Cscan: A Correlation-based Scheduling Algorithm for Wireless Sensor Networks

Qingquan Zhang, Yu Gu, Tian He and Gerald E. Sobelman

***Abstract*— Dynamic scheduling management in wireless sensor networks is one of the most challenging problems in long lifetime monitoring applications. In this paper, we propose and evaluate a novel data correlation-based stochastic scheduling algorithm, called *Cscan*. Our system architecture integrates an empirical data prediction model with a stochastic scheduler to adjust a sensor node's operational mode. We demonstrate that substantial energy savings can be achieved while assuring that the data quality meets specified system requirements. We have evaluated our model using a light intensity measurement experiment on a Micaz testbed, which indicates that our approach works well in an actual wireless sensor network environment. We have also investigated the system performance using Wisconsin-Minnesota historical soil temperature data. The simulation results demonstrate that the system error meets specified error tolerance limits and up to a 70 percent savings in energy can be achieved in comparison to fixed probability sensing schemes.**

I. Introduction

Wireless Sensor Networks (WSNs) have been used in many application domains [1], [2], [3], [4]. Due to the limited power supply and difficulties in harvesting ambient energy, low power energy management is a critical research issue. Energy consumption for the sensing operation dominates the lifetime of a sensor network. Therefore, it is important to design protocols which minimize the amount of sensing required by the sensor nodes. In the past few years, many solutions have been proposed for energy conservation by applying different power switching strategies (e.g. [5]) in which hardware components such as CPU and memory can operate with different power modes. Other semantic-based efforts, such as TAG [6], focus on reducing the sensing and communication load. Even though those methods show some interesting results, there is a need for improvement in several directions. Moreover, most real-time power control protocols have no robust error control guarantee mechanism.

In this paper, we propose a systematic dynamic sensing scheduling algorithm, called *Cscan*, specifically for long lifetime applications such as military surveillance or habitat monitoring. The key idea of our framework is to activate a sensor during cycles in which there is a high probability that the model's prediction would exceed a specified error tolerance. Our approach builds on the observation that data sensed and collected by sensor networks over time may exhibit similar data patterns and the data disseminated over time could be well correlated. The key techniques used in our approach are: 1) the construction of a data prediction model, i.e. an empirical model which captures the prominent features of the data collected over time, and 2) an error-sensitive stochastic scheduling algorithm. This methodology allows sensor nodes to remain predominantly inactive, while achieving a high data integrity. As we will present in this paper, our contributions can be summarized as follows:

- We present a new energy-efficient scheduling algorithm that includes a very accurate but hardware-friendly prediction model to capture recent data trends.
- We introduce the concept of error implication, which exploits data correlations among multiple sensing cycles over a given time period.
- We provide an extensive experimental study of our framework using real data sets from different domains and compare our results against the most commonly accepted data aggregation approach. We also implement our algorithm into the sensor network we built for a light intensity monitoring application. Our experiments demonstrate that our algorithm can save up to 70 percent of the energy while still meeting the error rate requirement.

II. Overview and Objectives

The strategies exploited in our $Cscan$ framework are specifically developed for long-term environmental monitoring applications in which energy conservation and data accuracy are of most interest. The system should try to avoid any unnecessary sensing and data acquisition while assuring acceptable data quality, as defined by the application. The system performance is quantified by defining three criteria: the miss ratio, which denotes the fraction of scheduling cycles that the system fails to present acceptable prediction data, the energy consumption and the data sample error rate.

To successfully achieve our energy and error control objectives, a data management scheme is investigated and integrated into the system. The architectural framework is shown in Figure 1. Those functional blocks will support the following key features:

A. Prediction model construction

We seek to identify correlated sensor data patterns in a sensing period in order to predict sensing data over time. The sensors' sampling data are fed into the model constructor during the initialization training stage and an empirical prediction model is created. After that, model constructors keep updating the model whenever new sensing data becomes available.

B. Duty-cycle optimization

This is an approach to manage the power consumption and the prediction error rate in a given cycle.

Qingquan Zhang and Gerald E. Sobelman are with Department of Electrical and Computer Engineering, University of Minnesota, Twin Cities, USA zhan0511@umn.edu, sobelman@umn.edu

Yu Gu and Tian He are with Department of Computer Science and Engineering, University of Minnesota, Twin Cities, USA yugu@cs.umn.edu, tianhe@cs.umn.edu

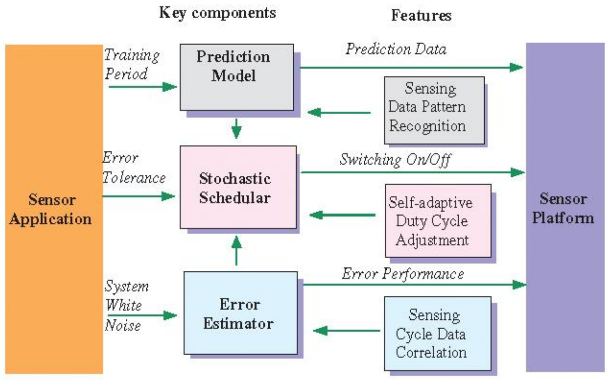

Fig. 1. The architecture of $Cscan$.

C. Error estimator

The error estimator will serve to ensure that the operation of a sensor is such that the data quality requirement is not violated. We must create a balance between energy savings and the rate of prediction errors.

III. Data Prediction Algorithm

A sensor can lower its operating duty cycle, meaning it can switch into a sleep state to conserve energy. This operation is based on the fact that the sensor's readings may form a recognizable pattern during certain periods, especially in the case of environmental monitoring applications. Those patterns can be well approximated and used for predicting future readings if the specific application is well understood. The system will start building the prediction model in the initialization phase. Then, we separate the sensor's operation into a data resampling phase and a prediction phase. In the data resampling phase, we use the latest sampling data to update the model built during the training cycle. In the prediction phase, the node will switch off to conserve energy and the predicted sensing results are generated by the predictor which has been updated in the resampling phase.

1) Empirical Model Construction: An empirical model is used to find strong correlations in the data and to arrange them in a certain way so that future data can be extracted from the empirical or historical data. Depending on the duration of the system and the data accuracy requirements of an application, the empirical models can be constructed in different ways. Here we introduce an hourly-based empirical model, as shown in Figure 2 During a data training cycle, initial sensing differences between two adjacent hours are calculated and updated throughout the training cycle. A weighted moving average method is used to smooth the data. For example, if the sensed temperature data at 1 AM and 2 AM are 20 degrees F and 22 degrees F respectively, then the difference between 1 AM and 2 AM is 2 degrees F. At the end of the training cycle, a model is constructed such that the sensing data difference between any two adjacent hour times can be estimated at the sensor node.

2) Prediction Model Update: Once the sensor is in the resampling phase, the system will not only get precise readings but can also refresh the empirical model parameters. The system compares the prediction values produced by the empirical model with the real sensing data. If the difference is below the specified error tolerance level, the system is regarded as good ("hit") and the prediction model can be used. This can be expressed in the following equation:

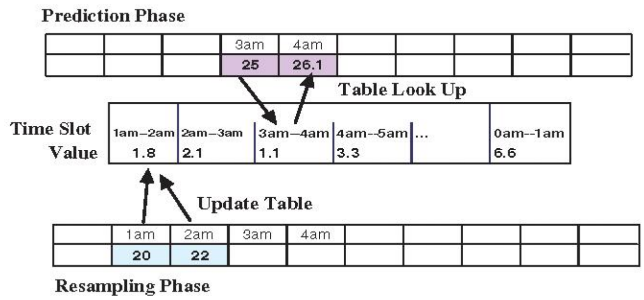

Fig. 2. Construction of the empirical prediction model.

$$ABS(\frac{V_p - V_r}{V_r}) \leq e_t \tag{1}$$

V_p is the value output from the estimator, V_r is the true sensed data and e_t is the error tolerance level that can be accepted, as specified by the user.

A system corrective action will be taken to update the empirical model by refreshing the original model with the latest results for $V_r(k)$ and $V_r(k-1)$.

Compared to a regression model, the advantage of using an empirical model in this application domain is that it simplifies the processing requirements while providing a reliable reference for prediction. As a result, the hardware cost can be minimized. Moreover, data resampling helps to update the predictor's model parameters when the sensor nodes are in a dormant state.

IV. Scheduling Algorithm

In this section, we present our scheduling algorithm that includes the underlying data prediction model and the data quality requirements to control the sensing/resampling of the sensors. We seek to minimize the sensor energy consumption according to different system operation modes while satisfying the data quality constraint.

A. Our problem formulation

In order to conserve their limited power supply, the sensors do not continuously sense data but rather operate only during certain cycles as long as acceptable data quality can be met. The scheduling can be adjusted based on the algorithm that we will elaborate later. We assume that the baseline sensor operation sequence consists of N data cycles, which include k cycles used for training. In each cycle i, the probability that the sensor will be active is defined as p_i. We further assume the average energy consumption for sensing (the energy cost of a node to sense, process and communicate) is E_a, the defined prediction error tolerance is e_t and the potential error at each cycle due to inactive sensor status is e_i. Therefore, the goal of our design is to minimize the energy consumption during each baseline period:

$$E = k \cdot E_a \cdot t_u + E_a \cdot t_u \cdot \sum_{i=1}^{N-k} p_i \tag{2}$$

under the constraint that

$$\frac{\sum_{i=1}^{N} (1-p_i) \cdot e_i}{N} = \frac{\sum_{i=k+1}^{N} (1-p_i) \cdot e_i}{N} \leq e_t \tag{3}$$

where t_u is the unit cycle length and e_t is the error tolerance set by the design requirements. The constraint will enforce that the potential statistical error caused by the prediction will be less than the error tolerance. The range of possible values of p_i will be bounded to satisfy the constraint equation.

B. Adaptive scheduling algorithm

The minimization of energy consumption deals with several key issues, e.g. the length of the training cycle and the prediction model used. The goal of the scheduling algorithm is to find the appropriate p_i for a given error range e_i obtained from past data values. To solve for p_i at a specific e_i requires a joint distribution of a process for e_i at a specific time instance or period. This would require a heavy computational capability and storage burden on the limited resources of the sensor node. Obtaining a solution for p_i will be extremely difficult to calculate during transitions. Instead, we introduce a simpler method for computation that allows the sensor to choose the value within a range. We first determine the boundary for p_i, and the scheduling algorithm will choose one value within the boundary according to a node's operational status. It should be clear that the higher the value of p_i, the larger the expected energy consumption. The lower the value of p_i, the higher the chance that the error due to prediction will be greater than the tolerance. Therefore, analysis of the boundary of p_i will be investigated to optimize this trade-off.

C. Determining the Sensing Probability Boundary

We use a bottom-up approach to set a boundary the for sensing probability. That is, we will not violate the constraint equation during each cycle instance so that the sum of all cycle error products $(1 - p_i) \cdot e_i$ will not violate the constraint. As noted, this decision sets a stricter requirement than the constraint equation over all sampling instances. Therefore, 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

$$p_i^{lb} = \begin{cases} 0 & 0 \leq e_i \leq e_t \\ 1 - \frac{e_t}{e_i} & e_t \leq e_i \leq 1 \end{cases} \tag{4}$$

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

D. The Selection of Sensing Probability

In this section, we focus on choosing the appropriate value of p_i which requires us to determine the effective error estimation e_i at cycle i. To accurately evaluate the e_i, we included two categories of errors that we believe make up the major contribution to the possible errors in prediction. The first category is the system intrinsic error, τ, that takes into account all the system white noise and environmental instability within the system. The other category is the implied error, e_{im}, which is a function of the data correlations. Our technique relies on adaptive sensing to adjust both of them. The sensors keep a record of past sensing data, comparing the authentic recorded data with the outputs from the prediction models constructed. The error rate will then be fed back to the sensor operation platform where processing of two error categories will take place. A probability estimation algorithm (Algorithm 1) is called during the initialization and update procedures of the sensing operation to select the probability value. The algorithm takes the error tolerance e_t, initialized intrinsic error τ_i and the implied error e_{im} as inputs. The input variables will be used to choose the corresponding probability from the available rate ranges as described in Algorithm 1.

Algorithm 1 Probability Determination Algorithm

Require: e_t,τ_i,e_{im}

1: Determine the boundary of p_i from section IV-C
2: **if** $e_{i0} < e_{im}$ **then**
3: $\quad e_j = e_{im}$
4: **else**
5: $\quad e_j = \tau_i$
6: **end if**
7: Achieve the value of p_i from the constraint equation
8: return p_i

In this algorithm, we choose the higher error estimation between the intrinsic error τ_i and implied error e_{im}. A high error rate indicates environmental instability or a poor prediction model outcome while a low value signals a potential to cut down the resampling rate for energy conservation purposes. However, since τ_i and e_{im} will change over time, a mechanism is necessary to estimate them in an adaptive manner.

1) Update intrinsic error τ_i: The intrinsic error τ_i represents the information about the prediction instability of data at cycle i. A high value indicates a greater chance that the prediction model will fail in estimating the real value. We update τ_i whenever the sensor node switches on at that cycle by using a moving average:

$$\tau_i = \alpha \cdot e_s + (1 - \alpha) \cdot \tau_i \tag{5}$$

where e_s is the error between the predicted value and the actual recorded data when the sensor switches on. In our experiments, we choose α to be 0.5. As we can see, if the prediction model outputs a lower error data value in comparison to the real value, the new τ_i will become smaller.

2) Achieving the implied error e_{im}: The implied error e_{im} is obtained from the correlation coefficient between the current cycle and the latest cycle in which the sensor node switched on. It can be expected that if the two cycles have a strong correlation, the error in one cycle can be well estimated from the correlated cycle. Otherwise, two cycles having low correlation will have a high potential error mismatch. Therefore, the sensor platform must take this into account when determining whether the sensor node needs to switch on. Based on our observation, the implied error can be well estimated as:

$$e_{im} = \frac{A \cdot e_j}{C_{ij}} \tag{6}$$

where A is a constant, e_j is the latest measured error when the sensor node switched on at cycle j and C_{ij} is the correlation coefficient between the current cycle i and cycle j. The procedure

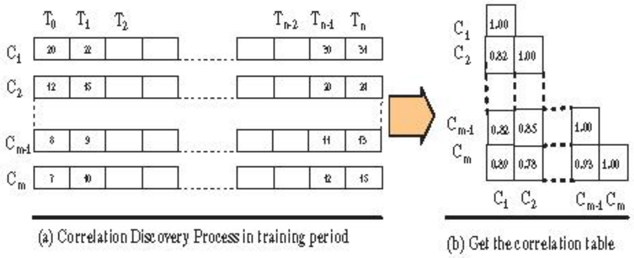

Fig. 3. Construction of the table of correlation coefficients.

Fig. 4. Testbed and snapshot of experimental data events (inset shows the light pattern).

for constructing the correlation coefficient table e_{im} is illustrated in Figure 3.

It should be noted that that for simplification purposes, we assume that correlation among cycles remain relatively stable throughout each operation.

V. Evaluation

A. System Implementation

The architecture has been implemented on our newly constructed test-bed, shown in Figure 4, with more than 100 sensor nodes which provides a realistic controllable environment for design verification and performance evaluation. The design has been implemented on a Berkeley TinyOS/Micaz platform. Sensor nodes are placed randomly over the board, giving us a better reflection of the sensing algorithm. Both random and controlled scanning light patterns are created to emulate the light intensity change in environment and then projected onto the test-bed with three projectors switching on simultaneously. The sensed data is recorded and processed according to our sensing algorithm. The evaluation results (e.g. error rate, energy conservation vs. error tolerance) allow further analysis to optimize the overall system.

B. System Evaluation

In order to test the performance of the proposed $Cscan$ algorithm, especially for error control in terms of energy conservation, we have conducted a series of experiments to track the sensor status on our test-bed. Different light intensity patterns are projected onto the test-bed to emulate various environment conditions. The sensor nodes detect the light intensity and dynamically process those values using $Cscan$. A period of 1000 cycles (corresponding to 1000 sample points) was selected for each run. Each run was repeated multiple times with different parameter settings, such as the length of the training cycle. Figure 5 shows the resulting dynamic energy consumption. As seen in the figure, $Cscan$ does not conserve energy in the initialization period during which the prediction model and inter-cycle correlations are built. After that, however, the energy consumption is reduced, as desired. We can also observe in the figure that the energy conservation in certain cycles remains at a flat level, corresponding to those times in which the sensor node is in its prediction mode. The energy consumption as a function of error tolerance is shown in Figure 6. Three sets of results representing different experimental scenarios are presented. The first scenario is one in which a sensor randomly switches on/off with probability 50 percent. The second scenario is where the sensor has a 90 percent probability of being active in every cycle. In the third scenario, sensors operate according to the $Cscan$ scheduling algorithm. We can see that energy conservation reaches above 70 percent when the error tolerance e_t is relatively high. $Cscan$'s energy conservation is less than that in the random case when e_t is low, implying that the sensors have a higher chance of switching on if the environment is not stable. The error performance results are presented in Figure 7. These results also show that $Cscan$ can control the error rate according to system requirements and that the $Cscan$ algorithm can be practically and effectively implemented.

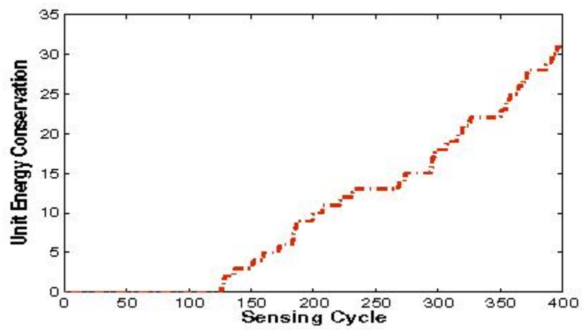

Fig. 5. One sample of dynamic energy conservation in a sensor.

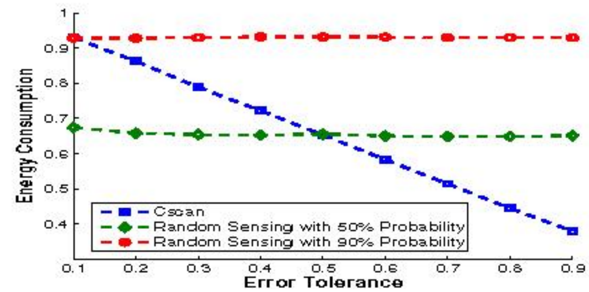

Fig. 6. The measured energy consumption of $Cscan$ vs. other strategies.

C. Emulation Setting

To evaluate the performance of the $Cscan$ sensing strategies in a real application, a simulation program with historical soil temperature data was developed. The data was collected from the Wisconsin-Minnesota Cooperative Extension Agricultural Weather Page where soil temperature is monitored regularly. The soil temperature is sampled twice per hour, 24 hours per day. This full record of soil temperature data over the past 10 years allows us to extensively test the efficiency of our strategy.

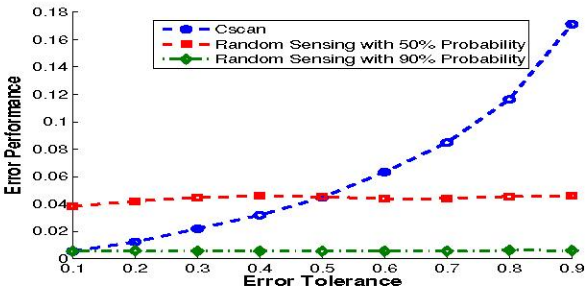

Fig. 7. The measured sensor error performance of $Cscan$ vs. other strategies.

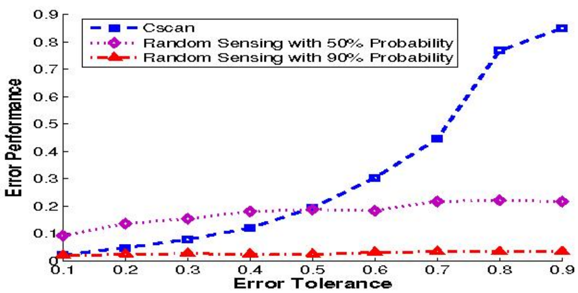

Fig. 8. The influence of error tolerance on the average error.

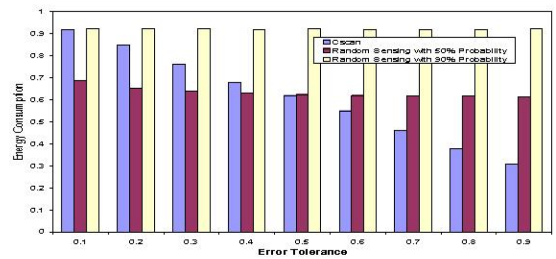

Fig. 9. The influence of error tolerance on energy consumption.

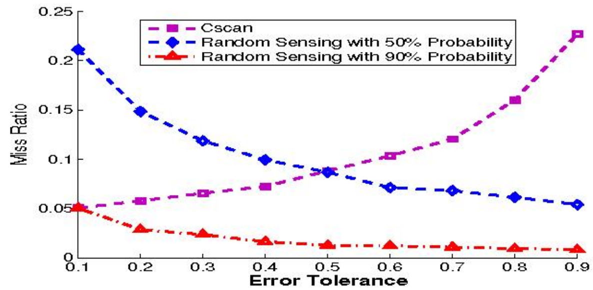

Fig. 10. The influence of error tolerance on the prediction miss ratio.

By doing so, we can reduce the randomness and investigate the impact of different configurations on the performance of energy conservation and error control.

D. Performance Analysis

In this section, we evaluate the error rates as a function of some key design parameters, including the effect of error tolerance and the length of the training period. A study of the effects of these parameters can provide insights into methods for improving system performance. We begin the evaluation by measuring the error rate of the $Cscan$ system. Then, we compare the energy conservation for different parameter values. Finally, we study the miss ratio (defined as sensor prediction results which violate the error tolerance requirement) performance for our adaptive scheduling algorithm.

1) Impact of Error Tolerance: During this evaluation, the level of error tolerance varies from 10 percent to 90 percent while the length of the training period was kept constant at 12 percent of the total number of simulation cycles. The total number of cycles is approximately 9000, which corresponds to more than one year of data. This data set is large enough to significantly reduce unsystematic errors caused by limited sample size. Figure 8 shows the estimated error rate as a function of error tolerance under different scenarios. We measure the average prediction error of the estimator in scenarios 1 and 2, as described earlier. It turns out that the prediction error in scenario 1 is about 40 percent for most of the error tolerance levels, which means that little improvement in energy consumption is achieved in this case. Also, the error rate is low for scenario 2, as expected. As suggested from the simulation results, the prediction error for the $Cscan$ algorithm increases in proportion to the increase of error tolerance. Most importantly, the prediction error from $Cscan$ met the requirements in almost all cases. For example, $Cscan$'s data error rate is only 20 percent when the system error tolerance is 50 percent. This will likely be an acceptable error rate for many long-term monitoring applications. The results verify that the dynamic strategy in $Cscan$ can effectively meet the data quality requirements of an application. The energy consumption for different scenarios in Figure 9 are also provided to demonstrate the effectiveness of $Cscan$ as compared to other approaches.. It can be seen that $Cscan$ achieves a better error/energy margin when the error tolerance is between 20 and 50 percent. Intuitively, the higher the error tolerance, the more the energy consumption can be reduced. We also investigated the effectiveness of our prediction model through the measurement of the miss ratio, as shown in Figure 10. The prediction miss ratio for $Cscan$ increases as error tolerance increases. This is not surprising because a high error tolerance implies that the sensor won't be able to anticipate an abrupt change in the environment. However, the highest miss ratio measured is only about 25 percent, which suggests that our prediction model provides a reasonably high prediction accuracy.

2) Impact of training period length: In this experiment, we evaluated the influence of different lengths of the training period on the error rate and energy conservation. When a sensor node begins sensing, an initialization period is required to build both the correlation table and the prediction model. The accuracy of the prediction model will depend on the sample size of the data fed to the model constructor. As we can see in Figure 11, the error rates decrease as the length of training period becomes longer. This becomes more evident when the error tolerance e_t is larger. This can be explained by the fact that the scheduling algorithm has more flexibility to adjust the duty cycle as the error tolerance becomes larger. Notice that the energy consumption also increases with increases in the length of the training period. According to our experiments, the energy consumption increased from 27 percent to 38 percent as the length of training period

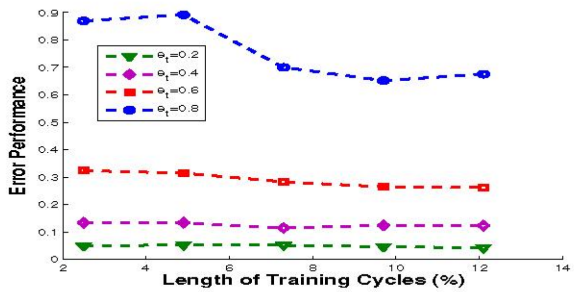

Fig. 11. The influence of training period length on the prediction error rate.

was increased from 2.3 percent to 12 percent of the total number of simulation cycles. As a result, the system exhibits a trade-off between data accuracy and energy conservation.

VI. Related Work

In recent years there has been increasing interest in studying approaches for energy-efficient operation of wireless sensor platforms. These studies include data aggregation techniques to reduce the communication overhead [5], [7], [8]. To more aggressively keep sensor nodes in a dormant state, data prediction has also been investigated. Both numerical approaches and empirical models have been implemented [9], [10]. Using a Dual Kalman Filter, Jain et al. [11] proposed a prediction model to minimize resource usage under a precision requirement. However, the prediction model that was used requires sophisticated computation that results in hardware complexity and increased power consumption at the cluster head. In [10], empirical analysis results revealed the relationship between the configuration parameters and the quality of the search. In references [12], [13], data correlations with spatial coherence and routing efficiency were investigated. Research on dynamic sensing schedulings to balance accuracy and energy saving were also conducted [14], [15], [16]. In eSense [17], a stochastic sensing algorithm used probability bounds for the miss ratio constraint. However, their approach is not sensitive to the degree of data error. In contrast, our approach employs an empirical prediction model to predict sensing data that does not require complicated hardware. Furthermore, we also use data cycle correlations in error estimation to determine the sensing probability, which allows us to achieve significantly higher energy conservation for a given error tolerance.

VII. Conclusions

In this paper, we have presented a stochastic sensing algorithm to reduce energy consumption. Our approach does not require powerful computational ability at the sensor nodes to construct an accurate data prediction model. Observed correlations between different data cycles has been used to estimate the prediction error, thus allowing the scheduler to adjust its operation accordingly. The measurement and simulation results show that system prediction error remains within the specified error tolerance while saving up to 70 percent of the required energy. For our future work, we would like to evaluate the energy performance of individual sensor network components so that the algorithm can be further optimized.

References

[1] H. X.-M. J. M. L. Leblanc, "Wireless sensor network for streetlight monitoring and control," in *SIGMOD*, April 2004.

[2] T. Imielinski and S. Goel, "DataSpace - Querying and Monitoring Deeply Networked Collections in Physical Space," in *Proc. of the Intl. Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'99)*, August 1999.

[3] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat Monitoring: Application Driver for Wireless Communications Technology," in *Proc. of the 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, April 2001.

[4] V. Kottapalli, A. Kiremidjian, J. Lynch, E. Carryer, T. Kenny, K. Law, and Y. Lei, "Two-tiered Wireless Sensor Network Architecture for Structural Health Monitoring," in *Proc. of the Intl. Symp. on Smart Structures and Materials*, March 2003.

[5] I. Solis and K. Obraczka, "in-network aggregation trade-offs for data collection in wireless sensor networks," *International Journal of Sensor Networks*, 2006.

[6] S. Madden, M. Franklin, J. Hellerstein, and W. Hong, "TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks," in *Operating Systems Design and Implementation*, December 2002.

[7] M. Lee and V. Wong, "An energy-aware spanning tree algorithm for data aggregation in wireless sensor networks," Victoria, BC, Canada, 2005.

[8] A. Sinha, A.; Chandrakasan, "Toward optimal data aggregation in random wireless sensor networks," in *INFOCOM 2007*.

[9] D. Kramarev, S. Bae, and K. Kim, "Empirical data fusion for decentralized detection in sensors systems with sequential structure," 2006.

[10] Q. Qiu, Q. Wu, D. Burns, and D. Holzhauer, "Lifetime aware resource management for sensor network using distributed genetic algorithm," Tegernsee, Germany, 2006//, pp. 191 – 6.

[11] A. Jain, E. Chang, and Y. Wang, "Adaptive stream resource management using kalman filters," 2004.

[12] S. Pattem, B. Krishnmachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *IPSN*, 2004.

[13] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "On network correlated data gathering," 2004.

[14] C. L. Y.-X. H. N. Xiong, "An energy-efficient dynamic power management in wireless sensor networks," in *Parallel and Distributed Computing, 2006. ISPDC06*.

[15] P. R. C. C. L. A. M. R.A.F., "Dynamic power management in wireless sensor networks: An application-driven approach," in *Wireless On-demand Network Systems and Services, 2005. WONS 2005*.

[16] H. C. S. K. Wang, "An adaptive hybrid dynamic power management method," in *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on*.

[17] L. H. C. A. S. J., "esense: energy efficient stochastic sensing framework for wireless sensor platforms," in *IPSN 2006*.