Collaborative Scheduling in Dynamic Environments Using Error Inference

Qingquan Zhang, *Senior Member, IEEE*, Lingkun Fu, *Student Member, IEEE*, Yu (Jason) Gu, *Member, IEEE*, Lin Gu, Qing Cao, *Member, IEEE*, Jiming Chen, *Senior Member, IEEE*, and Tian He, *Senior Member, IEEE*

Abstract—Due to the limited power constraint in sensors, dynamic scheduling with data quality management is strongly preferred in the practical deployment of long-term wireless sensor network applications. We could reduce energy consumption by turning off (i.e., duty cycling) sensor, however, at the cost of low-sensing fidelity due to sensing gaps introduced. Typical techniques treat data quality management as an isolated process for individual nodes. And existing techniques have investigated how to collaboratively reduce the sensing gap in space and time domain; however, none of them provides a rigorous approach to confine sensing error is within desirable bound when seeking to optimize the tradeoff between energy consumption and accuracy of predictions. In this paper, we propose and evaluate a scheduling algorithm based on error inference between collaborative sensor pairs, called *CIES*. Within a node, we use a sensing probability bound to control tolerable sensing error. Within a neighborhood, nodes can trigger additional sensing activities of other nodes when *inferred* sensing error has aggregately exceeded the tolerance. The main objective of this work is to develop a generic scheduling mechanism for collaborative sensors to achieve the error-bounded scheduling control in monitoring applications. We conducted simulations to investigate system performance using historical soil temperature data in Wisconsin-Minnesota area. The simulation results demonstrate that the system error is confined within the specified error tolerance bounds and that a maximum of 60 percent of the energy savings can be achieved, when the *CIES* is compared to several fixed probability sensing schemes such as eSense. And further simulation results show the *CIES* scheme can achieve an improved performance when comparing the metric of a prediction error with baseline schemes. We further validated the simulation and algorithms by constructing a lab test bench to emulate actual environment monitoring applications. The results show that our approach is effective and efficient in tracking the dramatic temperature shift in dynamic environments.

Index Terms—Collaborative scheduling, duty cycle control, energy efficient, neighbor error control

1 Introduction

Wireless Sensor Networks (WSNs) have found applications in a wide range of problems, from military surveillance to environmental monitoring, disaster relief, and home automation. In spite of their broad utility, energy efficiency remains a critical challenge because many WSNs nodes are normally equipped with limited power sources. As is well known, one of the major sources of power consumption is the energy cost of sensing activities. If working continuously, a sensor node can typically sustain only a few days. On the other hand, long-term applications [2] are normally required to last for weeks or even months. Therefore, unnecessary sensing activities should be avoided to extend the lifetime of sensor networks. The discrepancy between limited resources and stringent requirements makes it necessary to develop scheduling protocols to turn on and off (i.e., duty cycle) sensors to conserve energy. Dynamic sensing scheduling is an effective method to reduce sensing activities and thereby improve network energy efficiency. Especially a collaborative way between nodes is relevant when the node sensors are needed to monitor a dynamic environment.

There is a great amount of research work on collaborative sensing [3], [4], [5], [6], [7]. Most of these projects focus on how to efficiently select or deploy a minimum set of sensor nodes to provide a full/partial spatiotemporal coverage. We note this work determines sensing activities of nodes based on coverage requirements in *space* and/or *time*. None of them focuses on how to schedule sensing activities based on *sensing error*, and hence fails to provide a rigorous approach to confine data accuracy within desirable bounds when seeking to optimize the tradeoff between energy consumption and accuracy of predictions.

Our work incorporates the advantage of both dynamic scheduling and sensors collaboration, seeking to optimize the tradeoff between energy consumption and accuracy of predictions. By taking a completely different approach, we schedule sensing activities based on two types of information: 1) local estimated error and 2) inferred error from neighbor nodes. A node turns on its sensors when either

- *Q. Zhang is with Lemko Corporation, 721 S Oak St, Palatine, IL, 60067. E-mail: zhan0511@umn.edu.*
- *L. Fu and J. Chen are with the Room 517, Zhejiang University, Zheda Road 38#, Hangzhou, Zhejiang 310027, P.R. China. E-mail: {fzjuben, jmchen.zjug}@gmail.com.*
- *Y.J. Gu is with the Pillar of Information System Technology and Design, Singapore University of Technology and Design, 20 Dover Drive, Singapore 138682. E-mail: jasongu@sutd.edu.sg.*
- *L. Gu is with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Office: 3562 (Lift 25/26), Clear Water Bay, Kowloon, Hong Kong. E-mail: lingu@cse.ust.hk.*
- *Q. Cao is with the Department of Electrical Engineering and Computer Science, University of Tennessee, Min Kao 430, Knoxville, TN 37996. E-mail: cao@utk.edu.*
- *T. He is with the Department of Computer Science and Engineering, University of Minnesota Twin Cities, 4-205 EE/CSci Building, 200 Union Street SE, Minneapolis, MN 55455. E-mail: tianhe@cs.umn.edu.*

Manuscript received 1 Apr. 2012; revised 9 Sept. 2012; accepted 21 Jan. 2013; published online 14 Feb. 2013.
Recommended for acceptance by M. Thai.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2012-04-0341.
Digital Object Identifier no. 10.1109/TPDS.2013.28.

TABLE 1
A Breakdown of Energy Cost for Different Components on Sensor Node

Components	Energy Consumption Rate
ATmega128L	6 mA
Transceiver	10 mA
Magnetometers	90 mA
Camera sensor	107 mA

error type exceeds a user specified error tolerance. The major challenge in our design is how to determine neighbor nodes' potential risk of violating the data accuracy requirement while minimizing the energy consumption. Our design has several major advantages over existing single-node scheduling methods [8]: 1) nodes can share sensing error information and process it with limited resources, 2) nodes can collectively control sensing errors through neighborhood coordination, and 3) a network can respond to dramatic environmental changes more quickly, which property is desirable in environment monitoring applications. Meanwhile, as our approach targets the long-time duration monitoring application, the general sensing coverage service [7], which requires minimum coverage for an interested sensing region, can be satisfied through neighbors' error inference mechanism that will trigger neighbors to be activated when necessary.

We investigate a sensing-scheduling algorithm, called *collaborative inferred error sensing (CIES)*, focusing on monitoring applications such as habitat monitoring in which high sensitivity to rapid environment change is desired. It can provide high data accuracy while minimizing the amount of energy used. Specifically, our design exploits neighbor node coordination to reduce possible violations against sensing fidelity requirements. The driving idea of our work is *error inference*, where the term *error* is defined as the difference, in percentage, between the ground truth environmental data and the corresponding value generated by the predictor of sensor nodes, which is a direct performance indicator of the sensor system. The error information is not only used by the local sensing scheduler, but is also shared among neighbors. Nodes can trigger additional sensing activities of other neighbor nodes when the inferred error has aggregately exceeded the error tolerance. We refer to our proposed approach as *CIES*.

The main contributions of this work are:

- The design of a local error control algorithm to guarantee a specified error bound.
- The introduction of a distributed inference model of prediction error for neighbor nodes.
- The integration of both local and neighbor error control into a unified architecture to adjust duty cycles of sensor nodes.
- The simulated study and test-bed implementation of the proposed design that conserves as much as 60 percent of energy compared to other solutions, while confining sensing error within specified error tolerance.

The remainder of this paper is organized as follows: We present the overview for our design in Section 2. Sections 3 and 4 describe the details of the error control mechanisms.

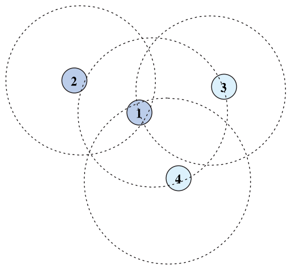


Fig. 1. Initial operation graph.

The performance evaluation is presented in Section 6. Section 7 surveys the related work and Section 8 concludes the paper.

2 Overview and Objectives

This section presents an overview of our *CIES*. We first present the network model and assumptions of the work, then describe the overall system design.

2.1 Network Model

Assume a wireless sensor network is composed of N sensor nodes. Each sensor node has two states: an active state and a dormant state. An active node performs all functionalities, such as sensing events, transmitting packets, and receiving packets. A dormant node turns off most functional modules except the radio for listening to incoming traffic. All nodes have their own schedules that are controlled by the duty cycle controller on the nodes. A dormant node wakes up when 1) it is scheduled to switch the an active state, or 2) it receives triggering packets from neighbors and decides to change into the active state.

2.2 Assumptions

We assume that we use off-the-shelf sensor node products [9]. Without loss of generality, in our design and implementation, sensor nodes are homogeneous and can be distributed as a random process. We also assume that in our target sensing platforms, sensing is much more expensive than communication, so that it is necessary to coordinate sensing activities among neighbor nodes for better energy efficiency. Certainly, this assumption does not hold for all platforms, but it does apply to a few existing ones. For example, the magnetometers used in the MICA sensor boards and XSM nodes draw about 90 mA of current, as compared to 6 mA for the ATmega128L microcontroller and 12 mA for the transceiver [10]. For clearer description, we summarize a breakdown of energy cost for different components on sensor node in Table 1. This assumption also holds well in platforms where expensive sensors (e.g., camera [11] and micro-power-impulse radar (MIR) [12]) and low-power-listening techniques [13] are used.

2.3 Motivating Example

A simple example is used to illustrate the basic idea of *CIES*. Fig. 1 shows an example with a node set $G = \{1, 2, 3, 4\}$.

1. After deployment, nodes 1, 2, 3, and 4 initialize their local error control operation processes and neighbor

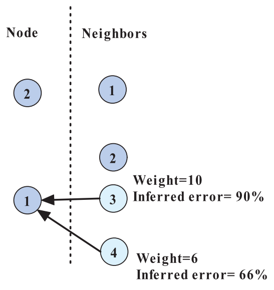


Fig. 2. The computation of weighted inferred error.

error-control procedures independently. Node 1 has neighbor nodes 2, 3, and 4 as shown in Fig. 1.

2. In one sampling time duration, nodes 3 and 4 are initially in full operation (in light color as shown in Fig. 1). Nodes 1 and 2 are scheduled to be in a sleep mode, within which mode their radio is switching on and off periodically to monitor traffic.
3. When there is a dramatic change in the environment, nodes 3 and 4 calculate their inferred errors on node 1 to be 90 and 66 percent, respectively, as shown in Fig. 2. These error estimations from node 3 and 4 are sent to node 1. Because both nodes 1 and 2 are in sleeping mode, there is no message exchange between them.
4. After receiving the inferred error messages from nodes 3 and 4, node 1 evaluate the weighted average inferred error to be 80 percent. Then, it determines whether error tolerance is violated. If the error tolerance in this simple example is 30 percent, node 1 is awakened immediately after the sampling time duration because the calculated weighted inferred error is larger than the threshold.
5. The operational modes of these nodes are indicated in Fig. 3.

From this simple example, we have the following observations. First, *CIES* exhibits great potential as for reducing error rates due to rapid environmental change. Compared to sole prediction as in eSense [8], *CIES* improves the capability for sensor collaboration. Second, the prediction model and data observation correlation among nodes are critical, and distributed algorithms are required.

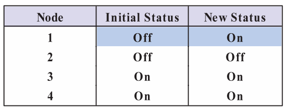
2.4 A Walk-Through of the Basic Operating Procedures

In this section, we overview the collaborative scheme of our design using a walk-through example, and the details are given in Section 1 of the supplementary file, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.28.

2.5 Objective and Challenge

As illustrated by above walk-through, *CIES* exploits neighbor collaboration in the event detection, thus showing a unique potential for reducing errors in environment monitoring applications. Compared to other local error control schemes [8], *CIES* extends the error control from local node level to the network level, setting the foundation for more complex applications.

The main objective of this work is to develop a generic scheduling mechanism for collaborative sensors to achieve error-bounded scheduling control in monitoring applications. The fundamental challenge in our design is how to reconcile the conflict between energy consumption and error tolerance. To achieve high accuracy of measurements for highly dynamic environments among collaborating sensor nodes, optimized approaches to accurately determine the inferred errors between neighbor nodes are investigated. At the same time, to minimize energy consumption, the minimum duty cycle needs to be determined.



Node	Initial Status	New Status
1	Off	On
2	Off	Off
3	On	On
4	On	On

Fig. 3. The initial and final operational modes.

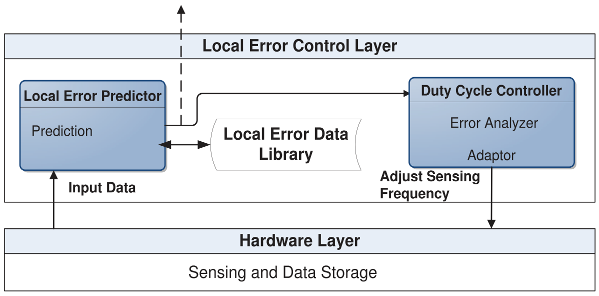


Fig. 4. The local error control layer illustration.

3 Local Error Control

The design of the local error control is motivated by the observation that a sensor node should be able to achieve the desired sensing fidelity independently even in isolation. Therefore, the data detected and stored locally should also be fault tolerant, a goal that is achieved by the local error control.

For convenience, we refer to the local error control as *Noncollaborative IES*, which consists of the duty cycle control and local error predictor as shown in Fig. 4. To save energy, a sensor node uses its local error predictor to predict the environment status without performing actual sensing operation. When data are obtained through actual sensing, a node compares predicted sensing values with the actual sensing values, and then stores the prediction errors into the local error data library. Based on the accuracy of the local error predictors, the duty cycle controller adjusts the sensing frequency through *error bound control*, which serves to confine the system prediction error within a user specified bound.

3.1 Local Error Predictor

To conserve their limited power supply, sensors do not continuously sense data. Instead, they operate at some selective cycles as long as the data quality is acceptable. The data in the remaining cycles are reconstructed through appropriate prediction models. If the environment exhibits cyclic patterns, an empirical model is used to establish strong correlations in the data and to organize them in a certain way so that future data can be extracted from the empirical or historical ones.

Fig. 5. The preliminary temperature measurement experiments.

Depending on system lifetime and the data fidelity requirement of an application, the empirical model can be constructed in different ways. Similar to the work of [14], we developed a cycle-based empirical model [15], which has been proven to be efficient for environment monitoring applications. The error predictor is mainly responsible for generating prediction errors, defined as e_i, for each node i. Our preliminary experiments of temperature measurements, as shown in Figs. 5 and 6, demonstrate that the error predictor can adapt to the environmental changes sufficiently well.

Since the energy resource on individual sensor nodes is limited, empirical model in this application domain can simplify data processing, and thus extend the lifetime of sensor nodes. However, the model selection can be flexible, and the *duty cycle controller* can adapt the system to the relative error induced by different prediction models. Moreover, the local error predictor provides a reliable reference for duty cycle controller to perform further analysis.

3.2 Duty Cycle Controller

The duty cycle controller receives and analyzes the prediction errors from the local error predictor. The first step in designing the duty cycle controller is modeling sensing behavior of the system mathematically to derive the relationships among the local prediction error, the current duty cycle, and system requirements. In this design, we separate the controller into the error analyzer and duty cycle adaptor, two processes that can run collaboratively.

3.2.1 Error Analyzer

We determine the error analyzer theoretically as follows: We assume that the sensing baseline consists of N data cycles, in which k warm-up cycles are used for building controller models. In each cycle, the probability that a sensor node performs actual sensing operation is defined as sensing probability p_i. To simplify the description, without loss of generality, instead of considering energy cost spent on different components, (e.g., sensing and processing), we use average energy consumption to represent the total energy cost of a node to sense, process and communicate in each sensing period. Let the average energy consumption for sensing be E_a. When a sensor is inactive, it does not sample the environment; instead, it uses the local predictor to estimate sensing readings, which introduces prediction errors. Let the potential prediction error at each cycle be e_i. And let the maximum prediction error tolerance specified by a user be e_t. Therefore, the goal

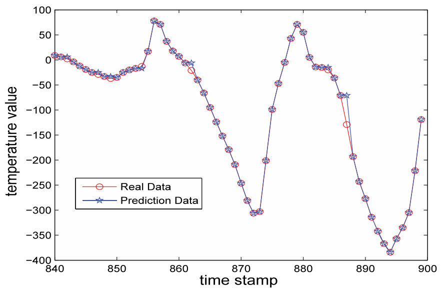


Fig. 6. An example of output from predictor (error tolerance 10 percent, x-axis is the time-stamp).

of our design is to minimize the energy consumption during each baseline period:

$$E = \sum_{i=1}^{k} E_a \cdot t_i + \sum_{i=1}^{N-k} p_i \cdot E_a \cdot t_i, \tag{1}$$

under the constraint that

$$\frac{\sum_{i=1}^{N}(1-p_i)\cdot e_i}{N} = \frac{\sum_{i=k}^{N}(1-p_i)\cdot e_i}{N} \leq e_t, \tag{2}$$

where t_i is the unit cycle length, k is the length of cycles used to stabilize the scheduling system and N represents the total length of operational cycles . The constraint enforces that the potential statistical error caused by the prediction is smaller than the error tolerance. The range of possible values of p_i is bounded to satisfy the constraint equation.

The minimization of energy consumption deals with several key issues, for example, the length of the training cycle and the prediction model used. Now the problem is to determine the appropriate p_i for a given error range e_i obtained from past data values. To solve for sensing probability p_i at a specific e_i requires a joint distribution of a process for e_i at specific time instance or period. This requires a heavy computation and storage overhead on the limited resource of the sensor node. Obtaining a solution for sensing probability p_i is extremely difficult to calculate during transitions. Instead, we introduce a lightweight method for computation which allows the sensor to choose a value within a range. We first determine the bound for sensing probability p_i, and the algorithm chooses one value within that bound.

It should be clear that with a higher value of sensing probability p_i, larger energy consumption is needed. With a lower value of sensing probability p_i, a higher probability is for the error because the prediction result is greater than the tolerance. Therefore, we need to analyze the bound of sensing probability p_i to optimize this tradeoff.

3.2.2 Determining the Sensing Probability Bound

We use a bottom-up approach to set a bound for the sensing probability. That is, if we do not violate the error constraint in every cycle instance, it is certain that the inequality (2) holds. As noted, this sets a stricter requirement than the constraint equation over all sampling instances. By doing so, our probability constraint problem can be simplified into choosing the p_i at each scheduling cycle to satisfy the constraint on $(1-p_i)\cdot e_i$, which can be solved as

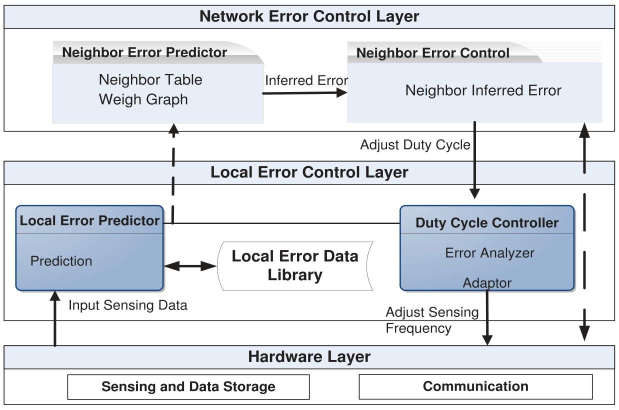


Fig. 7. The network error control layer illustration.

$$p_i^{lb} = \begin{cases} 0 & 0 \le e_i \le e_t \\ 1 - \dfrac{e_t}{e_i} & e_t \le e_i \le 1 \end{cases} \tag{3}$$

The p_i^{lb} is the lower bound of p_i that guarantees data quality requirement at each sensing cycle instance. Only values higher than this assure that the constraint requirement would not be violated under any circumstances. We should also be careful in the selection of p_i, as a higher p_i implies more energy consumption by the sensor node.

We also note in (3) that the lower bound p_i^{lb} is affected by the prediction error e_i. A large prediction error e_i imposes a higher bound, leading to high energy consumption. The critical issue is to reduce this prediction error e_i. Clearly it can be achieved with a better prediction model; however, individual nodes are limited in their ability of sensory measurements and collaborative schemes usually prove better [16], [17]. Therefore, we need online method to improve error control, which is achieved through the *network error control* described in next section.

4 Network Error Control

In this section, we present the design of network error control as shown in Fig. 7. Recall in our analysis, in Section 3.2.2, it is essential to predict the neighbors' model prediction error accurately and to share such information among them effectively. To ensure the accuracy of such prediction and information sharing, we assign tasks to two processes: 1) neighbor error predictor, 2) neighbor error controller. The process running on this network control layer is named *collaborative IES*, which aims to maximize the energy saving and minimize the prediction error of the sensor system.

Before further discussion, we define several terms used to describe the processes.

Definition 1 (Inferred Error e_{ij}). *Given a node i and its neighbor node j, the node-pair inferred error e_{ij} is defined as the inference error at neighbor j from the point view of node i.*

Definition 2 (Node-Pair Weight w_{ij}). *The weight is defined as the extent of a node-pair's data correlation and indicates how similar the sensing observation is between two neighbor nodes i and j.*

Definition 3 (Error Probability Density Function $\rho(e)$). *The error PDF is a collection of distributions of detection errors in which the past detection errors for sensors are stored and processed so that the detection errors can be directly linked to corresponding occurrence probability. The neighbor nodes exchange the error PDF locally. We can easily derive its statistical accrual error probability mass function(PMF) once the PDF is given.*

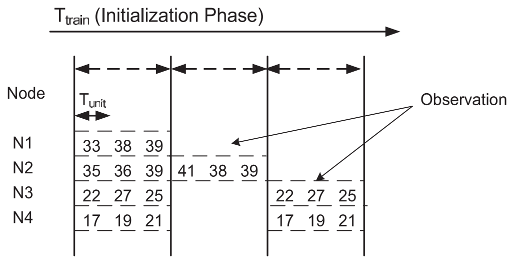


Fig. 8. The process of correlation calculation.

4.1 Design of Neighbor Error Predictor

Since the neighbors can change dynamically, we need to iteratively estimate the neighbors' prediction error, given a certain relationship among neighbors.

Step 1: Neighbor Recognition. The control process starts with neighborhood discovery. During this phase, the sensors acquire the knowledge that which close sensors around them can build up a "trust" relationship, which can be characterized as node-pair weights. The formation of neighborhood may be based on different requirements such as vicinity, link quality or the displacement along the routing path of the sensing data. In this stage, each node recognizes its neighbor nodes and assigns a table for each neighbor to build the weight graph. Note that the neighborhood formation is a dynamic process which is refreshed after a defined period. By the end of this process, sensors recognize their neighbors and data storage structures created for neighbors are also initialized.

Step 2: Weight Graph Construction. As pointed in our earlier assumptions, nodes are synchronized with each other [18], [19], [20], and T_{train}, the time for initialization, is divided into equal time durations T_{build}, as shown in Fig. 8. Each time duration includes m equal duration intervals, where an interval is a unit sample time period set by the user.

For each round, each node N_i stores its observation vector $\{o_1^i, o_2^i, \ldots, o_m^i\}$ obtained through discrete sampling at $T_i = \{t_1^i, t_2^i, \ldots, t_m^i\}$. At the end of each round, each node exchanges the observation vector, which is used to calculate the correlation between nodes. This process is repeated until the end of T_{train}, so that the average sensing correlation between nodes can be estimated.

Specifically, we use an approach to calculate data correlation between two observation vectors $C(i,j)$ by node N_i and node N_j, and also the weight value $w(i,j)$ among nodes, as reported in Section 2.1 of the online supplemental material.

Step 3: Achieving the inferred error e_{ij} for neighbors. The control of sensing errors in the network is further guaranteed by the collaboration of neighbor nodes. The observations that sensor nodes demonstrate spatial correlations found in [21], [22], [23] are also supported by our preliminary experiments described in Section 3. The measurement distributions of collaborative nodes have the same confidence level. Motivated by such observations, we can predict error of neighbor nodes using local prediction error. That is, an active sensing node, by comparing its real-time sensing values with corresponding predicted values, can infer the prediction errors of correlated neighbor nodes. In this way, our neighbor-error inference scheme ensures real-time tracking and quick response to the error status change within a sensing group. The process can be summarized as follows:

- At a sampling cycle m, assuming sensor i is active in sensing and computation, we can easily calculate the observation error e_i^m at source node i based on the difference between actual sensing data and prediction values that are generated by our prediction model.
- From its error *probability density function* $\rho(x)$, node i evaluates the cumulative distribution function $PMF_i(e_i^m)$ as in (4). From this result, we can infer the statistical confidence level of the worst error occurrence at node i as

$$PMF_i\left(e_i^m\right) = \int_{-e_i^m}^{e_i^m} \rho(x)dx. \tag{4}$$

- Upon obtaining the PMF_i, the active sensor node i calculates the inferred error of a neighbor sensor j, based on *PDF* information of sensor node j. Given the neighbors error models, an iterative step is performed:

$$e_{ij} = PMF_j^{-1}(PMF_i(t[k])), \tag{5}$$

and the variable $t[k]$ is expressed as

$$t[k] = \begin{cases} 2 \cdot t[k-1] & PMF_j(t[k-1]) < PMF_i\left(e_i^m\right) \\ \frac{t[k-1]+t[k-2]}{2} & PMF_j(t[k-1]) > PMF_i\left(e_i^m\right), \end{cases} \tag{6}$$

in which $PMF^{-1}()$ is the inverse function of PMF and $t[0] = 0, t[1] = e_i^m$. The iterative process does not stop until $PMF_j\ (t[k-1]) = PMF_i(e_i^m)$.

Two examples of the inferred error computation process are given in Section 2.2 of the online supplemental material.

4.2 Neighbor Error Control

After estimating the error value for neighbor sensor j, sensor i sends the inferred error e_{ij} to neighbor error control process where the error information is sent and received. The neighbor error control process monitors the channel through which its neighbors send error information. To minimize the false positive risk, a weighted average approach is adopted in this design.

Definition 4 (Weighted Average Inferred Error). *Given a neighborhood $G(V, E)$, a sensor j's weighted average inferred error e_j is the weighted average of all node-pair inferred errors, i.e, e_{dj}, where d and j are a neighborhood pair.*

The weighted average error e_j is obtained according to the following rationale. Node-pairs provide different inferred errors due to correlation relationships or other influences. Pair inferred errors have a specific weight value based on their degree of similarity. A higher weight value means a higher probability for data similarity. Therefore, the sensor platform must take this into account when determining whether the sensor needs to adjust its current operating status. Based on our observation, the inferred error can be expressed as

$$e_j = \frac{\sum_k e_{kj} \cdot w_{kj}}{\sum w_{kj}, k \in N(j)}, \tag{7}$$

where k is size of the neighborhood of the node j, w is the node-pair weight, and $N(j)$ is the neighborhood list of node j. As shown in Fig. 4d of the online supplemental material, sensor node j receives several isolated error estimations from neighbor sensors i, l, and q. The e_j should be viewed as a total effective contribution from all the neighboring inferred errors on a weighted basis. Apparently if one sensor detects that the estimation error currently violates the error threshold, its neighbor nodes having a high weight value are expected to experience a high risk of violating the data accuracy as well.

The process that should be executed during all phases of operation is described in Algorithm 1 as given in Section 3 of the online supplemental material.

5 Insight of *IES* and *CIES* System

We investigate some insights of *IES* and *CIES* system, such as sensor lifetime estimation based on *IES* and the analysis of node density to the performance of *CIES*, which details are given in Section 4 of the online supplemental material.

6 Evaluation

To evaluate, we develop a simulation program that uses historical soil temperature data. The temperature data were collected from the Wisconsin-Minnesota Cooperative Extension Agricultural Weather Page [24] where the soil temperature is monitored continuously, sampled twice per hour, 24 hours per day, for over 10 years. This data set is large enough to reduce sampling randomness, allowing us to verify our algorithm and investigate the impact of different configurations on the performance of energy conservation and error control. In our experiments, we use randomly deployed sensors in a square area of $200\ \mathrm{m} \times 200\ \mathrm{m}$. We define a pair of nodes as neighbors when their distance is less than 20 m. Moreover, we use a diffusive model to fill up the simulation environment, in which we consider the environment as a homogeneous semi-infinite medium. Various benchmark approaches such as the fixed-probability sensing scheme are simulated to generate the metric data.

6.1 Metrics and Baseline

To evaluate the scheduling quality of a sensor network, we define metrics as follows:

- *Error Rate*: This metric is defined as the error rate that the prediction system produces during the same observation window.
- *Miss Ratio*: This metric is defined as the ratio that the sensor system fails to respond to an event, for example, a rapid change in environment.

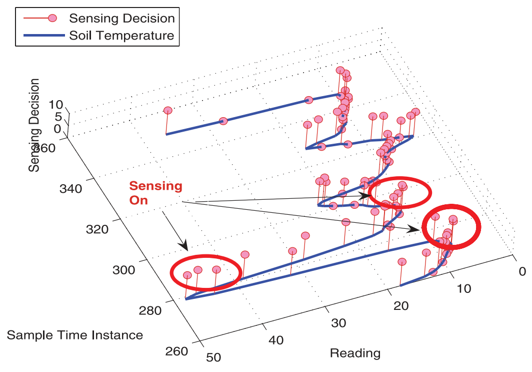


Fig. 9. The sample of sensor activities.

- *Energy Consumption*: This metric is defined as the total energy consumed by the network during the operation period.

The sensing schemes proposed are assessed using the above metrics with respect to different system parameters, for example, the error tolerance. Through these examples, comparison between different benchmarks and our proposed *CIES* is used to demonstrate the performance of our design.

6.2 The Mechanism of Our Error Bounded Approach

One of the benefits of this adaptive and collaborative approach is that it tries to reduce the average error for the entire operation period. The stricter guarantee is that it limits the error at each sampling instance, which enhances the system performance. The difficulty in limiting the errors is to determine when to switch on the sensors whenever there is a dramatic change in the environment. Our approach achieves this by relying upon both local and network error control mechanisms. Local error control can guarantee the error bound when model-based prediction works well. However, when environments experience dramatic changes, model-based prediction no longer works. In such scenario, the network error control mechanism relies on active nodes to trigger inactive nodes, when the inferred error of the inactive nodes exceeds the bounds.

As shown in Fig. 9, when environment temperature experiences corner-like change, sensing activities becomes more intensive than other sampling periods. This is because the network error control triggers more nodes to start sensing to avoid violation of the error bound.

6.3 The Impact of Training Period Length

In this experiment, we evaluate the influence of different length of training period on the error rate and on the energy conservation. When the node starts sensing, an initialization period is required to build both the correlation table and the prediction model. The accuracy of the prediction model depends on the sample sizes of data feed into the model constructor. As we can see in Fig. 16, the error rates decrease as the length of the training period become longer. The results become more obvious as the error tolerance e_t is larger. This can be explained in that the scheduling algorithm has more flexibility to adjust the duty cycle as the error tolerance gets larger. Notice that the energy consumption also increases with the increase of training period length. According to our experiment, the energy consumption increases from 27 to 38 percent as the length of training period goes from 2.3 to 12 percent during the simulation. As a result, the system exhibits a tradeoff between data accuracy and energy conservation.

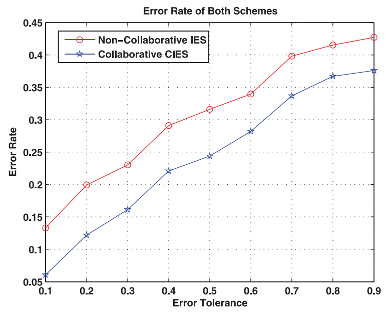


Fig. 10. The error performance with different error tolerance.

6.4 The Detailed Analysis of Neighbor Error Control

In the first example, there are only two sensors adjacent to each other. This example shows the performance of the simplest case of *CIES*.

Fig. 10 illustrates the error performance of noncollaborative and collaborative *CIES*. Over error tolerances, we can see that both approaches can satisfy the error performance requirement. However, under the collaborative *CIES* scheme, the error rate is at least 20 percent less than with stand-alone *IES*.

Fig. 11 demonstrates the metric of miss ratio for stand-alone *IES* and *CIES*. In the extreme region, (i.e., the error tolerance is 10 percent), the miss ratio is about 25 percent less with the collaborative information. The purpose is well demonstrated here.

Fig. 12 demonstrates the energy consumption for both schemes. Compared to the 25 percent error rate improvement in *CIES*, the additional energy consumption is small as the maximum difference between the two schemes is only 5 percent.

From the above three figures, we can conclude that the *CIES* method is superior to the stand-alone *IES* scheme with slightly more energy consumption. This cost can be traded for the reduction in miss ratio, which has been considered important in certain monitoring applications [2].

6.5 The Impact of Node Density

In the second example, we raise the node density to 20 and investigate the effect of node density to performance

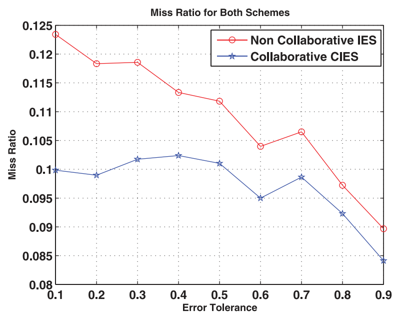


Fig. 11. The miss ratio with different error tolerance.

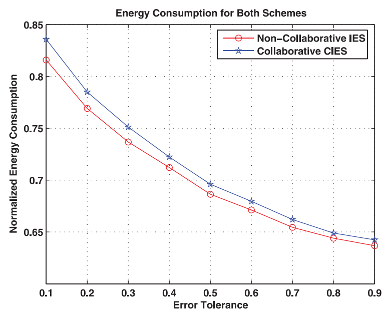


Fig. 12. The energy consumption with different error tolerance.

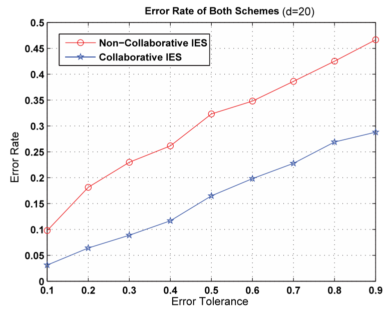


Fig. 13. The error performance with different error tolerance.

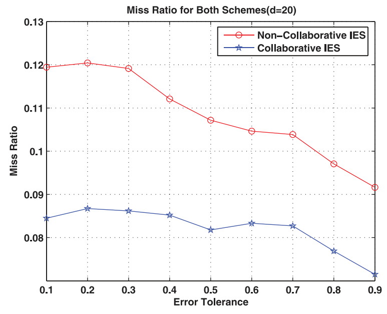


Fig. 14. The miss ratio with different error tolerance.

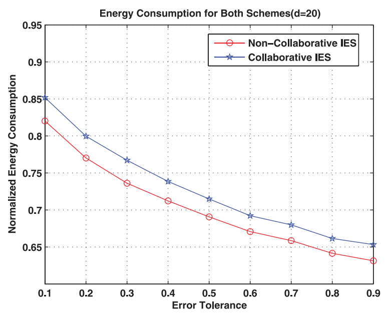


Fig. 15. The energy consumption with different error tolerance.

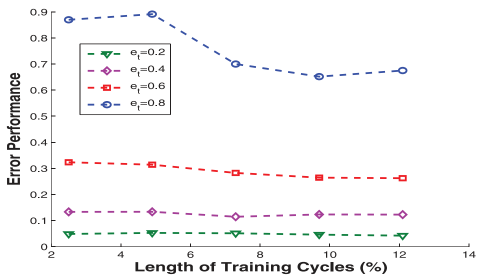


Fig. 16. The influence of training period length on the prediction error rate.

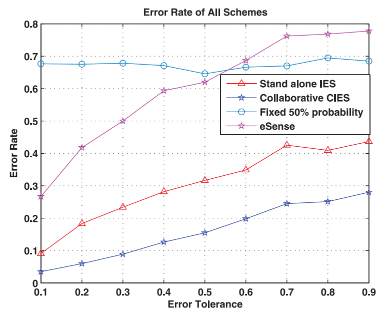


Fig. 17. The error performance with different error tolerance.

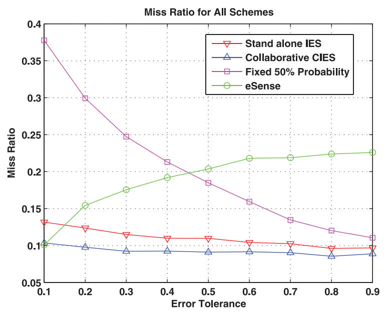


Fig. 18. The miss ratio with different error tolerance.

metrics. Figs. 13, 14, and 15 show the error rates associated with each approach. Compared to the results in Fig. 10, the error rates are dramatically reduced while the gap between the two schemes is increased. This is expected because there is greater chance for the sensor to be awakened by neighbor nodes.

From the miss ratio performance result, we can draw a similar conclusion. The miss ratio is reduced almost by 45 percent for different levels of error tolerance. This shows the validity of collaborative *IES*.

However, as shown in Fig. 15, the energy consumption is slightly higher but still acceptable considering the improvement in miss ratio. It is the application's choice to balance the tradeoff between energy consumption and other performance metrics.

We also compare the performance of our approaches to the state-of-the-art eSense approach [8] (eSense has only local error control) and an additional benchmark. The benchmark is to set with a random 50 percent probability rate for P_m for each time instance m. We implement the principle of eSense to control the probability for miss ratio performance into our simulation system. The performance results of all approaches are demonstrated below:

- *The error rate comparison.* The error performance is demonstrated in Fig. 17. As we can see, the benchmark is not affected too much by the setting of error tolerance. Although the error rate for eSense does not increase too much due to the increase of error tolerance, its error rate is much higher than both *IES* and *CIES* approaches. Note that the error rate determined by eSense cannot satisfy the error rate boundaries while both stand-alone *IES* and collaborative *CIES* approaches meet the requirement.
- *The miss ratio comparison.* The miss ratio, as shown in Fig. 18, for eSense continues to increase with the increase of error tolerance, while our methods seem to be stable. The hike in eSense is due to the reduction in sensitivity to the change when the threshold or risk tolerance increases. The higher

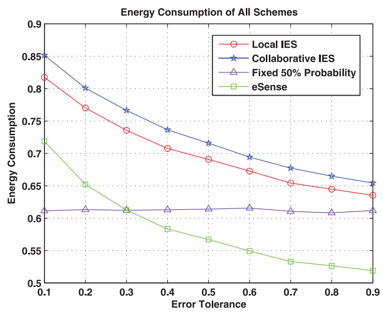


Fig. 19. The energy consumption with different error tolerance.

the risk tolerance, the less sensitive eSense is to the data change, leading to a higher miss ratio. Our system can guarantee the absolute error rate, which keeps tracking the past error and adjusts accordingly. Therefore, our approaches do not experience a similar hinderance.

- *The energy consumption comparison.* The energy consumption shown in Fig. 19 indicates that eSense consumes slightly less than 15 percent of that of our *CIES*. However, the stability of our system is much better than eSense as the maximum variance of energy consumption is just 12 percent compared to almost 100 percent in eSense. These achievements only cost 20 percent more energy consumption than eSense, which is acceptable in most applications. In some situations, the rare event detection is as important as the lifetime management.

To see the performance compare more directly, we list the performance metrics of our design with those reported in *eSense* [8]. The results are shown in Table 2.

Based on comparisons, we conclude that *CIES* can outperform the eSense with respect to the miss ratio and total error rates. The results also confirm that the error-bounded scheduling limitation, which is missed in eSense, is achieved in our approach.

6.6 System Implementation

The architecture has been implemented on our newly constructed testbed and the details are given in Section 5 of the online supplemental material.

7 Related Work

Scheduling control has been an effective method in wireless sensor platforms to improve energy efficiency. It allows networked nodes to reduce their transmitting and sensing power while preserving sensing quality. A common technique to reduce power is local data control, for example, reducing the duty cycle (the active duration) of sensors by turning them off while using a prediction model to estimate the actual data at each sensor node [25], [26]. In general, a higher duty cycle leads to a better prediction accuracy, but consumes more energy. In contrast, a reduced duty cycle is preferred due to its lower energy consumption and reduced data traffic within the network, but this may decrease the prediction accuracy. Methods in [8], [27] further introduce a self-adaptive scheduling mechanism to address the data accuracy issues. We acknowledge that these technologies do provide frameworks to manage the energy cost without comprising data accuracy. However, these design approaches are isolated in that the management of data accuracy can only be used for each individual sensor. Thus, these techniques do not fully exploit the potential of network collaboration for data accuracy management under rapidly changing conditions in the environment, and their ability to tradeoff data accuracy with energy reduction is thus inherently limited [14], [28], [29]. Our work incorporates the advantages of both dynamic scheduling and sensors collaboration, seeking to optimize the tradeoff between energy and the accuracy of predictions.

Traditionally, complex and dedicated models have been used in local data control, aiming to determine if the model is accurate enough to ensure high precision [30], [31], [32], [33]. In [31], empirical analysis results are used to reveal the relationship between the configuration parameters and the quality of the tracking application. It shows that empirical models [14], [28] can be effectively applied without sacrificing the data prediction accuracy. One useful insight from an empirical modeling approach is that data correlation can be utilized for other purposes [34], [35], such as monitoring applications. In eSense [8], a stochastic sensing algorithm that computes the sensor switching probability at each sampling cycle is introduced. It determines the lower and upper bounds of sensing probability to satisfy missing ratio constraints, a metric to determine the percentages that the prediction model output does violate the data performance requirement. In actual situations, however, this kind of approach cannot necessarily characterize the volatile nature of the environment, caused by the insensitivity of the prediction model to sharp changes in natural environment [36], thus leading to inefficient data prediction.

Another category of data control is the implementation of a distributed data management scheme. Data aggregation approaches have been widely acclaimed as useful techniques to reduce communication overhead in sensor networks [26], [37]. However, there has been little cooperation between sensor nodes. Although those approaches offer data management mechanisms which reduce the error and energy cost of sensing activities, they fail to improve the system performance through network coordination. A collaborative mechanism among nodes, together with local data management, can provide the opportunity to

TABLE 2
The Performance Comparison with eSense

Algorithms	MR($e_t = 10\%$)	EC($e_t = 10\%$)	MR($e_t = 20\%$)	EC($e_t = 20\%$)	MR($e_t = 30\%$)	EC($e_t = 30\%$)
eSense(dynamic)	0.09	0.71	0.14	0.67	0.18	0.62
IES	0.09	0.82	0.098	0.76	0.101 s	0.72
CIES	0.075	0.85	0.077	0.80	0.081 s	0.76

accurately associate networking nodes for higher data accuracy and increased network capability, for example, the detection of a rapid environmental change.

8 Conclusions

In this paper, we have presented a stochastic sensing algorithm to reduce energy consumption. Our approach used the data correlation between nodes to reduce the error rate for prediction model performance. Observed correlations between nodes have been used to estimate the neighbor nodes' errors, and to adjust their operation accordingly. We demonstrated that our design achieved better control of data accuracy than baseline approaches and still retained the energy saving properties. The measurement and simulation results showed that system prediction error remained within a specified error tolerance while saving up to 60 percent of the required energy. For our future work, we will evaluate the energy performance of individual sensor network components so that the algorithm can be further optimized.

Acknowledgments

A preliminary version of this paper was presented in the 7th International Conference on Mobile Ad-hoc and Sensor Networks, Beijing, China, 2011 and has been awarded the best paper award [1]. Research was supported under Grant: US National Science Foundation (NSF) grants CNS-0845994, CNS-0917097, SUTD-ZJU/RES/03/2011, Singapore-MIT IDG31000101, and IBM OCR fund.

References

[1] Q. Zhang, Y. Gu, L. Gu, Q. Cao, and T. He, "Collaborative Scheduling in Highly Dynamic Environments Using Error Inference," *Proc. Seventh Int'l Conf. Mobile Ad-hoc and Sensor Networks (MSN '11),* 2011.

[2] T. He, S. Krishnamurthy, L. Luo, T. Yan, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J.A. Stankovic, T.F. Abdelzaher, J. Hui, and B. Krogh, "VigilNet: An Integrated Sensor Network System for Energy-Efficient Surveillance," *ACM Trans. Sensor Networks,* vol. 2, no. 1, pp. 1-38, 2006.

[3] H. Luo, J. Wang, Y. Sun, H. Ma, and X. Li, "Adaptive Sampling and Diversity Reception in Multi-Hop Wireless Audio Sensor Networks," *Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS '10),* 2010.

[4] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," *Proc. First Int'l Conf. Embedded Networked Sensor Systems (SenSys '03),* 2003.

[5] X. Bai, C. Zhang, D. Xuan, and W. Jia, "Full-Coverage and k-Connectivity $(k = 14, 6)$ Three Dimensional Networks," *Proc. IEEE INFOCOM '09,* 2009.

[6] Z. Yun, X. Bai, D. Xuan, T.H. Lai, and W. Jia, "Optimal Deployment Patterns for Full Coverage and k-Connectivity $(k\ leq\ 6)$ Wireless Sensor Networks," *Proc. IEEE INFOCOM '09,* 2009.

[7] Z. Zhou, S. Das, and H. Gupta, "Connected k-Coverage Problem in Sensor Networks," *Proc. 13th Int'l Conf. Computer Comm. and Networks (ICCCN '04),* 2004.

[8] H. Liu, A. Chandra, and J. Srivastava, "Esense: Energy Efficient Stochastic Sensing Framework for Wireless Sensor Platforms," *Proc. Fifth Int'l Conf. Information Processing in Sensor Networks (IPSN '06),* 2006.

[9] C. Chong and S.P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges," *Proc. IEEE,* vol. 91, no. 8, pp. 1247-1256, Aug. 2003.

[10] *Atmega128l Datasheet,* http://www.atmel.com/dyn/resources/, 2008.

[11] *Image sensor,* http://www.aptina.com/products/image_sensors/mt9m034/, 2013.

[12] *Micro-Power Impulse Radar,* http://www.datasheetarchive.com/MIR-800-datasheet.html, 2013.

[13] J. Polastre and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *Proc. Second Int'l Conf. Embedded Networked Sensor Systems (SenSys '04),* 2004.

[14] C. Lin, Y. He, and N. Xiong, "An Energy-Efficient Dynamic Power Management in Wireless Sensor Networks," *Proc. Fifth Int'l Symp. Parallel and Distributed Computing (ISPDC '06),* 2006.

[15] Q. Zhang, Y. Gu, T. He, and G.E. Sobelman, "Cscan: A Correlation-Based Scheduling Algorithm for Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Networking, Sensing and Control (ICNSC '08),* pp. 1025-1030, 2008.

[16] S.M. George, W. Zhou, H. Chenji, M.G. Won, Y.O. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, "Distressnet: A Wireless Ad Hoc and Sensor Network Architecture for Situation Management in Disaster Response," *IEEE Comm. Magazine,* vol. 48, no. 3, pp. 128-136, Mar. 2010.

[17] Z. Ruan, E.C.H. Ngai, and J. Liu, "Wireless Sensor Deployment for Collaborative Sensing with Mobile Phones," *Computer Networks,* vol. 55, no. 15, pp. 3224-3245, 2011.

[18] B. Choi, H. Liang, X. Shen, and W. Zhuang, "DCS: Distributed Asynchronous Clock Synchronization in Delay Tolerant Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 23, no. 3, pp. 491-504, Mar. 2012.

[19] Q. Li and D. Rus, "Global Clock Synchronization in Sensor Networks," *IEEE Trans. Computers,* vol. 55, no. 2, pp. 214-226, Feb. 2006.

[20] Z. Zhong, P. Chen, and T. He, "On-Demand Time Synchronization with Predictable Accuracy," *INFOCOM '11,* pp. 2480-2488, 2011.

[21] V. Rajamani and C. Julien, "Blurring Snapshots: Temporal Inference of Missing and Uncertain Data," *Proc. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '10),* pp. 40-50, 2010.

[22] R. Olfati-Saber, "Distributed Tracking for Mobile Sensor Networks with Information-Driven Mobility," *Proc. Am. Control Conf. (ACC '07),* pp. 4606-4612, 2007.

[23] R. Graham and J. Cortes, "Spatial Statistics and Distributed Estimation by Robotic Sensor Networks," *Proc. Am. Control Conf. (ACC '10),* pp. 2422-2427, 2010.

[24] *Wisconsin-Minnesota Cooperative Extension Agricultural Weather Page,* Univ. of Minnesota, http://www.soils.wisc.edu/wimnext/, 2008.

[25] A. Mainwaring, J. Polastre, R. Szewczyk, D.E. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," *Proc. ACM Workshop Sensor Networks and Application (WSNA '02),* 2002.

[26] Y. Wu, X.Y. Li, Y.H. Liu, and W. Lou, "Energy-Efficient Wake-Up Scheduling for Data Collection and Aggregation," *IEEE Trans. Parallel and Distributed Systems,* vol. 21, no. 2, pp. 275-287, Feb. 2010.

[27] R. Szewczyk, A. Mainwaring, and D. Culler, "An Analysis of a Large Scale Habit Monitoring Application," *Proc. Int'l Conf. Embedded Networked Sensor Systems (SenSys '04),* 2004.

[28] R.M. Passos, C.J.N. Coelho, A.A.F. Loureiro, and R.A.F. Mini, "Dynamic Power Management in Wireless Sensor Networks: An Application-Driven Approach," *Proc. Second Ann. Conf. Wireless On-demand Network Systems and Services (WONS '05),* 2005.

[29] H.C. Shih and K. Wang, "An Adaptive Hybrid Dynamic Power Management Method," *Proc. IEEE Int'l Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing,* 2006.

[30] R. Mangharam, R. Rajkumar, S. Pollin, F. Catthoor, B. Bougard, L.V. der Perre, and I. Moeman, "Optimal Fixed and Scalable Energy Management for Wireless Networks," *Proc. IEEE INFOCOM '05,* 2005.

[31] Q. Qiu, Q. Wu, D. Burns, and D. Holzhauer, "Lifetime Aware Resource Management for Sensor Network Using Distributed Genetic Algorithm," *Proc. Int'l Symp. Low Power Electronics and Design (ISLPED '06),* pp. 191-196, 2006.

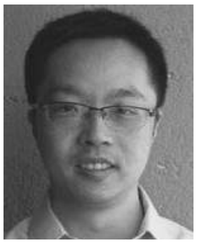
[32] Y. Zhang, L. Zhang, and X. Shan, "Robust Distributed Localization with Data Inference for Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Comm. (ICC '08),* 2008.

[33] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links," *Proc. MobiCom '09,* 2009.
[34] S. Pattem, B. Krishnmachari, and R. Govindan, "The Impact of Spatial Correlation on Routing with Compression in Wireless Sensor Networks," *Proc. Third Int'l Symp. Information Processing in Sensor Networks (IPSN '04),* 2004.
[35] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On Network Correlated Data Gathering," *Proc. IEEE INFOCOM '04,* 2004.
[36] S. Goel and T. Imielinski, "Prediction-Based Monsitoring in Sensor Networks: Taking Lessons from MPEG," *ACM Computer Comm. Rev.,* vol. 31, no. 5, pp. 82-98, Oct. 2001.
[37] A. Sinha and A. Chandrakasan, "Toward Optimal Data Aggregation in Random Wireless Sensor Networks," *Proc. IEEE INFOCOM '07,* 2007.

Qingquan Zhang graduated with the PhD degree from Department of Electrical and Computer Engineering at University of Minnesota, Twin Cities in 2008. He is a research scientist at Lemko Corporation. His research interest include wireless sensor network, telecommunication systems especially 3G, 4G-LTE systems. He is a senior member of the IEEE.

Lingkun Fu is working toward the PhD degree as a member of the Group of Networked Sensing and Control (IIPC-nesC) in the State Key Laboratory of industrial control technology at Zhejiang University. His research interest include optimization for wireless rechargeable sensor networks. He is a student member of the IEEE.

Yu (Jason) Gu received the PhD degree from the University of Minnesota, Twin Cities, in 2010. He is currently an assistant professor in the Pillar of Information System Technology and Design at the Singapore University of Technology and Design. He is the author and coauthor of more than 40 papers in premier journals and conferences. His publications have been selected as graduate-level course materials by more than 20 universities in the United States and other countries. His research include networked embedded systems, wireless sensor networks, cyber-physical systems, wireless networking, real-time and embedded systems, distributed systems, vehicular adhoc networks, and stream computing systems. He is a member of the IEEE and the ACM.



Lin Gu received the BS degree from Fudan University, the MS degree from Peking University, and the PhD degree in computer science from the University of Virginia. He is an assistant professor at the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology (HKUST). His research interest include cloud computing and wireless sensor networks.

Qing Cao is an assistant professor in EECS Department at the University of Tennessee. His research include wireless sensor networks, operating systems, and mobile systems. He has authored more than 30 papers in these areas with more than 1,600 citations. He is a member of the IEEE.

Jiming Chen is a full professor in Department of Control at Zhejiang University. His research interests include estimation and control over sensor network, sensor and actuator network, coverage, and optimization in sensor network. He has authored more than 80 papers in these areas. He is a senior member of the IEEE.

Tian He received the BSc and PhD degrees in control science and engineering from Zhejiang University in 2000 and 2005, respectively. He received the PhD degree under professor John A. Stankovic from the University of Virginia, Virginia in 2004. He is an associate professor in CSE Department at the University of Minnesota. He has received a number of research awards in the area of sensor networking, including five best paper awards (MSN 2006, SASN 2006, MASS 2008, MDM 2009, and MSN 2011). His research interest include wireless sensor networks, cyber-physical systems, and real-time embedded systems. He has authored more than 100 papers in these areas with more than 9,000 citations (H-index 38). He is a senior member of the IEEE.