

# Robust Topology Control for Indoor Wireless Sensor Networks

Gregory Hackmann, Octav Chipara, Chenyang Lu  
Department of Computer Science and Engineering  
Washington University in St. Louis, USA  
{gwh2, ochipara, lu}@cse.wustl.edu

## ABSTRACT

Topology control can reduce power consumption and channel contention in wireless sensor networks by adjusting the transmission power. However, topology control for wireless sensor networks faces significant challenges, especially in indoor environments where wireless characteristics are extremely complex and dynamic. We first provide insights on the design of robust topology control schemes based on an empirical study in an office building. For example, our analysis shows that Received Signal Strength Indicator and Link Quality Indicator are *not always* robust indicators of Packet Reception Rate in indoor environments due to significant multi-path effects. We then present *Adaptive and Robust Topology control (ART)*, a novel and practical topology control algorithm with several salient features: (1) ART is robust in indoor environments as it does not rely on simplifying assumptions about the wireless properties; (2) ART can adapt to variations in both link quality and contention; (3) ART introduces zero communication overhead for applications which already use acknowledgements. We have implemented ART as a topology layer in TinyOS 2.x. Our topology layer only adds 12 bytes of RAM per neighbor and 1.5 kilobytes of ROM, and requires minimal changes to upper-layer routing protocols. The advantages of ART have been demonstrated through empirical results on a 28-node indoor testbed.

## Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design—*Network topology*; C.2.2 [Computer-Communication Networks]: Network Protocols

## General Terms

Algorithms, Measurement, Performance

## Keywords

Power Management, Topology Control

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SenSys'08, November 5–7, 2008, Raleigh, North Carolina, USA.  
Copyright 2008 ACM 978-1-59593-990-6/08/11 ...\$5.00.

## 1. INTRODUCTION

Topology control can reduce the power consumption of packet transmissions and channel contention by adjusting the transmission power of each node in a wireless sensor network. Topology control for wireless sensor networks faces significant challenges, especially in indoor environments where wireless characteristics are particularly dynamic and complex. Wireless links have highly irregular and probabilistic properties. Furthermore, link quality can vary significantly over time, especially in indoor environments due to human activity and multi-path effects. Topology control algorithms must therefore deal with changes in link quality at different power levels using online measurements. To facilitate efficient link profiling, commodity radios [1, 2] commonly provide instantaneous link quality indicators such as the Receiver Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI). Unfortunately, the correlation of these indicators with link quality (i.e., packet reception rate (PRR)) is highly sensitive to the environment. For example, RSSI is shown to have a good correlation to PRR in some outdoor and indoor environments [3, 4], while other studies (including our own) indicate the opposite [5]. A practical topology control algorithm must therefore be robust against environmental and workload changes while introducing minimal communication, processing, and memory overhead.

Specifically, this paper makes the following contributions.

- We present an extensive empirical study performed in an office building and provide key insights on the design of robust topology control algorithms in such environments. In particular, we analyze the correlation between RSSI/LQI and PRR and show that RSSI and LQI are not *always* robust indicators of PRR in indoor environments with significant multi-path effects.
- We present the *Adaptive and Robust Topology control (ART)* algorithm, a practical topology control protocol algorithm designed for complex and dynamic environments. ART has the following salient features. (1) ART is designed to be robust; we made a conscientious effort to minimize our design assumptions and to validate each assumption through empirical studies. An important design decision was not to rely on indirect indicators of link quality, such as RSSI or LQI, because we found them insufficiently robust in indoor environments. (2) ART is an adaptive topology control protocol that adapts the transmission power in response to variations in link quality triggered by either environmental changes (e.g., changes in signal or

noise levels) or varying degrees of network contention. (3) ART is efficient, introducing zero communication overhead for applications which already use packet acknowledgements.

- We have implemented ART in a topology layer between the MAC and routing layers in TinyOS 2.x. Our topology control layer only adds 392 bytes of RAM and 1.5 kilobytes of ROM and requires minimal changes to upper-layer routing protocols.
- We assess the performance of ART on a 28-node indoor testbed through micro- and macrobenchmarks. The microbenchmarks demonstrate that ART can lower the energy consumption of individual links by an average of 15% with no loss in link quality. The macrobenchmarks show that ART’s performance meets that of a representative topology control algorithm, PCBL, without introducing PCBL’s bootstrapping cost. We also show that ART can improve energy efficiency by up to 40% under heavy contention.

The remainder of the paper is organized as follows. In Section 2, we describe prior empirical link quality studies and several existing topology control schemes. In Section 3, we discuss the findings of our empirical link study. Section 4 describes the ART algorithm in detail. We discuss our implementation of ART in Section 5 and provide an empirical analysis of ART’s performance in Section 6. Finally, we conclude in Section 7.

## 2. RELATED WORK

In this section, we describe existing studies into the impact of various environmental and spatial factors on the performance of wireless sensor network links. We also discuss state-of-the-art topology control algorithms that attempt to select the proper power setting for wireless links in the face of the complex behavior observed in these studies.

### 2.1 Empirical Link Studies

A significant number of existing link quality studies [3,6–9] evaluate link performance at a single, fixed power setting. More closely related to our work are empirical link studies which explore the properties of wireless links at varying power levels. [10, 11] observe that radio ranges are highly irregular and do not fit circular radio models in practice. [5] finds a temporal impact on link quality with the Chipcon CC1000 radio [1] in an indoor testbed environment and notes that some middling-quality links can be converted to good-quality links with small changes in transmission power. [4] notes a strong correlation between RSSI and PRR on the Chipcon CC2420 radio [2], independent of time or transmission power, in three different environments; it also shows a strong correlation between transmission power and RSSI which varies across links and across time.

Our own link quality study builds on these existing studies and is complementary to this work. We perform our experiments on a complex indoor testbed of motes equipped with 802.15.4-compliant Chipcon CC2420 radios; previous studies consider older, proprietary radios [5, 10, 11] or were carried out in a simplified indoor environment [4]. Our experiments also provide new insight into several areas of interest to topology control algorithms, such as the impact of transmission power on contention; whether link indicators

are robust indoors; and whether links which are high-quality over the short term remain high-quality over the long term.

## 2.2 Topology Control Schemes

We discuss here a number of existing topology control algorithms. We will begin by examining a class of theoretical algorithms that have been evaluated only in software simulations. We will then describe two state-of-the-art topology control algorithms, which are based on extensive link quality studies and have been deployed on real sensor networks. Throughout this section, we highlight several key assumptions made by these algorithms; we will evaluate the robustness of these assumptions using our own link quality data in the following section.

### 2.2.1 Theoretical Model-Based Algorithms

Traditionally, topology control schemes have been built on simplified theoretical radio models (such as graph-based connectivity) and tested in simulation environments. However, these traditional topology control schemes are not always appropriate for wireless sensor networks. These topology schemes make simplifying assumption such as circular radio range [12,13] or uniform node distribution [14] which are unrealistic in many wireless sensor network applications. Moreover, a typical wireless sensor node has limited computational power and storage capacity, which mandate topology control algorithms with low processor overhead and memory consumption.

A recent trend in theory-based topology control is to quantify the interference of a network graph and explicitly consider this metric when selecting a network topology. Theoretical models that consider interference represent a more accurate view of real-world wireless sensor network environments than those that do not. Nevertheless, existing interference-based algorithms still make unrealistic simplifying assumptions such as circular radio range [15], global knowledge [16], or the ability to perform complex computations online [17]. These assumptions limit the applicability of these algorithms for real sensor network deployments.

LMST [13] is a representative theoretical model-based algorithm which is specifically designed to fit the communication and computational constraints of wireless sensor networks. LMST computes a reduced-power network topology by constructing a minimum spanning tree over the network in a fully-distributed fashion. The transmission power of each link in this topology is shrunk according to the observed path loss. When computing the network’s MST, LMST assigns a cost to each link proportional to its physical length. However, existing studies have found a much more complex relationship between link length and link quality [5, 10, 11]. LMST also requires that all nodes know the physical distance to their neighbors in order to operate in a truly decentralized fashion.

### 2.2.2 PCBL

PCBL [5] is one of the first topology control algorithms to be deployed in a real-world sensor network testbed. Using extensive empirical data collected from an indoor testbed of PC104 motes, the authors observe that high-quality links also tend to be highly stable. Specifically, they note that links with a PRR above 98% during a seven-day experiment had a standard deviation of 2.2%, while links with a PRR above 90% had a standard deviation of 19.8%. Based on

these observations, the authors proposed that PCBL maintain two separate bounds on link quality. Links which fall below some the lower PRR bound at all power levels are considered unreliable and blacklisted (i.e., never used for transmissions). Links which achieve the upper PRR bound at some power setting are considered highly reliable, and their power is shrunk to the lowest such setting. Finally, links which fall between these two extremes are considered reliable enough to use, but only at maximum power.

It is impractically expensive to maintain comprehensive PRR data over all power levels at runtime, since this would require PCBL to continuously probe a node’s neighbors at all power levels. Instead, the basic PCBL algorithm approximates this behavior by extensively probing links once at all power settings and then freezing this link quality data. The intuition behind this approximation is that high-quality links are also highly stable, so PCBL will inherently favor links which are resilient to changes in network conditions.

[5] suggests that PCBL’s runtime overhead could be lowered by initially collecting link statistics at maximum power, allowing the routing layer to bootstrap and select its neighbors, and then only tuning the transmission power over those links actually used for routing. This lowered overhead would allow PCBL to bootstrap more frequently and hence be even more resilient against link quality fluctuations. However, this approach would conflict with highly-dynamic routing engines like TinyOS’s Collection Tree Protocol [18] which continuously send beacons during the application’s lifetime to discover less-expensive routes.

### 2.2.3 ATPC

ATPC [4] is designed to avoid costly link probing by using an instantaneous link quality metric as a proxy for PRR. The authors gathered RSSI, LQI, and PRR statistics in three different environments: a parking lot, a grassy field, and an office building. They discovered a strong correlation between RSSI and PRR (and between LQI and PRR) along a monotonically-increasing curve. The shape of this curve varied for each environment but was consistent across all links, power settings, and times within a given environment. Once the authors collected enough data offline to construct this curve, they were able to convert a lower-bound on PRR into a corresponding lower bound on RSSI or LQI.

The authors also noted a linear correlation between transmission power level and RSSI/LQI readings at the receiver. Unlike the RSSI-to-PRR curve, this parameters of this line varied across links and over time. ATPC estimates the slope and Y-intercept of this line at runtime for each link and dynamically adjusts this model using a closed feedback loop. Using this model, ATPC selects the proper transmission power to achieve the necessary lower bound on RSSI and, by proxy, the lower bound on PRR.

## 3. EMPIRICAL STUDY

In this section, we present an empirical study which considers four questions at the core of topology control algorithms: (1) is topology control beneficial?; (2) what is the impact of transmission power on contention?; (3) is it necessary to dynamically adapt transmission power online?; and (4) are instantaneous indicators of link quality (such as RSSI and LQI) robust in indoor environments? In answering these questions, we provide guidelines for the development

of topology control algorithms, which we apply to the design of our own ART algorithm in Section 4.

The results presented in this section are complementary to the empirical power control studies presented in [4,5]. The empirical results presented in [5] were obtained using the CC1000 radio platform; our experiments are performed on Chipcon CC2420 radios, which comply to the IEEE 802.15.4 physical layer specification. The two radio platforms are significantly different [1,2]: they operate in different frequency domains and use different modulation schemes. The empirical results in [4] were also obtained using the CC2420 radio; however, they considered only simple network layouts where nodes have line-of-sight communication. In contrast, our deployment spans a floor of an entire building. A significant number of links in our deployment are formed between nodes without line-of-sight.

The empirical study we present validates many of the findings of previous studies. Moreover, we also provide important new insight into the impact of transmission power on contention. In particular, we found that some assumptions underlying the design of PCBL and ATPC do not hold on our testbed. These discrepancies are caused by the different radio platforms and environments in which the experiments were performed. Our observations served as the foundation for developing a topology control algorithm that is *robust* in different indoor environments.

### 3.1 Experimental Setup

All experiments are performed on a testbed consisting of 28 TelosB motes equipped with CC2420 radios using TinyOS 2.x’s default CSMA/CA MAC layer. Each node is connected to a central server using a wired USB and Ethernet backbone. This backbone is used as a back-channel to issue commands to the motes and collect experimental results without interfering with ongoing wireless transmissions.

The CC2420 radio chip can be programmed to operate at 8 different power levels<sup>1</sup> with output power ranging from -25 dBm to 0 dBm and current consumption ranging from 8.5 mA to 17.4 mA [2]. The CC2420 radio also provides an RSSI reading and LQI reading embedded in the metadata of all incoming packets. The RSSI reading represents a sampling of the signal strength (transmission + background noise) taken at the beginning of the packet reception, while the LQI reading represents the average symbol correlation value over the packet’s first eight symbols. In order to estimate background noise, applications may sample the CC2420’s RSSI register when the radio is idle. The CC2420 may be programmed to operate on different frequencies; all experiments here are performed on a channel that does not overlap with the building’s 802.11g network.

### 3.2 Is Topology Control Beneficial?

Topology control is an attractive mechanism because it can simultaneously improve energy efficiency and network performance. Here, we evaluate these potential benefits of topology control in our indoor testbed.

To isolate the impact of topology control on link quality and energy savings, we performed an experiment in which there is no network contention. Each node broadcasts 50 packets while its neighbors record the sequence number,

<sup>1</sup>Though the CC2420’s transmission power register can be set between 0–31, the CC2420 datasheet only defines output power information for 8 of these 32 settings.

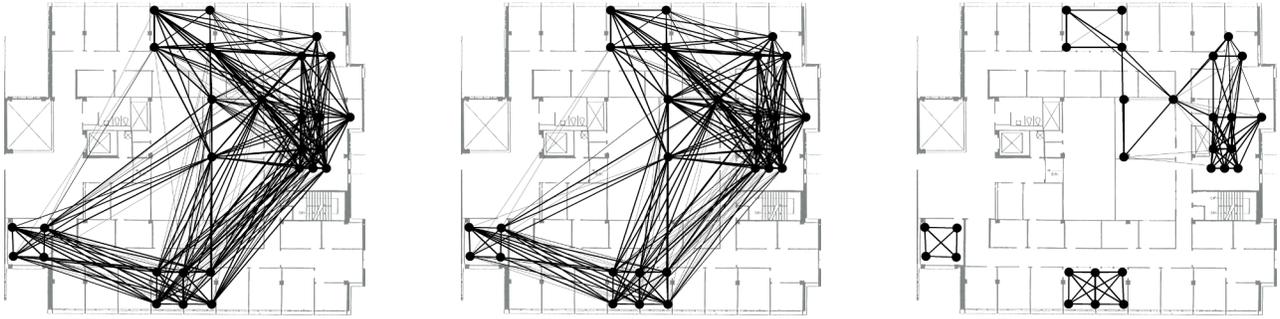


Figure 1: The testbed topology when transmitting at 0 dBm, -5 dBm, and -25 dBm

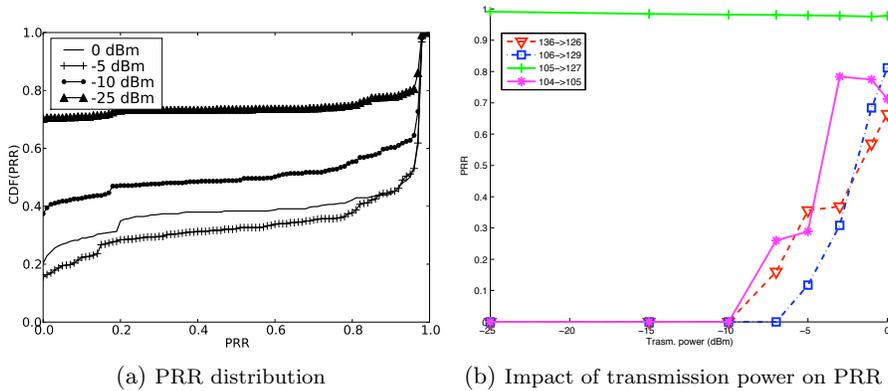


Figure 2: A significant fraction of poor quality links may be transformed into high quality links through topology control

RSSI reading, and LQI reading of all packets that they receive. The node repeats this procedure at each of the CC2420’s eight discrete power levels for a total of 400 packets. After a node completes sending its 400 packets, the next sending node is selected in a round-robin fashion. This cycle repeated for 24 hours, giving each node 49 rounds to transmit 400 packets each.

During the course of the experiment, we recorded at least one successful packet transmission on 524 of the network’s 756 possible unidirectional links. Figure 1 shows the network topology during this experiment at three different power levels, where line thickness is proportional to the packet reception ratio (PRR). We see that the topology at maximum power (0 dBm) and medium power (-5 dBm) are fairly similar, both in terms of connectivity and link quality. In contrast, the minimum power (-25 dBm) topology has partitioned the network into clusters of mostly high-quality links. These figures highlight the potential benefits of assigning non-uniform transmission powers to different nodes: it is sufficient to transmit at -25 dBm to reach nearby nodes, but other links require higher transmission powers to achieve connectivity. This confirms experiments carried out in [5] and [4] showing that uniform power settings across the network lead to non-uniform behavior.

To better understand the benefits of tuning the transmission power, we computed the PRR of each link during the entire benchmark run; the CDF of link PRRs is shown

Figure 2(a). Changing the transmission power can affect a large fraction of the links in the testbed: for example, 368 links (70.2%) have a PRR of 0 at -25 dBm, compared to 82 (15.6%) at -5 dBm. We see a similarly dramatic effect in Figure 2(b), where three links selected from our testbed go from unusable at -10 dBm to medium quality at -5 dBm. This confirms the results in [5] which indicate that transmission power can transform a significant number of poor quality links into good quality links. We also note that a slightly higher proportion of links have poor link quality at 0 dBm than at -5 dBm in Figure 2(a). A handful of nodes performed worse when transmitting at 0 dBm than at lower power settings, which we believe is caused by multipath effects that are more pronounced at maximum power. This phenomenon may also be seen in Figure 2(b) for link 104 → 105.

Similarly, reducing a link’s transmission power can result in significant energy savings. To quantify these savings, we inspected the traces collected during the previous experiment. For each link, we computed the PRR for each round using the max-power data, and then selected the lowest power level for each round that had no degradation in PRR compared to max-power (i.e., the power setting that a topology control algorithm with perfect knowledge *would have* picked). When the maximum transmission power is used, a node would draw 17.4 mA, compared to an average current draw of 11.4 mA under an ideal power assignment.

*Insight 1: Transmission power should be set on a per-link basis to improve link quality and save energy.*

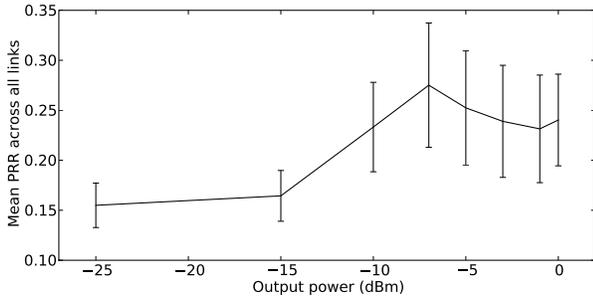


Figure 3: Effect of transmission power on contention

### 3.3 What is the Impact of Transmission Power on Contention?

While increasing transmission power improves the quality of *individual* links, its also may result in increased contention. To understand the impact of transmission power on contention, we performed the following experiment. We select ten links at random from our testbed and have them simultaneously transmit packets as fast as possible for 24 hours. Each node transmits 200 packets, after which the power level is changed. Figure 3 plots the average PRR over all links as transmission power is increased from -25 dBm to 0 dBm. Increasing the transmission power up to -7 dBm has a positive effect on link quality; however, increasing the power further results in decreased PRR. This happens because the benefits of increased transmission power are offset by higher contention, increasing packet collisions and decreasing throughput. The data indicates that power control is an effective mechanisms for controlling the degree of network contention. Moreover, robust topology control cannot be performed without accounting for this link quality/contention tradeoff.

*Insight 2: Robust topology control algorithms must avoid increasing contention under heavy network load.*

### 3.4 Is Dynamic Power Adaptation Necessary?

One important consideration for topology control protocol design is the rate at which their power decisions need to be re-evaluated. If the rate of change in link quality is sufficiently low, then it is feasible to make infrequent decisions that incur high communication or computational overhead. To address this question, we ran a long-term experiment on a few links to determine the time-scale of link variations. The setup for this experiment is identical to that in Section 3.2, except it was carried out over only 3 links with 100 packets per power level per round. By reducing the number of links profiled and increasing the number of packets per link, we are able to obtain a fine-grained view of how link quality varies over time. We remind the reader that we select a radio frequency for these experiments with little background noise. However, we also sample 50 signal strength readings in the beginning of each round when no node is transmitting to validate that the background noise does not vary significantly over the duration of our experiment.

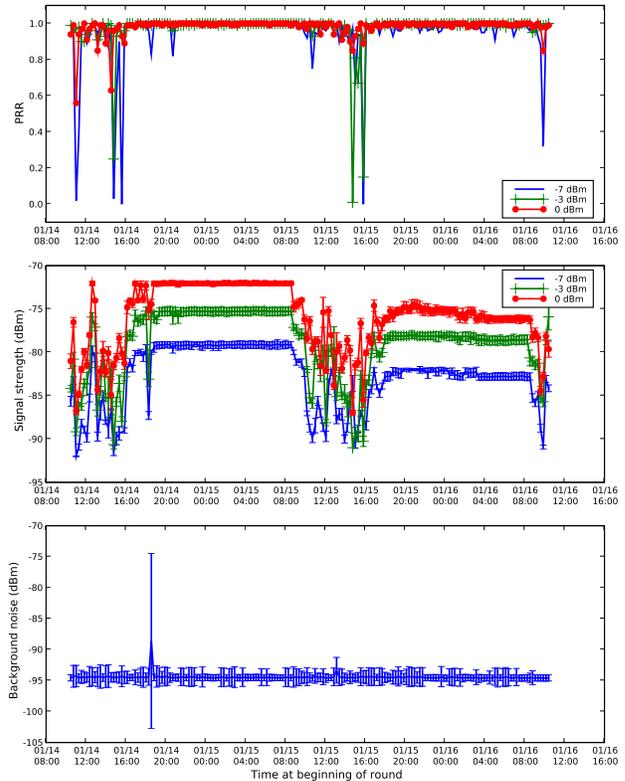


Figure 4: The PRR, mean signal strength, and background noise collected over link 110  $\rightarrow$  139

Figure 4 shows the PRR, RSSI, and background noise for one of the links that we sampled; the other measured links had similar results. Figure 4 indicates a correlation between RSSI and human activity: during work hours, from 8:00 to 18:00, there is a significant reduction in RSSI along with an increase in its variation; in contrast, during the night, the link is significantly more stable.

A similar correlation may be observed for PRR, which is more pronounced for lower transmission powers. This correlation is expected because even small noise variations may cause packets transmitted at lower power levels to be corrupted. The trace shows that in order to maximize energy savings, the transmission power must be tuned dynamically based on environmental conditions.

Apart from one outlying data point, the background noise on our wireless channel is stable during the entire benchmark run, suggesting that increased activity on the wireless spectrum during the workday was not responsible for this cyclical link quality fluctuation. We believe that people walking around the building during the daytime attenuate the signals to varying degrees, which causes sharp variations in link quality. Similar results were observed on the CC1000 radio [5]. These results indicate that topology control schemes must frequently adapt link transmission power over time in order to avoid significant variations in link quality.

An important characteristic of high-quality links is that they have high PRR and low standard deviation. This assumption has been validated on long-term experiments on the CC1000 radio [5, 19]. Protocol developers use this as-

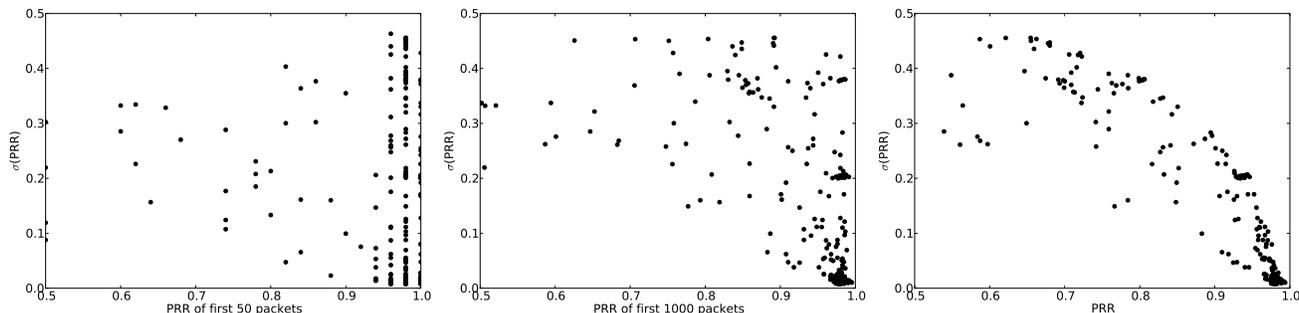


Figure 5: The relationship between overall PRR and standard deviation in PRR

sumption to argue in favor of performing a bootstrapping phase in which such high quality links are identified. Unfortunately, we found that we cannot establish a strong correlation between PRR and standard deviation without collecting an impractically large amount of PRR data. Using the data collected in Section 3.2, we plot the relationship between each link’s overall standard deviation in PRR (i.e., its actual stability over the entire benchmark) against the PRR calculated by looking at the first data round (50 packets/link), 12 hours’ worth of link data (1000 packets/link), and the entire benchmark dataset (1950 packets/link), as shown in Figure 5. We observe that links with an overall PRR of 98% or higher during our full 24-hour dataset indeed have low standard deviation (right). However, we observe a 6.8-fold increase in standard deviation if we look at links with 98% PRR within our first round of data (left). Even selecting links with  $\geq 98\%$  PRR over 12 hours’ of data (center) would result in a 3.2-fold increase in standard deviation compared to the full dataset. This indicates that, in some environments, even many hours’ worth of bootstrapping data is insufficient to properly predict a link’s long-term behavior. PCBL [5] suffers from this design pitfall, because a short bootstrapping phase is used to predict the long-term link quality.

*Insight 3: Robust topology control algorithms must adapt their transmission power in order to maintain good link quality and save energy.*

### 3.5 Are Link Indicators Robust Indoors?

Commodity radios generally provide per-packet link quality indicators, such as RSSI and LQI. Because these metrics are instantaneous and inexpensive to collect, they are an attractive proxy for more expensive link quality indicators such as PRR. The relative benefits of these metrics are still a subject of debate in sensor network community: for example, [3] advocates measuring link quality using RSSI over LQI, while [4] reports that both RSSI and LQI are good indicators of link quality. In this section, we present an empirical study designed to understand if either RSSI or LQI are *always* good indicators of link quality in complex indoor environments. To this end, we transmitted 50 packets over each pairwise link in our testbed at maximum power. Each time a packet was received, the recipient logged the packet’s sequence number, RSSI reading, and LQI reading. By restricting our experiments to a single power level, we were able to collect 672 rounds of data over a period of 32 hours.

To evaluate the quality of RSSI and LQI as link estimators, we attempted to find a correlation between RSSI/LQI and PRR. If there exists some critical RSSI or LQI threshold which separates “good” (high PRR) links from “bad” (low PRR) links, then these inexpensive metrics could be used as good indicators of PRR. We attempted to find these critical RSSI and LQI thresholds in our collected data, using several different PRR thresholds to define “good” links. These critical thresholds should represent the best compromise between false positives (i.e., links above the LQI/RSSI threshold but below the PRR threshold) and false negatives (i.e., links above the PRR threshold but below the LQI/RSSI threshold). For the purpose of brevity, we present here the results for two links with PRR thresholds of 80% and 90%.

Figure 6(a) shows the results for link  $106 \rightarrow 129$ . Each point in the scatter plots represents the relationship between mean RSSI/LQI and PRR for one 50-packet round. The graphs seem to indicate a correlation between the instantaneous link estimators and PRR. When setting the PRR threshold to 90%, we find the best trade-off between false positive and false negative rates when LQI = 76. At this threshold, the false positive and false negative rates are at 18% and 16% respectively. We see similar results for RSSI, with a false positive rate of 6% and false negative rate of 8% at the critical RSSI threshold of 86 dBm. The results when reducing the PRR threshold to 80% are similar. These results indicate that both LQI and RSSI are good indicators of this link’s quality.

Figure 6(b) shows the results for link  $104 \rightarrow 105$ . The scatter graphs indicate that the correlation between link indicators and PRR is worse than that observed on the previous link. Indeed, when setting the PRR threshold to 90%, the optimal LQI threshold of 70 has a false positive rate of 30% and a false negative rate of 36%. RSSI performs even more erratically, with the optimal threshold being at either -85 dBm or -84 dBm. At an RSSI threshold of -85 dBm, the false positive rate is 4% while the false negative rate is 62%; increasing the RSSI threshold to -84 dBm causes these rates to jump to 66% and 6% respectively. This sharp transition indicates that RSSI would be an unstable estimator of this link’s quality, while LQI would have a significant fraction of false negatives and positives. We observe a similar behavior with a PRR threshold of 80%.

This set of experiments demonstrate that, although there are links for which LQI and RSSI are good link quality indicators, there are others for which they are both poor indi-

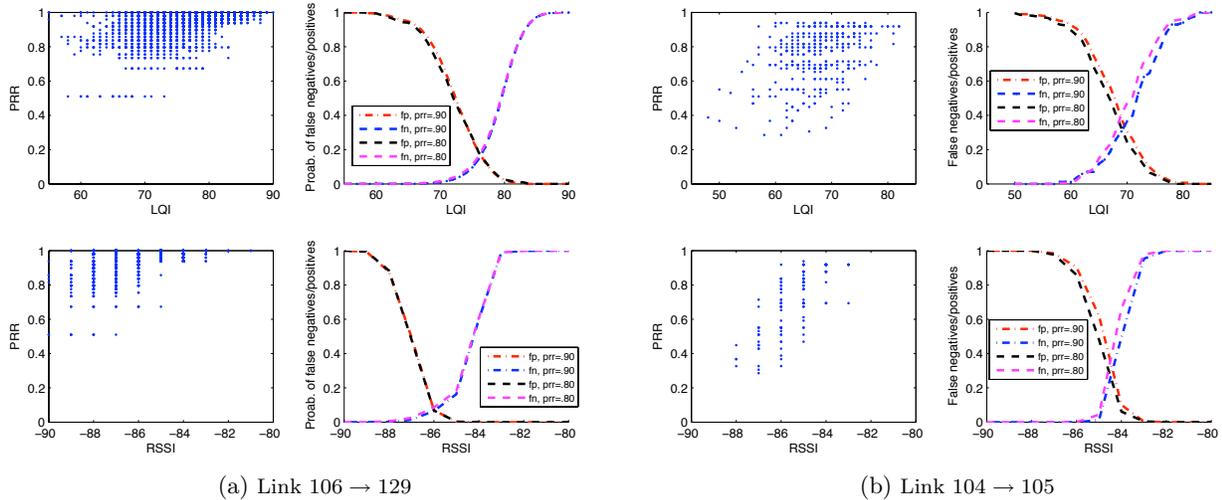


Figure 6: Quality of RSSI and LQI as instantaneous indicators of link quality

caters. Accordingly, neither LQI nor RSSI may be used for developing *robust* topology control algorithms.

ATPC [4] relies on RSSI as an indicator of link quality. Since RSSI is not *always* a good indicator of high quality links, we do not expect ATPC to be sufficiently robust for operating in all indoor environments.

**Insight 4:** *Instantaneous LQI and RSSI are not robust estimators of link quality in all environments.*

## 4. THE ART ALGORITHM

This section presents the design of a new topology control algorithm, *Adaptive and Robust Topology control (ART)*. ART is designed based on the key observations in our empirical study and has the following salient features. (1) ART is designed to be a *robust* topology control algorithm: we chose not to use indirect measurements of link quality because they are not sufficiently robust in different indoor environments. (2) ART is an *adaptive* algorithm in that it changes the transmission power of a link based on its observed PRR. Moreover, ART employs a lightweight adaptation mechanism and does not employ prolonged bootstrapping phase for link profiling. (3) ART can dynamically adapt the transmission power in response to high channel contention. (4) ART is specifically designed to be *efficient*, so that it can be realistically deployed on memory-constrained wireless sensor platforms with low runtime overhead.

### 4.1 ART Algorithm Description

ART individually tunes the transmission power over each of a node’s outgoing links. A link is initially set to transmit at its maximum power. ART monitors all outgoing packet transmissions and keeps a record of whether each transmission failed or succeeded in a sliding window of size  $w$ . While the window is filling, we say that the link is “initializing”. When the sliding window is full, ART compares the number of recorded transmission failures to two thresholds  $d$  and  $d'$ , where  $d > d'$ . The link remains in this “steady” state as long as the number of failures is between these two thresholds.

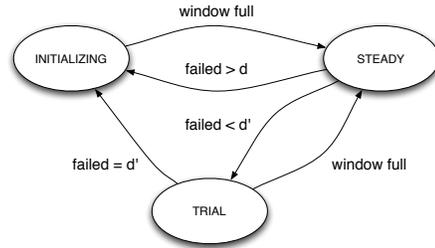


Figure 7: ART state transition diagram

If the recorded number of failures is above  $d$ , then ART adjusts the link power to improve its quality. ART may raise the link’s transmission power to improve its quality under low contention, but may lower its transmission power under high contention. As detailed in Section 4.3, ART uses a simple gradient-based mechanism to detect high contention based on recent history. After selecting a new power setting, ART will flush the transmission window and re-enter the initializing state.

If the number of failures is below  $d'$ , then ART will enter a “trial” state where it temporarily lowers the power by one level. If the link experiences  $d'$  more transmission failures at any time while in the trial state, then it returns to the previous power level, flushes its transmission window, and goes back to the initializing state. If the link’s window fills with fewer than  $d'$  recorded transmission failures, then the new power setting is made permanent and the link goes back to the steady state.

This algorithm has several salient features worth emphasizing. First, it makes its decisions by monitoring traffic already being generated by the upper layers, and therefore introduces no communication overhead apart from packet acknowledgements<sup>2</sup>. Second, the largest component of its

<sup>2</sup>Because packet acknowledgements are needed for reliable data transmission and part of many routing protocols, in

memory overhead is a sliding window of one bit per entry, which in practice can be as small as 20 to 50 entries per link. Third, ART makes no computationally-complex decisions, and hence can be implemented with minimal ROM and CPU overhead. Finally, ART does not always assume that increasing transmission power will improve link quality, since this is not always the case under high contention and interference (see Section 3.3).

Figure 7 summarizes the ART algorithm as a state diagram. We will now discuss in further detail the intuition behind this algorithm.

## 4.2 Link Quality Thresholds

The most straightforward indicator of a link’s quality is its PRR. ART is therefore designed to lower the link’s power while still maintaining an application-specified target PRR. Consider an application that specifies a target PRR of  $p$ . We can inexpensively track the link’s recent behavior by creating a sliding window of  $w$  bits, where each bit represents a single transmission and indicates whether it was ACKed by the recipient. A link meets a target PRR  $p$  if there are  $d = w \cdot (1 - p)$  or fewer failed transmissions in its window at any given time. We therefore wish to tune each link’s power to the lowest power that can keep  $d$  or fewer failures in the link’s window.

However, we cannot necessarily lower a link’s transmission power each time we determine it to be within its upper-bound on failures. Because of the bimodal relationship between transmission power and PRR observed in Section 3.2, this might result in an actual PRR much lower than  $p$ . After the link changes power, it must flush its packet window to reflect the potential change in link quality. The application must therefore transmit an entire window of  $w$  packets over the link before ART can detect that its link quality has degraded. If the link’s previous power was at the critical threshold between a high-quality link and a low-quality link, there may be more than  $d$  failures at its new power state. In the worst case, the link may alternate between two power states in which  $d - 1$  transmissions fail in one window, followed by all  $w$  in the next window.

We therefore establish two policies to address this problem. First, we observe that we don’t need to fill the entire window to detect link failure. As soon as there are  $d$  delivery failures in the window, it is impossible to meet the target PRR  $p$  once the window is full. Thus, links are first moved into a “trial” state, where they transmit packets at a lower power setting but immediately return to a higher power level if too many failures are detected. Links which successfully pass the trial with a full window are moved back into the “steady” state at the reduced power.

Second, we create another PRR threshold to accommodate the potential transmission failures in this “trial” state. Specifically, we want to select a second PRR threshold  $p'$  and a corresponding failure threshold  $d' = w \cdot (1 - p')$ . We select  $p', d'$  such that a link may lose up to  $d' = w \cdot (1 - p')$  packets during a  $w$ -packet window, allowing it trial to a lower power; fail up to  $d'$  packets at the lower power level, forcing it back to its original power level; and still achieve a PRR of  $p$ . In the worst case, this is satisfied by failing  $d' = w \cdot (1 - p')$  out of  $w$  transmissions at the current power level, then failing all  $d' = w \cdot (1 - p')$  out of the next  $d'$  at a lower power;

practice ART will often receive these acknowledgements “for free”.

i.e.,  $\frac{w \cdot (1 - p') + w \cdot (1 - p')}{w + w \cdot (1 - p')} \leq 1 - p$ . This inequality holds when  $p' \leq \frac{2p}{p+1}$ , and hence we choose  $d' = \frac{2p}{p+1} \cdot w$ .

## 4.3 Handling High Contention

We now consider the case when a link fails, i.e., it falls below its PRR threshold of  $p$  by accumulating more than  $d$  transmission failures in its window. As noted in Section 3.3, increasing transmission power can in fact decrease overall link quality, due to contention and interference from other nodes. ART addresses this problem by maintaining a flag which indicates whether to increase or decrease transmission power on link failure. This flag is initially set so that ART responds to link failure by increasing its transmission power by one level.

Each link also maintains a one-element history recording the number of transmission failures in its window the last time that the link failed. When ART determines that a link has failed, it compares the current failure count against this history. If the current failure count is higher (i.e., increasing the power made things worse), then ART inverts this flag. In effect, this flag allows ART to track the “gradient” of its current position on the power/PRR curve without maintaining a full multi-window history.

Rather than devising a complex scheme for detecting congestion or coordinating the power decisions across nodes, we opted for the “gradient” solution due to its *simplicity* and *elegance*: (1) minimal state is required for maintaining the gradient; (2) there is no significant processing overhead; and (3) it introduces no additional communication overhead. Our macrobenchmarks presented in Section 6 show the effectiveness of the proposed solution.

## 4.4 Handling Broadcast Traffic

Throughout this section, we have assumed that the radio is sending traffic in a strictly unicast fashion. However, broadcast traffic is frequently used in sensor network applications for disseminating information to multiple neighbors in the node’s one-hop neighborhood. It is important to note that there are actually two distinct types of broadcast traffic as far as topology control is concerned: true broadcast data, and multicast data.

Multicast data packets are those which need to be distributed to all (or some subset) of neighbors in the node’s neighbor table. For example, data dissemination packets may fall under this category. For these packets, we wish to transmit with the maximum power setting among all the node’s one hop neighbors. This policy ensures that all neighbors can receive the message, but that the node may still be able to transmit below maximum power if it has good-quality links to all of its neighbors.

True broadcast packets, on the other hand, should be sent to all one-hop neighbors *including* those that are not in the node’s neighbor list. Routing-layer beacon packets are a good example of this kind of traffic: lowering the power setting to cover the known neighborhood is counter-productive, since the routing layer intends to discover neighbors that it does not already know about. ART handles this traffic by broadcasting it at maximum power.

## 4.5 Overhead Analysis

It is worth noting that ART generally introduces no communication overhead aside from packet acknowledgements, since it operates solely on data packets being sent by the

upper layers. ART also has very low memory overhead: it needs a single bit to track the contention “gradient”; one byte to track the broadcast power; three bytes per link to track its state, transmission power, and last packet failure count;  $w$  bits per link to store its PRR window data; and  $2 \log w$  bits per link to store its window size and position. As demonstrated by our measurements in Section 6.1, this RAM requirement is well within the capabilities of existing sensor network hardware.

In networks with sporadic traffic patterns, there may not be enough data packets for ART to keep its sliding window up-to-date. There should be at least  $w$  packets sent within the time that it takes for links’ quality to fluctuate (which is dependent on network properties). ART could be augmented to deal with low-traffic workloads by injecting beacon packets into the network when the transmission rate over a link falls below this lower bound. We also note that, because ART operates below the routing and link layers, it can often leverage these layers’ control packets to update its sliding window even in the absence of application-layer transmissions.

## 4.6 Energy Efficiency

As a topology control algorithm, ART is designed to minimize the transmission power of individual links. In many sensor network applications, it is important to reduce the total *energy* consumption, since it has a direct impact on sensor lifetime.

Because ART is designed to minimize the transmission power of a link without violating the user-specified PRR bound, its energy efficiency depends the user selecting an appropriate PRR threshold for their application. For example, a threshold of 50% may not be appropriate for an application which requires 100% end-to-end reliable data transmission, because ART may cause nodes to spend more energy on re-transmissions than is saved by reducing the radio power. A threshold of 95% would be more appropriate for this environment, since a 5% retransmission overhead would likely be offset by similar or larger reductions in transmission power. (For example, on the CC2420 radio, even reducing the output power level from 0 dBm to -1 dBm will reduce the radio’s current consumption by over 5%.) As we will show in Section 6, ART is able to reduce transmission power with proportionally smaller drops in PRR. In addition, because of ART’s contention-handling optimization, it is sometimes able to *increase* the PRR while reducing the transmission power. These benchmarks demonstrate that ART is energy-efficient in practice.

## 5. IMPLEMENTATION

In this section, we present our implementation of ART for the TinyOS 2.0 operating system [20]. Our implementation of ART is built on top of the component-based MAC Layer Architecture (MLA) [21]. MLA augments TinyOS’s low-level radio drivers to provide the hardware-independent interfaces required by timing sensitive power management protocols. By leveraging these pre-existing interfaces, we were able to implement ART as platform-independent components within MLA. MLA also includes components that represent common MAC functionality and implements several optional power-saving MAC layers; we used a MAC layer which implements TinyOS 2.0’s default CSMA/CA logic.

In this section, we will discuss two major aspects of our implementation effort. First, we discuss the design of a new topology control layer on top of MLA. Second, we will describe our changes to TinyOS’s Collection Tree Protocol (CTP) [18] routing layer to allow it to modularly support a variety of underlying topology control schemes.

### 5.1 Interfacing with MLA

To implement ART, we used the existing MLA codebase and augmented it to add a new layer for topology control. The current public release of MLA is built on top of TinyOS 2.0.2 [22], the stable release of TinyOS as of this writing. TinyOS’s routing and link estimator components have had numerous bugfixes and enhancements since the release of TinyOS 2.0.2. In order to leverage these changes, we updated MLA to work with a CVS snapshot of TinyOS 2.1.

Once we applied these updates to MLA, we inserted a topology control layer into the radio driver architecture above the existing MAC layer, as shown in Figure 9. In keeping with the MLA design goals, we wished to design our topology control layer in a hardware-independent fashion, allowing it to be plugged into future MLA-supported radio stacks with little or no additional effort. We found that TinyOS and MLA already included platform-independent interfaces for the majority of the radio functionality needed by topology control schemes. However, there were two specific radio features for which we needed to create platform-independent hooks: adjusting the radio power and getting the signal strength of incoming packets.

To allow the topology control layer to adjust the radio power, we created the `PacketPower` interface:

```
interface PacketPower {
    async command uint8_t getPower(msg);
    async command void setPower(msg, power);

    async command uint8_t minimum();
    async command uint8_t maximum();
}
```

The `getPower()` and `setPower()` commands respectively get and set fields in the packet metadata corresponding to the power level at which the radio should transmit the packet. These commands are taken from TinyOS’s `CC2420Packet` interface, where the 8-bit `power` value is mapped directly onto the format of the CC2420’s 5-bit `PA_LEVEL` register. We therefore adjusted the semantics of the `PacketPower` interface to be more radio-independent. We added `minimum()` and `maximum()` commands to represent the range of the radio output power, and defined the behavior of the `getPower()` and `setPower()` commands so that all discrete values between `minimum()` and `maximum()` inclusive are mapped to radio-supported settings. As noted in Section 3, the CC2420 datasheet only defines the power output behavior for 8 of the possible 32 `PA_LEVEL` settings: 3, 7, 11, etc. We therefore modified the CC2420 stack to present its power range to the application layer as the contiguous range 0 ... 7, which it maps internally to supported `PA_LEVEL` settings.

The `PacketQuality` interface contains a single `getRssi` command, which returns the signal strength of an incoming packet<sup>3</sup>:

<sup>3</sup>We did not include a corresponding `getLqi()` command since LQI only applies to physical layers based on the 802.15.4 specification.

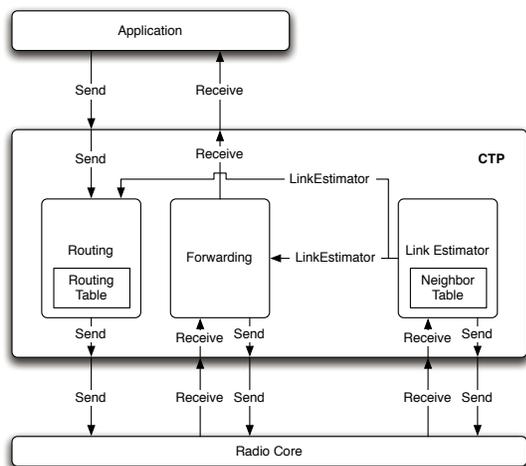


Figure 8: CTP’s original organization and architecture

```
interface PacketQuality {
    async command int8_t getRssi(msg);
}
```

Like the `getPower()` and `setPower()` commands, we extracted the existing `getRssi()` command from TinyOS’s `CC2420Packet` interface but modified its semantics to be radio-independent. The `CC2420Packet` interface directly returns the RSSI reading provided by the CC2420 radio chip, which is  $\sim 45$  dBm above the actual signal strength [2]. We redefined `getRssi()` to return the actual signal strength of the packet in dBm, and correspondingly modified the CC2420 stack to subtract 45 from RSSI readings provided to the application layer. While our ART algorithm does not use this functionality, other topology control algorithms (such as ATPC) require RSSI readings to select the appropriate output power level.

We also observed that the current CC2420 radio stack leaves the radio’s `PA_LEVEL` register set according to the most recent data packet transmitted. This behavior has a subtle implication for ACK packets: they will be transmitted at whatever power setting the last data packet was transmitted at, even if the ACK is being sent to a different neighbor. Because the CC2420 radio automatically generates ACK packets in hardware, we cannot instrument the CC2420 stack to set the optimal power setting of these ACK packets according to the topology layer’s decision. Instead, we reset the `PA_LEVEL` register to the maximum power after transmitting each data packet, so that all subsequent ACKs will be sent at max power.

## 5.2 Interfacing with CTP

The tree-based CTP routing protocol is the default routing protocol in TinyOS 2.x. CTP designates one or more nodes in the network as sink nodes. All other nodes in the network recursively form routing trees which are each rooted at one of these sink nodes. Nodes periodically broadcast beacon packets which serve two purposes. First, they contain a sequence number field which TinyOS’s link estimator component uses to compute the Estimated Transmission Count (ETX, roughly  $\frac{1}{PRR}$ ) to each node’s one-hop neighbors. Sec-

ond, nodes embed in these advertisements an estimate of the total cost (initially 0 for sink nodes and  $\infty$  for all other nodes) of routing a data packet to the sink through them. Non-sink nodes then select a parent on a routing tree by collecting their advertised routing costs, adding their one-hop ETX, and selecting the neighbor with the lowest total routing cost. Because CTP sends these beacons periodically, nodes can dynamically change their parents as link quality fluctuates.

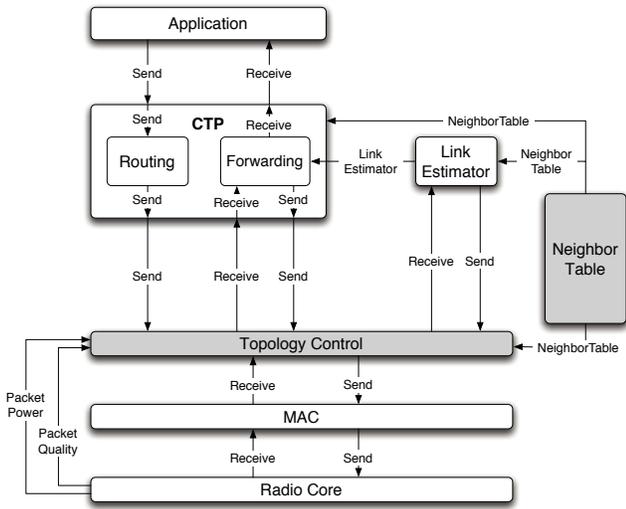
Our topology control layer is largely agnostic to the routing and application layers sitting on top of it: it only requires an external neighbor table for storing its own link quality data at runtime. We discovered that CTP’s default implementation is poorly-suited to allow other components to embed data in its neighbor table. As shown in Figure 8, TinyOS does not provide a single shared neighbor table component. Instead, TinyOS’s link estimator component (which computes the ETX across one-hop links to neighbors) and the CTP routing component (which computes the ETX across paths going through neighbors) maintain separate tables for their respective link quality data. This design choice increases the complexity of both components, since they must each include code to manage their own neighbor tables and to keep the two tables coherent. It also forces the `LinkEstimator` interface to include additional commands and events for the sole purpose of keeping the two tables coherent. As a result, although CTP and the beaconing link estimator are nominally independent components, as currently implemented they are tightly coupled.

We determined that extending this approach to include a third neighbor table (for the topology control layer) would be too clumsy. Instead, we extracted the neighbor table management code from CTP and the link estimator and used it to create a separate `NeighborTableC` component. We split each entry in the table into three “columns”: one each for the link estimator, routing engine, and topology control. To flexibly support different link estimator, routing, and topology control components, each component defines a nesC struct type representing its own data (`link_estimator_data_t`, etc.) which the neighbor table treats as a black box.

We extracted all of the neighbor table management functionality from the `LinkEstimator` interface and moved it into a new `NeighborTable` interface, which simplified wiring and provided a better separation-of-concerns. The resulting architecture is shown in Figure 9; the link estimator, routing components, and topology control layer are now decoupled. The only significant inter-component dependency is that CTP’s forwarding engine uses the simplified `LinkEstimator` interface to query the link estimator component.

As discussed in Section 4.4, we expect ART to send CTP’s broadcast beacons at maximum power. Because TinyOS does not differentiate between multicast and broadcast traffic at the radio layer, our ART implementation instead approximates our desired behavior by treating CTP control packets as a special case and transmitting them at maximum power. While this approximation is specific to CTP, we note that it does not introduce a compile-time dependency between CTP and ART: ART simply looks for a well-known constant in the TinyOS packet header which represents CTP control traffic.

Using this architecture, we implemented PCBL and ART as self-contained, platform-independent topology control layers. We also implemented a default topology control layer



**Figure 9: Our augmented architecture; major new components are shaded**

which simply passes through all packets untouched. Because these layers are self-contained, it is possible to interchange them at compile time using a compiler switch.

## 6. EXPERIMENTAL RESULTS

In this section, we present an empirical evaluation of ART on our testbed of TelosB motes. We first measure the ROM and RAM overhead of our implementation of ART within TinyOS. We then evaluate ART’s performance at the link level, and then compare ART’s performance against PCBL in a data collection scenario<sup>4</sup>. Finally, we evaluate the effectiveness of ART’s optimization for handling contention under heavy load.

Throughout this section, we deploy ART with a target PRR of 95% and a window size of 50 packets. Where not otherwise specified, our implementation of ART includes the contention-handling optimization described in Section 4.3. We use a neighbor table size of 32 entries in all experiments. We note that `NeighborTableC` includes code to evict old neighbor table entries, which was extracted from TinyOS’s link estimator. Therefore in practice, the CTP default of 10 neighbors should be sufficient for most applications; we increased the table size to 32 rows for the purposes of this benchmark in order to isolate the topology control layers’ behavior from that of the eviction routine.

### 6.1 Memory Footprint

A primary goal of ART is to provide a robust topology control algorithm which can realistically be deployed on hardware-constrained sensor hardware. It is therefore important that ART can be implemented with realistically-low overhead on RAM and ROM consumption.

Table 1 examines ART’s impact on application footprint. We compare the ROM and RAM usage statistics for the

<sup>4</sup>We did not include ATPC in this performance comparison, because the codebase used in [4] is not publicly available as of this writing, and ATPC’s relative complexity made it impractical to reimplement.

	ROM	RAM
Max Power	17794	4614
ART	19376	5006

**Table 1: The RAM and ROM overhead (bytes) of ART**

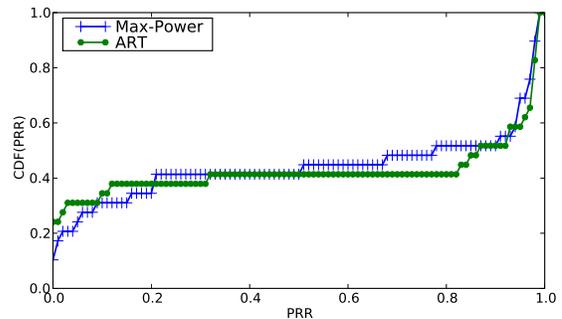
	PRR	Avg. Current
Max Power	56.7% ( $\sigma = 2.5\%$ )	17.4 mA ( $\sigma = 0$ )
ART	58.3% ( $\sigma = 2.1\%$ )	14.9 mA ( $\sigma = 0.32$ )

**Table 2: The link-level performance of max-power and ART**

benchmark application described in Section 6.2 when compiled for the TelosB motes, with and without ART; these statistics are generated by the TinyOS toolchain. There is a 392-byte difference in RAM consumption between ART and the default (max power) topology layer. 384 of these bytes can be attributed to the 12-byte topology control data column stored in the 32-row neighbor table. As noted above, most applications will not need a neighbor table of this size and will see a proportionally smaller memory overhead. The ROM overhead is larger at 1582 bytes, which is insignificant when compared to the ROM size of representative sensor hardware (e.g., 48 KB for TelosB).

### 6.2 Link-Level Performance

To examine the impact of ART on a per-link basis, we performed the following benchmark. We selected 29 links at random from the 524 links detected in our testbed during Section 3.5. We then deployed an application which cycled through these links round-robin, sending 100 packets over the one-hop link each time it was selected. Since our benchmark transmits only over a single link at a time, there is minimal contention. This cycle repeated for 150 rounds over the course of 24 hours. We performed this benchmark with no topology control (i.e., maximum power) and with our ART topology control layer; both benchmark runs used the same 29 links.

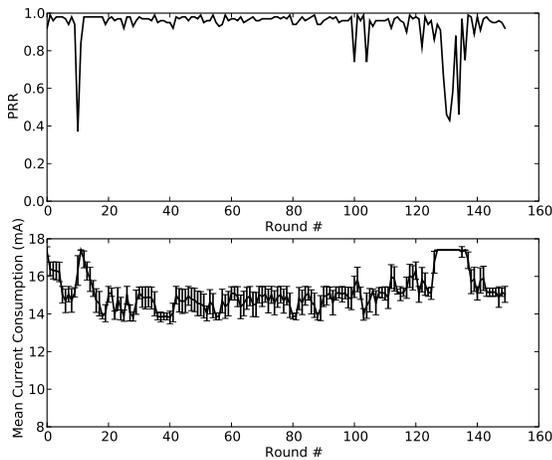


**Figure 10: The PRR distribution under max-power and ART**

The overall results are shown in Table 2. Max-power and ART have an insignificant difference in PRR results, demonstrating that ART indeed selects power levels equivalent in PRR to the maximum power setting. (Because there is minimal contention in this benchmark, ART cannot achieve a significant increase in PRR against max-power.) ART achieves

this with a 15% average reduction in current consumption over max-power.

ART’s overall PRR of 56.7% is significantly lower than this target PRR of 95%. This occurs because, even at maximum power, there is a bimodal distribution of link qualities as shown in Figure 10. For example, 15 of the 29 links in this experiment achieve a  $\text{PRR} \geq 90\%$ , while 9 of the 29 links achieve a  $\text{PRR} \leq 10\%$ . We note that ART and max-power have similar PRR distributions, again indicating that ART achieves similar PRR to max-power even on links where it is unable to meet its target.



**Figure 11:** The behavior of link 129 → 106 under ART

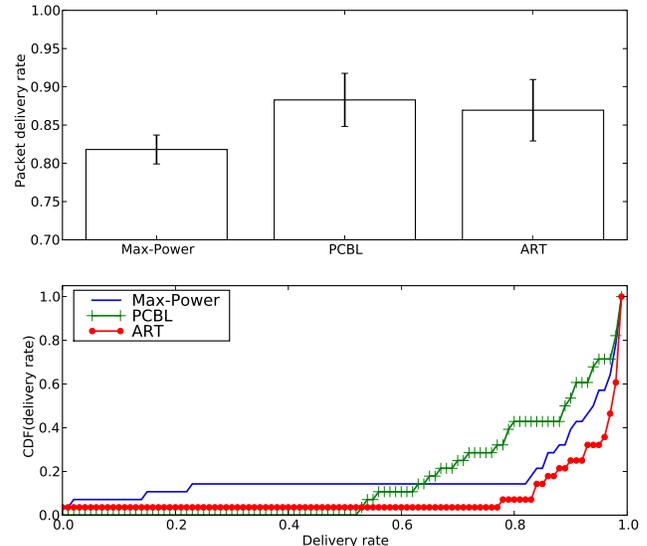
We now take a closer look at the ART’s behavior over one interesting link in the testbed, which is generally high-quality but still shows some link quality fluctuation. The PRR and average current consumption of this link are shown in Figure 11. We see that ART is able to lower the link’s current consumption by an average of 2.3 mA, responding to link quality fluctuations by tuning the power level accordingly. Of particular interest is ART’s behavior during round 10 and rounds 120–140, when link quality sharply drops and ART attempts to salvage the link by quickly going to maximum power. As a result, ART achieves an overall PRR of 93.7% across this link, close to the target PRR of 95%. ART performs slightly below the target overall because of these two temporary but sharp drops in link quality.

### 6.3 Data Collection

We evaluated the performance of ART against max-power PCBL on a multi-hop data collection application built on top of the CTP [18] routing library. To get a better understanding of the link-layer packet loss, we disabled CTP’s automatic packet retransmission routine. The application designated a particular node in the testbed as the tree’s root, and then waited 5 minutes for the routing layer to bootstrap. It then selected one node from the testbed and instructed it to send 200 data packets to the sink node, which recorded the sequence number and hop count of all packets it received. After the sender was finished with its 200 packets, another sender node was selected in a round robin fashion. We performed this experiment for 9 rounds over 4 hours at

max-power, and then repeated the experiment with PCBL and ART.

We reimplemented PCBL using the architecture described in Section 5. We configured PCBL to use the thresholds of 90% and 98% identified in [5]; we observed similar thresholds in our own testbed (see Section 3.4). To simplify PCBL’s implementation, we performed its bootstrapping procedure offline using 200 packets per node per power level.



**Figure 12:** The end-to-end delivery rate of max-power, PCBL, and ART under low contention

Figure 12 shows the end-to-end delivery rate under these three schemes. Both ART and PCBL achieve good PRR in this experiment, outperforming the max-power scheme by 6.4% and 5.1%, respectively. PCBL collects a large amount of link quality data up-front, allowing it to blacklist poor-quality links and prevent CTP from ever considering them. ART achieves comparable performance to by reducing transmission power, which reduces intra-path contention even when there is only one node sending at a time. We emphasize that ART achieves this PRR without the need for PCBL’s extensive bootstrapping phase. We also note that 75% of the sources achieve a delivery rate of 90% or higher under ART, compared to 61% under max-power and 46% under PCBL.

Looking closely at the distribution of PRRs among the senders, we observe that max-power has starved three of the senders with the highest average hop-counts (see Figure 13). This occurs because, although there is only one node producing data at a time, CTP will allow the application to produce a new packet as soon as the previous packet is one hop away from the sender. Therefore, a single sender may contend with its own packets which are still traversing a multi-hop path to the sink. This self-contention effect is the most pronounced when all packets are sent as maximum power, resulting in starved nodes. This finding underscores the importance of transmission power control, especially in multi-hop networks.

Figure 14 illustrates the total energy consumed by packet transmissions during each of these benchmark runs, nor-

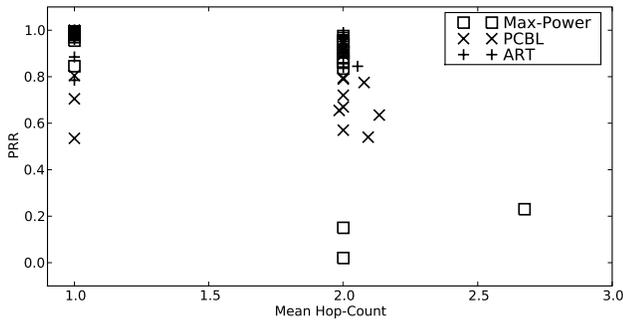


Figure 13: The relationship between PRR and hop count under max-power, PCBL, and ART

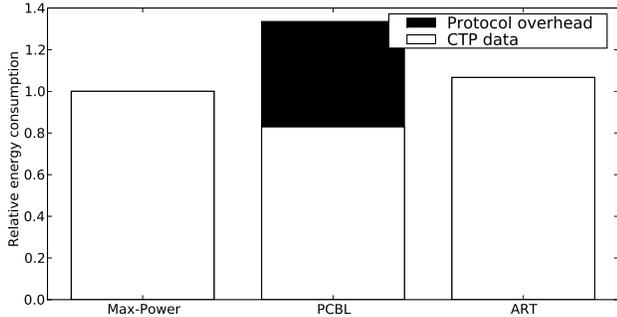


Figure 14: The energy consumption of max-power, PCBL, and ART under low contention

malized to the max-power energy consumption. ART has an energy consumption 6.6% higher than that of the max-power topology. This increase in power consumption occurs because for two reasons. First, ART has a 6.4% higher PRR than max-power and therefore transmits proportionally more packets through intermediate nodes. Second, as shown in Figure 13, max-power has starved the three nodes with the most expensive paths to the sink, which decreases total energy consumption at the expense of these senders.

Excluding its bootstrapping cost, PCBL achieves the lowest energy consumption, with a reduction of 17% compared to max-power. PCBL’s bootstrapping cost constitutes a 60% energy overhead in this benchmark; the relative overhead will decrease the longer the application remains active without rebooting PCBL. We project that PCBL would have achieved equal energy consumption to ART if the benchmark were extended to 8 hours and link conditions remained stable. We also note that rebooting PCBL can disrupt the network for extended periods of time: the bootstrapping phase took over 2 hours to complete in our testbed.

## 6.4 Handling High Contention

To explore the impact of ART’s contention handling optimization, we performed an experiment similar to that in Section 3.3. We selected ten links at random from the testbed and simultaneously sent data over all ten links in batches of 200 packets; we repeated this procedure for 30 minutes. (The same set of ten links was used throughout all benchmark runs; in the interest of fairness to PCBL, we

verified none of the ten links had been blacklisted.) We performed this experiment under the max-power, PCBL, and ART topology control schemes. In order to isolate the effect of ART’s contention-handling “gradient” optimization, we also repeated the benchmark with this optimization disabled.

Since we also wished to capture the effect of dynamic workload changes on PCBL’s behavior, we reused the PCBL bootstrapping data collected for the previous experiment. Accordingly, we do not include PCBL’s bootstrapping overhead when calculating energy efficiency.

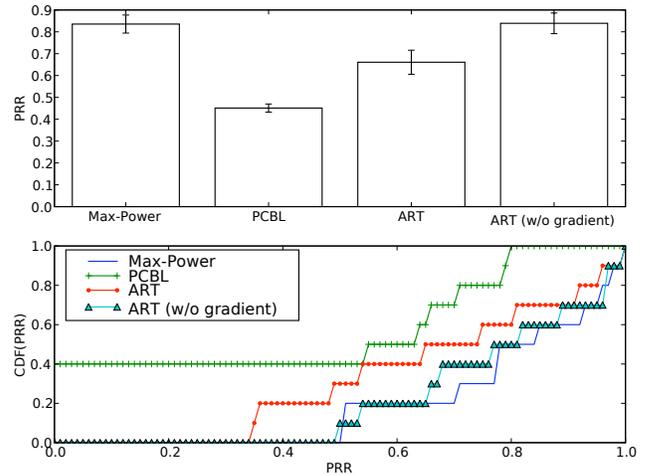


Figure 15: The PRR of max-power, PCBL, and ART under high contention

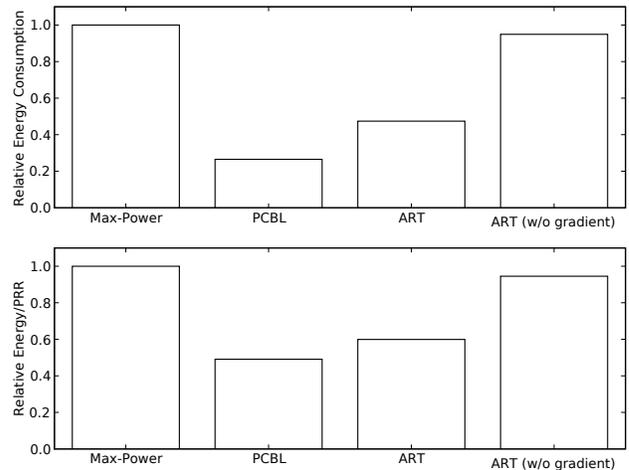


Figure 16: The energy consumption and efficiency of max-power, PCBL, and ART under high contention

Figure 15 shows the PRR of these benchmark runs. The difference in PRR between max-power (83.6%) and the unoptimized ART (83.9%) is insignificant. This occurs because the packet loss is too high for the unoptimized ART to ever leave the maximum power setting, and so its behavior is essentially identical to that of max-power. As shown in Figure 16, the unoptimized ART achieves only 5.1% energy savings over max-power for similar reasons.

The optimized ART achieves lower PRR (66.1%) than max-power, indicating that it cannot locate the optimal transmission power. This happens because there are many nodes which are rapidly sending packets and dynamically adjusting their transmission powers, both of which have a significant effect on the effective link quality. ART's sliding window mechanism cannot effectively track such rapid link quality fluctuations. We intend to explore this issue further in future work. Nevertheless, as shown in Figure 16, the optimized ART consumes only 47.4% that of max-power's energy. As a result, ART's energy efficiency (i.e., the average cost of successfully transmitting one packet) is 40.0% better than max-power.

PCBL achieves the lowest PRR (45.1%) and lowest energy consumption (26.5% that of max-power) among all four schemes. While this makes PCBL 50.9% more energy efficient than max-power and 18% more energy efficient than the optimized ART, it does so at the expense of starving four of the ten links. These four links had very good link quality during the bootstrapping phase, and so PCBL assigned them low transmission powers (two were assigned the lowest possible setting, while the other two were assigned the third-lowest setting). Under high-contention workloads, receiver nodes will overhear transmissions from nearby high-power transmitters and be unable to receive packets from these low-power transmitters.

## 7. CONCLUSION

Topology control can effectively save energy, maintain link quality, and alleviate contention. However, developing *robust* topology control protocols that perform well in indoor environments is challenging, due to complex and dynamic wireless characteristics. This paper first presents an empirical study that demonstrates the potential benefits and design guidelines for robust topology control in indoor environments. In particular, we found that RSSI and LQI are *not always* robust indicators of link quality in indoor environments and that profiling links even for several hours is not sufficient for identifying links whose PRR is consistently high. These negative results are important because they were the underlying assumptions of state-of-the-art topology control algorithms such as PCBL and ATPC. We then present ART, a *robust* topology control algorithm for dynamic indoor environments, which does not rely on RSSI/LQI as indirect measurements of link quality or on a prolonged bootstrapping phase. ART adapts the transmission power of a link in response to environmental changes as well as varying degrees of contention. ART also features a simple, yet effective "gradient"-based approach for handling network contention. Furthermore, ART is an *efficient* algorithm suitable for resource-limited sensor network platforms. It introduces minimal processing overhead; adds only 1582 bytes of ROM and 392 bytes of RAM; and introduces no communication overhead for applications using packet acknowledgements. Experiments run on our testbed show that ART reduces power consumption compared to maximum power without degrading link quality. In addition, macro-benchmarks which emulate a realistic data collection application indicate that ART outperforms the maximum power in terms of PRR, and achieves comparable performance to PCBL without relying on costly bootstrapping. ART also effectively handles heavy contention by improving energy efficiency while avoiding link starvation.

## Acknowledgement

This work is supported by NSF NeTS-NOSS Grant CNS-0627126 and CRI Grant CNS-0708460. We would also like to thank Paolo Santi and the reviewers for their valuable feedback.

## 8. REFERENCES

- [1] *CC1000 Single Chip Very Low Power RF Transceiver*, Texas Instruments.
- [2] *2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*, Texas Instruments.
- [3] K. Srinivasan and P. Levis, "RSSI is under appreciated," in *EmNets*, 2006.
- [4] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, "ATPC: adaptive transmission power control for wireless sensor networks," in *SenSys*, 2006.
- [5] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks," in *IEEE SECON*, 2004.
- [6] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *SenSys*, 2003.
- [7] N. Reijers, G. Halkes, and K. Langendoen, "Link layer measurements in sensor networks," in *IEEE MASS*, 2004.
- [8] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *MobiSys*, 2004.
- [9] D. Lal, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, and A. Keshavarzian, "Measurement and characterization of link quality metrics in energy constrained wireless sensor networks," in *IEEE GLOBECOM*, 2003.
- [10] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "Complex behavior at scale: An experimental study of low-power wireless sensor networks," UCLA Computer Science Department, Tech. Rep., 2002.
- [11] J. Zhao and R. Govindan, "Understanding packet delivery performance in dense wireless sensor networks," in *SenSys*, 2003.
- [12] L. Li, J. Y. Halpern, P. Bahl, Y.-M. Wang, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks," in *ACM PODC*, 2001.
- [13] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1195–1206, 2005.
- [14] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, 2006.
- [15] M. Burkhart, P. von Rickenbach, R. Wattenhofer, and A. Zollinger, "Does topology control reduce interference?" in *MobiHoc*, 2004.
- [16] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "Topology control with better radio models: Implications for energy and multi-hop interference," *Perform. Eval.*, vol. 64, no. 5, 2007.
- [17] Y. Gao, J. C. Hou, and H. Nguyen, "Topology control for maintaining network connectivity and maximizing network capacity under the physical model," in *INFOCOM*, 2008.
- [18] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, and A. Woo. The collection tree protocol (CTP). [Online]. Available: <http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html>
- [19] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, "Temporal properties of low power wireless links: modeling and implications on multi-hop routing," in *MobiHoc*, 2005.
- [20] [Online]. Available: <http://www.tinyos.net/>
- [21] K. Klues, G. Hackmann, O. Chipara, and C. Lu, "A component based architecture for power-efficient media access control in wireless sensor networks," in *SenSys*, 2007.
- [22] [Online]. Available: <http://www.cse.wustl.edu/wsn/index.php?title=MLA>