ while they lead to the same RSS observed by node v, the nodes in $p^k_{w,t} \in J^j_{u,v} \setminus J^i_{u,v}$ must be fake r-nodes. This is because, if these nodes were real r-nodes, a larger RSS should be resulted due to higher interference. In Rule 2, if an r-node has been classified as a fake r-node of m-node v, then any packet from the same r-node does not interfere with v. We note that both rules may be violated over time when a transmitter's signal strength weakens (e.g., due to higher path loss). However, PIM only applies the rules over a short history window of traffic, which ensures the validity of them. In addition, the two rules assume fixed transmit power for all nodes. When a node varies its transmit power, it may carry the new power in its packets and the two rules can be applied to the packets sent at the same power. The pseudo code of algorithm for detecting fake interfering packets is shown in Algorithm. 1.

Algorithm 1 Fake interfering packet detection for $J_{u,v}$

Sort set $J_{u,v}$ in ascending order of cardinality: $J_{u,v} = \{J^1_{u,v}, J^2_{u,v}, \cdots, J^n_{u,v}\}, |J^1_{u,v}| \leq |J^2_{u,v}| \cdots \leq |J^n_{u,v}|$;
$x = 1$;
while $x \leq n$ **do**
  Use $J^x_{u,v}$ to detect and remove fake interfering packets in $J^{x+1}_{u,v}, \cdots, J^n_{u,v}$ according to Rule 1;
  Remove fake interfering packets from all sets in $J_{u,v}$ according to Rule 2.
  $x++$
end while
return $J_{u,v}$

The algorithm first orders all concurrent packet sets in the ascending order of their cardinality. It then iterates through all sets, and applies Rule 1 for the set under consideration and all the sets after it. Rule 2 is applied by using the fake interfering packets detected by Rule 1. It can be seen that *any* fake interfering packet that can be detected by Rule 1 or 2 would be removed from the concurrent packet sets when the algorithm terminates. This is due to the fact that Rule 1 always uses a concurrent packet set to detect the fake r-nodes in a larger set, and every set has been examined using all sets smaller than it at the end of the algorithm.

D. Model Generation

After removing the fake interfering packets, the aggregator generates the PRR-SINR model (defined in Eq. (3)) of each m-node as follows. It first estimates the SINR of each packet (both successfully received or lost) by using the RSS and noise measurements. The PRR of the packets with the same SINR is then computed. We define the following notation used in the model generation. $R(v)$ and $L(v)$ are the sets of packets that v received or were lost. $SINR(p^i_{u,v})$ is the SINR of packet $p^i_{u,v}$. $[t_l(v), t_u(v)]_{dB}$ is transitional region of PRR-SINR model of node v.

For packet $p^i_{u,v}$ that is received by v, $SINR(p^i_{u,v})$ can be computed as the ratio of RSS of the packet $RSS(p^i_{u,v})$ to the sum of RSS of interfering packets $J^i_{u,v}$ and noise power $\overline{I}$:

$$SINR(p^i_{u,v}) = \frac{RSS(p^i_{u,v})}{\sum_{p^j_{x,y} \in J^i_{u,v}} RSS(p^j_{x,y}) + \overline{I}} \quad (4)$$

$$SINR(p^i_{u,v})_{dB} = 10log_{10}(SINR(p^i_{u,v})) \quad (5)$$

If packet $p^i_{u,v}$ was lost, the measurement of $RSS(p^i_{u,v})$ is not available. In such a case, PIM uses the RSS of the last packet successfully received from node u, as the RSS from the same sender remains stable over a short period of time according to our experimental results and recent empirical findings [10]. The RSS of a lost packet that interferes with $RSS(p^i_{u,v})$ can be estimated similarly. After the SINR of each packet is obtained, the aggregator computes the reception ratio of all packets that have the same SINR according to Eq. (6). Finally, PIM generates the model as the set of PRR-SINR pairs that fall in the transitional region. Our experimental results show that the SINR of transitional region ranges from 0 dB to about 5 dB.

$$PRR_u(x_{dB}) = \frac{|\mathcal{X}|}{|\mathcal{X}|+|\mathcal{Y}|}, \quad x \in [t_l(v), t_u(v)] \quad (6)$$
$$\mathcal{X} = \{p^i_u | (SINR(p^i_u) = x) \wedge (p^i_u \in R(u))\}$$
$$\mathcal{Y} = \{p^i_u | (SINR(p^i_u) = x) \wedge (p^i_u \in L(u))\}$$

E. Discussions

We now discuss several issues in the design of PIM that have not been addressed in earlier sections.

Although our discussion is focused on the PRR-SINR model given in Eq. (3), PIM can also be applied to the measurement of realistic interference models. As discussed in Section IV-D, PIM estimates a set of (PRR, SINR) pairs from the measurement results, which can be used to build various interference models. For instance, the regression model proposed in [19] can be easily instantiated by a set of (PRR, SINR) pairs.

In the design of PIM, the statistics measured by nodes are sent to the aggregator to generate the PRR-SINR models. The overhead of collecting statistics can be reduced by allowing nodes in the network to perform model generation. For a given m-node x, we can choose a node g to generate the model of x as follows. First, node g should be an upper stream node (i.e., closer to the base station) so that it can aggregate the measurement results gathered by downstream nodes. Moreover, both x and the r-nodes of x should be g's offsprings on the routing tree because their measurements are needed to generate the model of x. We note that the r-nodes of x can be easily identified by PIM based on the interference detection algorithm discussed in Section IV-C.

The accuracy of PRR-SINR models generated by PIM is related to several factors. First, PIM requires the existence of multiple SINRs in the packet receptions of an m-node in order to characterize the transitional region of the PRR-SINR model. Moreover, the amount of packet statistics gathered is also important for the statistical accuracy of measurement. As discussed earlier, PIM addresses these issues by taking advantage of the significant spatial and temporal dynamics of packet-level interference. Our experimental results in Section VI show that, these requirements can be satisfied even in a small network of 13 nodes with moderate traffic load. In

TABLE I
EXAMPLE OF R-NODE SET AND R-NODE SET SPACE

m-node	SINR	r-node Set Space
m_1	1_{dB}	$\{j_1,j_2\}$, $\{j_4,j_5\}$
m_1	2_{dB}	$\{j_1,j_3\}$, $\{j_2,j_3\}$, $\{j_3,j_4,j_5\}$
m_2	1_{dB}	$\{j_2,j_3\}$, $\{j_3,j_4\}$, $\{j_1,j_4,j_5\}$
m_2	2_{dB}	$\{j_1,j_3\}$, $\{j_1,j_5\}$, $\{j_3,j_5\}$

addition, PIM can be turned off to reduce overhead once enough packet statistics are gathered for model generation. Finally, as PIM only gathers statistics of data packets, it can be easily integrated with existing sleep scheduling protocols to reduce network power consumption.

V. MINIMUM REFERENCE SET COVER

The major overhead of PIM is due to the use of reference nodes to help the measurement of m-nodes' PRR-SINR models. As interference condition of the network is time-varying, PIM periodically reselects the r-nodes to ensure the required level of accuracy for the PRR-SINR models that are measured. Therefore, it is desirable to minimize the total number of r-nodes while maintaining the same level of accuracy in PRR-SINR model measurement. We assume the accuracy of a generated PRR-SINR model is determined by the total number of samples ($|\mathcal{X}| + |\mathcal{Y}|$ in Eq. (6)) used for computing each PRR-SINR pair. This is reasonable as more samples lead to a smaller statistical variation of measurement. In the following, we define a problem, referred to as the *minimum reference set cover*, which seeks to minimize the total number of r-nodes while achieving a given number of PRR-SINR measurement samples. We show that this problem is NP-hard and then propose an efficient greedy algorithm.

We first define the following notion. Suppose k samples are required for computing each PRR-SINR pair in Eq. (6) in order to achieve the desired level of statistical accuracy. We set k as 10 in our implementation. We denote $j_v(x)$ as *node v's r-node set under SINR* x, i.e., the set of r-nodes whose simultaneous transmissions collide with at least k packets destined to m-node v, and the SINR of each packet reception is x dBm. $\mathcal{J}_v(x) = \{j_v(x)\}$ is *v's r-node set space under SINR* x, which includes all the v's r-node sets under SINR x.

We now illustrate the above definition of r-node set and r-node set space in Table I. Suppose we need to measure the PRR-SINR model for two m-nodes m_1 and m_2, and both nodes have a 2 dB transitional region in their PRR-SINR models. Each row in Table I shows an r-node set space that contains several r-node sets that lead to the same PRR-SINR measurements. For instance, on the first row, the transmissions of nodes in either $\{j_1, j_2\}$ or $\{j_4, j_5\}$ collided with at least k incoming packets at node s_1 and the resulting SINR is 1 dB. The aggregator can obtain such information from the history data reported by the m-nodes and r-nodes in the network. Our objective is to find a minimum set of r-nodes that can cover at least one r-node set in each row. It can be seen the optimal solution is $\{j_1, j_2, j_3\}$. That is, only these three r-nodes need to execute PIM while the level of model accuracy is the same as using all five r-nodes. The minimum reference set cover problem can be formally defined as follows.

Definition 1 (Minimum Reference Set Cover): Given the set of m-nodes Φ, sets of transitional regions $\{[t_l(v), t_u(v)] \mid v \in \Phi\}$, and the r-node set space for each m-node $\{\mathcal{J}_v(x) \mid v \in \Phi\}$, find a set of r-nodes such that all r-node set spaces contain at least one subset of Ψ while the total number of r-nodes in Ψ is minimized. Formally, we have:

$$minimize \quad |\Psi| \tag{7}$$
$$\Psi = \{j_v(x) \mid \forall x \in [t_l(v), t_u(v)],\ \forall v \in \Phi,$$
$$\exists j'_v(x) \in \mathcal{J}_v(x), j'_v(x) \subseteq j_v(x)\}$$

We have the following theorem regarding the hardness of this problem.

Theorem 1: The Minimum Reference Set Cover problem is NP-hard.

Proof: We prove by a reduction from the Minimum Vertex Cover (MVC) problem. Given a graph, the goal of MVC is to find a minimum set of vertices such that any edge in the graph has at least one endpoint in the set. Consider a special case of our problem where each r-node set contains only one r-node and each r-node set space only contains two r-node sets. Construct a graph $G(V, E)$ where each edge corresponds to an r-node set space for a given node and SINR value. The total number of edges in the graph is $\sum_{v\in\Phi} t_u(v) - t_l(v)$. As each r-node set contains a single r-node, the Minimum r-node Set Cover problem is equivalent to finding the minimum number of r-node sets such that any r-node set space contains at least one chosen r-node set. This problem is identical to finding the minimum vertex cover of graph $G(V, E)$. As the MVC problem is NP-hard, the Minimum r-node Set problem is also NP-hard. ■

We now describe an efficient greedy algorithm for solving the above problem. We define the utility of each r-node set as the ratio of its cardinality to the frequencies that it appears. We then select the r-node sets in the ascending order of their utilities until all r-node set space are covered. In the example shown in Tab. I, both r-node sets $\{j_1, j_3\}$ and $\{j_2, j_3\}$ have the highest utility of one as they appear twice in the table while the utility of any other r-node set is below one. After choosing the two sets, all r-node set spaces are covered, i.e., any row contains at least a subset of the union of two sets $\{j_1, j_2, j_3\}$.

VI. EXPERIMENTATION

This section presents experimental results of PIM. We evaluate the accuracy, convergence, and overhead of PIM using two high fidelity WSN testbeds. In what follows, we first describe our experimentation methodology, and then discuss our measurement results.

A. Methodology

We implemented PIM in TinyOS-2.0.2 and used both a 13-node portable testbed and a 40-node static testbed of TelosB [2] motes in our experimental evaluation. The portable testbed

Fig. 4. *NetEye* wireless sensor network testbed.

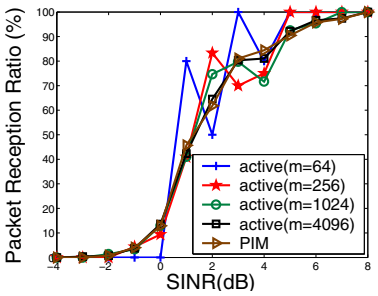


Fig. 5. PRR-SINR models generated using passive and active measurements.

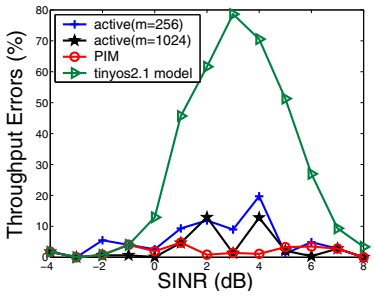


Fig. 6. Errors of throughput prediction using passive and active measurements.

enables us to evaluate PIM in a variety of settings, thus ensuring the generality of our conclusions. The portable testbed is deployed in an outdoor square of $40m \times 40m$. We evaluate the performance of PIM with different network topologies by changing the placement of nodes. In all experiments, we observed similar results. Due to the limited space, we only present the data for one scenario, where 12 motes are placed on a 3×4 grids. The max distance between two adjacent nodes is 10m. To corroborate our portable-testbed based results in large scale networks, we also conduct experiments remotely on the *NetEye* testbed at Wayne State University [1]. NetEye is an open testbed composed of 130 TelosB motes deployed in a 13×10 grid as shown in Fig. 4, where every two closest neighboring motes are separated by 2 feet. Each of these TelosB motes is equipped with a 3dB signal attenuator and a 2.45GHz monopole antenna to obtain multi-hop wireless topologies. In our experiments, we use a 10×4 subgrid of NetEye.

We study the performance of PIM in comparison to an active interference measurement method referred to *ACTIVE* [17][19]. ACTIVE generates a PRR-SINR model using m measurement rounds as follows. In each round, n r-nodes are selected for a pair of sender-receiver to build the PRR-SINR model of the receiver. Each round starts with a *syn* packet transmission that synchronizes the states of sender, receiver, and r-nodes. Then, the sender and the n r-nodes take turns to transmit packets to the receiver so that the receiver can measure the received signal strength (RSS) from the sender and each of the r-nodes. Finally, the sender and all the r-nodes transmit simultaneously so that the receiver can measure the packet reception status while all the r-nodes are transmitting (which corresponds to a specific SINR setting). To measure the statistics of PRR, we repeat the above round for m times and count the number of packets delivered or lost with the associated SINR values. To obtain different SINR values, we vary nodes' transmit power levels from 3 to 9. It is shown in [17][19] that the PRR-SINR models built by ACTIVE are highly accurate if a large number of rounds are used. We compare PIM with this method with respect to accuracy and overhead.

Our evaluation is based on a data collection scenario as follows. In our portable testbed, 12 motes in a 4×3 grid transmit packets to the sink. In NetEye testbed, we choose 40 nodes in a 10×4 grid, where the node in the middle of last row serves as the sink and all other nodes are sources. In both testbeds, nodes' transmit power level is set to 3 so that realistic multi-hop scenarios can be created. Each path to the sink has up to five hops in the 13-mote testbed. Each experiment in the portable testbed and NetEye lasts 30 minutes. For the PIM experiments, we employ the collection-tree-protocol (CTP) [3] in TinyOS as the routing protocol. During the data collection process, each node records and delivers to sink the information needed by PIM to derive PRR-SINR models as discussed in previous sections. We note that CTP may change the routing paths dynamically in response to the variations of wireless link quality. The maximum number of hops on a path of NetEye is between 5 to 9 hops.

B. Accuracy of PIM

We first evaluate the accuracy of PRR-SINR models generated by PIM. Each source in the experiments generates a traffic load of 10 packets/s. The transmission times of 10 packets are uniformly distributed at random in a second. A broadcast based simple time synchronization protocol synchronizes the clocks of all nodes every 5 second. We also experimented with other traffic load and time synchronization frequencies, and we will present the impact of traffic load and clock drift in Section VI-C. We first evaluate the accuracy of interferer detection algorithm of PIM (see Section IV-C). We observed that all 40 motes correctly detected no less than 85% of interferers while more than 30 motes correctly detected at least 95% of interferers. This result clearly demonstrates the effectiveness of passive interference detection. We plot models generated by different methods for a typical node in our outdoor testbed in Fig. 5. It can be seen that the model generated from PIM perfectly matches those generated from the ACTIVE with 4096 rounds. In contrast, the ACTIVEs with fewer rounds exhibit considerable variations. In the following, we use the models generated by ACTIVE with 4096 rounds as "ground truth" for evaluating the accuracy of PIM under different settings.

Fig. 6 shows the errors of throughput prediction using the models generated by different methods. In the experiment, PIM generates the PRR-SINR model of a chosen node using the statistics collected in 5 minutes. The model is then used to predict the throughput of the link from another node to

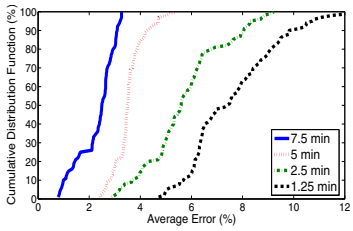


Fig. 7. The CDF of average errors over time.

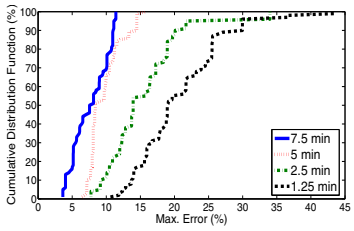


Fig. 8. The CDF of max. errors over time.

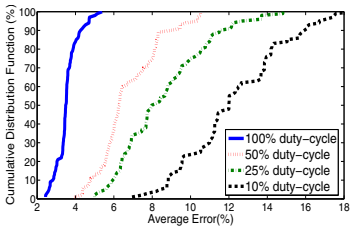


Fig. 9. The CDF of errors w/ duty cycles.

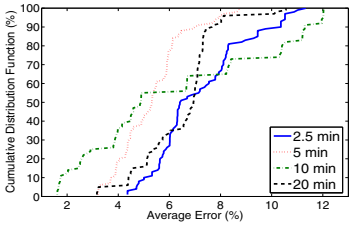


Fig. 10. The CDF of errors without time synchronization.

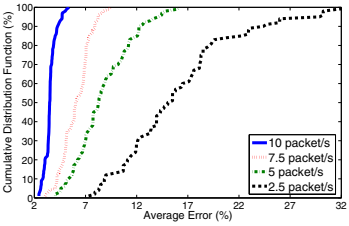


Fig. 11. The CDF of errors with different traffic workload.

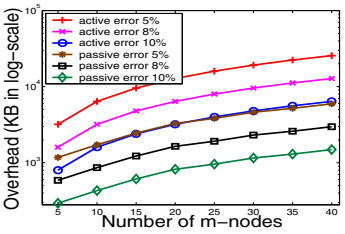


Fig. 12. Messaging overhead of PIM and the active method.

this node in the next 5 minutes. The transmit power of sender is varied to create different SINRs at the receiver. The absolute errors between predicted and measured throughputs are plotted. The model building time of ACTIVE varies with the number of rounds used. For comparison, we also plot the prediction error of the analytical PRR-SINR model that is currently implemented in TinyOS (see Section III). From Fig. 6, we see that the model built by PIM is very accurate in predicting link throughput and outperforms ACTIVE using 1024 rounds. In contrast, the analytical model in TinyOS yields significant prediction errors. This is because its parameters are only optimized for specific settings and cannot be adapted based on in-situ measurements.

The accuracy of generated models in PIM depends on the number of samples collected. Therefore, one question is whether the modeling process of PIM can converge quickly enough to be useful. We now analyze the accuracy of PIM models as data collection process evolves in NetEye. In the experiments, PIM periodically derives a PRR-SINR model for every node in the network and computes the error as the absolute difference from the model generated by the ACTIVE that uses 4096 rounds. Specifically, for each SINR value in the transitional region, we compute the absolution error between the PRRs of two models. The errors are averaged over all SINR values as the average error between two models. Fig. 7 and 8 show the average and largest errors at different instants in time, respectively. We see that the modeling process of PIM converges quickly. For instance, the average error has decreased to be less than 10% after 1.25 minutes of passive measurement, and the largest modeling error has also decreased to be less than 10% with high probability after 5 minutes of passive measurement.

C. Impact of duty cycling, clock drift, and traffic load

In this section, we evaluate the impact of several important factors on the performance of PIM, which include duty cycling, clock drift, and traffic load. Duty cycling protocols are widely adopted by WSNs to reduce the power consumption of idle radios. However, they can significantly reduce the amount of measurement statistics gathered by PIM and hence affect the modeling accuracy. Fig. 9 shows the CDF of average errors of the models generated by PIM when nodes operate in synchronous duty cycles. It can be seen that the errors drop quickly when the duty cycle increases because more samples are used for model generation. For instance, for a duty cycle of only 10%, the error falls below 13% in most of the time. We note that the error can be reduced by increasing the duration of measurement too, which is set to be only 5 minutes in this experiment.

PIM requires timing information of packets to generate models. We now evaluate the impact of clock drift on the modeling accuracy of PIM. Fig. 10 shows the CDF of errors for different measurement durations without time synchronization except at the beginning of the measurement. It can be seen that the modeling error falls below 10% in most of time. In particular, 50% of the errors are less than 5% when the clocks of nodes are not synchronized for 20 minutes. We also observed that the error increases drastically when the duration is longer than 20 minutes. This is because the clock drift becomes larger than the packet transmission time when the interval of time synchronization is longer than 20 minutes. As a result, PIM cannot correctly correlate packet transmissions in the computation of SINRs. Interestingly, as shown in Fig. 10, the error does not decrease monotonically with the increase of measurement duration. This is because more samples can be accumulated for a longer duration resulting in more accurate model estimation. Therefore, the modeling error caused by clock drift can be compensated by increasing the number of samples. The overall result in this experiment shows that PIM only requires coarse-grained time synchronization and is robust to clock drift.

Fig. 11 shows the CDF of errors under different traffic loads

in a duration of 5 minutes.. A low traffic load not only leads to less measurement statistics but also reduces the probability of packet collisions. Both factors may affect the accuracy of PIM. It can be seen from Fig. 11, when each node sends a rate of 2 packets/s, about 50% errors fall below 15%. The error drops quickly when the rate is above 5 packets/s. The results is this section can be used by network designer to minimize the overhead of PIM by tuning the measurement duration.

D. Overhead of PIM

The overhead of PIM lies in choosing m-nodes and delivering the measured information to the sink for PRR-SINR model generation. We comparatively study the overhead of PIM and ACTIVE for achieving different levels of modeling accuracy. Fig. 12 shows the overhead measured as the number of bits transmitted (in log scale) incurred for ensuring a modeling error of no more than 5, 8, and 10%, respectively. We can see that PIM incurs significantly lower overhead. For instance, with the overhead of ensuring an error of less than 10% in ACTIVE, PIM can ensure an error of less than 5%. Finally, we plot the number of r-nodes needed for building PRR-SINR models in Fig. 13. We compare our greedy algorithm with the optimal exhaustive search, and a simple algorithm that randomly picks new r-node sets until they cover all the PRR-SINR measurement points. When the models of 20 nodes are measured, the exhaustive search takes about 30 minutes to complete on a 1.6 GHz duo-CPU PC (served as the base station) while the greedy algorithm completes within a minute. Thus our algorithm significantly reduces the computational overhead of the exhaustive search, which is important for adapting r-node selection in real-time. Fig. 13 also shows that PIM significantly outperforms the random algorithm, and achieves a close-to-optimal performance when the number of m-nodes is small.

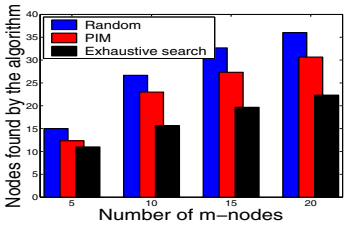


Fig. 13. Number of r-nodes needed for model generation.

VII. CONCLUSION

Interference modeling is crucial for the performance of numerous wireless protocols. In this paper, we propose the passive interference measurement (PIM) approach to tackle the complexity of accurate physical interference characterization. PIM employs a novel method for detecting the interferers of a node and a greedy algorithm for reducing the number of nodes used for model measurement. We implemented PIM in TinyOS-2.0.2 and conducted extensive experiments on both a 13-node and a 40-node testbeds of TelosB motes. Our experimental results show that PIM can achieve high accuracy of PRR-SINR modeling with significantly lower overhead than an active measurement method.

VIII. ACKNOWLEDGEMENT

This work is supported, in part, by the National Science Foundation under grant CNS 0916576 and National Grand Fundamental Research 973 Program of China under Grant No.2006CB303006.

REFERENCES

[1] Networked embedded sensing testbed: http://neteye.cs.wayne.edu.
[2] TelosB sensor node. http://www.crossbow.com/.
[3] TinyOS collection tree protocol (CTP). http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html.
[4] TinyOS TOSSIM. http://www.tinyos.net/tinyos-2.x/tos/lib/tossim/.
[5] 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.
[6] A. Cerpa, J. L. Wong, L. Kuang, M. Potkonjak, and D. Estrin. Statistical model of lossy links in wireless sensor networks. In *IPSN*, 2005.
[7] O. Goussevskaia, Y. A. Oswald, and R. Wattenhofer. Complexity in geometric sinr. In *MobiHoc*, 2007.
[8] IEEE. Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans). In *IEEE Standard 15.4*, 2003.
[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. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic. ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks. In *ACM SenSys*, 2006.
[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] T. Moscibroda and R. Wattenhofer. The complexity of connectivity in wireless networks. In *IEEE INFOCOM*, 2006.
[13] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan. A general model of wireless interference. In *MobiCom*, 2007.
[14] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. In *SIGCOMM*, 2006.
[15] S. Rangwala, A. Jindal, K.-Y. Jang, K. Psounis, and R. Govindan. Understanding congestion control in multi-hop wireless mesh networks. In *MobiCom*, 2008.
[16] 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.
[17] 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.
[18] D. Son, J. Heidemann, and B. Krishnamachari. Towards concurrent communication in wireless networks. Technical Report ISI-TR-646, Information Sciences Institute, 2007.
[19] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of concurrent transmission in wireless sensor networks. In *Sensys*, 2006.
[20] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys*, 2003.
[21] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *SenSys*, 2004.
[22] J. Zhao and R. Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *Sensys*, 2003.
[23] G. Zhou, T. He, J. A. Stankovic, and T. F. Abdelzaher. Rid: radio interference detection in wireless sensor networks. In *Infocom*, 2005.
[24] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. *SECON*, 2004.