

PSS: PREDICTIVE ENERGY-EFFICIENT SENSING SCHEDULING IN WIRELESS SENSOR NETWORKS

Haiyang Liu, Abhishek Chandra and Jaideep Srivastava*
Dept. of Computer Science and Engineering, University of Minnesota
Minneapolis, MN, 55455

ABSTRACT

Wireless sensor networks are being widely deployed for providing physical measurements to diverse applications that have wide variety of data quality requirements. Energy is a precious resource in such networks as sensor nodes are typically powered by batteries with limited power and high replacement cost. This paper presents PSS: an energy-efficient stochastic sensing framework for wireless sensor platforms. PSS is a node-level framework that utilizes knowledge of the underlying data streams as well as application data quality requirements to conserve energy on a sensor node. PSS employs a stochastic scheduling algorithm to dynamically control the operating modes of the sensor node components. This scheduling algorithm enables an adaptive sampling strategy that aggressively conserves power by adjusting sensing activity to the application requirements. Using experimental results obtained on Power-TOSSIM with a real-world data trace, we demonstrate that our approach reduces energy consumption by 29-36% while providing strong statistical guarantees on data quality.

1. INTRODUCTION

1.1 Sensor Energy Management

Unattended Ground Sensors (UGS) are being widely deployed for providing situational awareness that is vital to Army's Future Combat Systems (FCS). These small sensors are organized into mesh networks providing continuous monitoring functions over large areas. Energy efficiency has been widely recognized as one key issue and presents major challenges (Estrin,2002). Many sensor platforms now allow their main components to have multiple operating modes with significantly different power levels (Shayder,2004; Polastre,2005). Even low-end sensors such as temperature/humidity sensors on the Telos

platform (Telos,2004; SHT,2004) now allow automatic mode switching. Most existing research efforts in sensor energy management have focused on optimizing the power consumption of the radio and the CPU (Estrin, 2002; Boulis, 2003). These efforts have been driven largely by the conventional wisdom that these components consume most of the power on a sensor node (Estrin, 2002). In reality, the operation of sensors can be critical in determining the lifetime of a sensor node for the following reasons. First, specialized sensors can be energy consuming. For example, the heading sensor offered by xBow (xBow,2004) can consume a power of about 375 mW, which is much higher than the 60 mW consumed by the mica2 radio transmitting at full power. Second, after common CPU and radio energy management, even low power sensors, if not well managed, could account for a significant fraction of the total energy consumption. Our experiments, presented in Section 4, reveal that the SHT series temperature sensor integrated on the Telos platform, that uses only 1.65 mW of power while sampling, could consume up to 38% of the total energy at a modest sampling rate of 0.1 Hz. After excluding the inherent idle energy consumption, which can be improved only through better hardware design, the percentage of sensing is even higher (about 45-90%). Thus, effective modulation of the sensor operating modes is crucial for better energy conservation. Moreover, reduced sensing activity enables the CPU and the radio to spend more time in sleep mode, thus resulting in even higher energy savings for these components. Therefore, we believe that sensor power control is not only desirable but essential for sensor platform energy management.

1.2 Dynamic Data Quality Requirements

Sensor platforms support versatile applications, which have widely varying data quality requirements from the sensor data streams (Tatbul,2003). For instance, a Heating, Ventilation and Air-conditioning

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 01 NOV 2006		2. REPORT TYPE N/A		3. DATES COVERED -	
4. TITLE AND SUBTITLE PSS: Predictive Energy-Efficient Sensing Scheduling In Wireless Sensor Networks				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Dept. of Computer Science and Engineering, University of Minnesota Minneapolis, MN, 55455				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited					
13. SUPPLEMENTARY NOTES See also ADM002075., The original document contains color images.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 8	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

(HVAC) application might require fine-grained temperature readings of a building. On the other hand, a fire monitoring application may only need to know whether the temperature is greater than a pre-defined threshold, and could afford to have coarse-grained accuracy in its temperature readings. In addition, data quality requirements may change even for the same application over different time periods and for different value ranges (Tatbul,2003). For example, the HVAC application may require more precise readings during daytime when offices are occupied, while only coarse measurements might be sufficient at night when offices are empty. System support for dynamic data quality on sensor nodes also provides applications with an effective means to achieve graceful performance degradation (Deshpande,2005) when the network is congested or the sensor nodes are constrained. In case of such constraints, the application can throttle the data sensing and transmission rates by reducing its data fidelity requirement. As a result, sensor platforms must be able to satisfy dynamic data quality requirements.

1.3 Adaptive Data Sampling

Due to the dynamic data quality requirements, the determination of proper data sampling rate on a sensor platform must be driven by application semantics and the dynamics of the measured data. Existing sensor network applications such as TinyDB (Madden,2002) do not account for these requirements, and the conventional sampling rates used in such applications are static user-supplied parameters. Statically defined sampling rates result in either energy wastage under stable conditions, or unsatisfactory sample quality when the physical phenomenon experiences rapid changes. It is thus desirable to provide adaptive sampling as a system service to end applications, which only need to supply semantic data requirements. This distinction between the semantic and the actual sampling rates would benefit both low as well as high data rate applications by achieving a better tradeoff between energy and data quality. Since sensor data streams are measurements of physical phenomena, correlations within data streams are inherent. For instance, the temperature variation in a room is governed by heat transfer laws, which limit the amount of variation that can occur between two successive temperature readings. Such temporal correlations can be exploited for energy management by taking sensor measurements only when large variations are expected in the underlying data values. In this paper, we present

PSS: an energy-efficient sensing framework that utilizes knowledge of the underlying data streams to conserve energy on a sensor node, while satisfying the application data quality requirements. Coupled with data stream prediction models and data quality models, our scheduling framework dynamically controls the operating modes of the sensor node components. The core of our approach is a stochastic scheduling algorithm that performs data sampling in a probabilistic manner. Using experimental results obtained on an enhanced version of PowerTOSSIM (Shnayder,2004) we demonstrate that our approach reduces energy consumption by 29-36% while providing strong statistical guarantees on data quality.

2. SYSTEM ARCHITECTURE AND COMPONENT MODELS

Figure 1 shows the architecture of the scheduling framework, which implements the PSS scheme on a wireless sensor platform. A data stream model is first constructed from historic readings. State change probabilities can then be computed based on predicted future readings and data quality requirements. A stochastic scheduling algorithm is then applied to compute the sampling probability that minimizes the sampling activities while guaranteeing the quality requirements. Feedbacks from the real time readings are used to dynamically adjust parameters used in the scheduler.

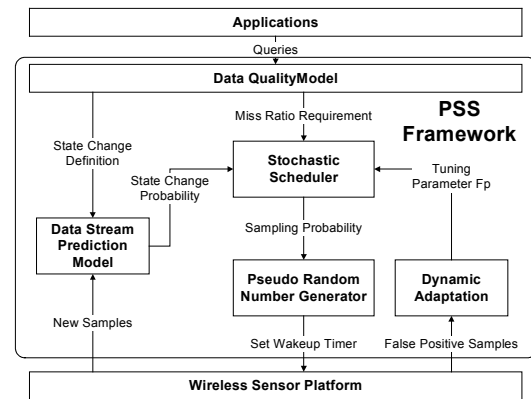


Figure 1: Architecture of PSS Scheme

2.1 Data Quality Model

The quality of measurement data can be generally quantified in terms of temporal resolution, measurement resolution, and sampling quality. Temporal resolution refers to the maximum available

sampling frequency, which determines the granularity of temporal changes that can be captured in the data stream. We define this maximum sampling frequency as the base sampling frequency. The base sampling frequency could depend on the physical limitations of the sensing device, the available communication bandwidth, or the highest temporal resolution required by the application. Thus the sampled data sequence at the base sampling frequency represents the closest approximation to the underlying process that could be achieved by a sensor node in an application. We refer to this data sequence obtained by sensing at the base sampling frequency as the baseline data sequence. The concept of measurement resolution refers to a data range around the measured value that contains the actual data value. For example, a measurement resolution of 2°C for a measurement value of 100 °C means that the true value is bounded in the range (98 °C, 102 °C). We refer to this measurement resolution as the relative resolution threshold σ , capturing the relative error range. In addition, we use the term absolute resolution threshold to denote the absolute difference between the measured value and an absolute threshold. Absolute resolution threshold is commonly used in predicate-based filtering operations in sensor network queries. We then define state change to be a change in the data value exceeding the resolution threshold. Intuitively, a state change corresponds to an interesting sensor measurement that needs to be reported to the application.

In the proposed PSS framework, unnecessary sensing operations are avoided through adaptive sampling, i.e., data is sensed only when a state change is expected. With this approach, it is possible to miss certain state changes if data was not sampled at those time instances. We refer to such missed state changes as false negatives or misses, as they correspond to a false expectation of not having a state change when actually there is one. Non-zero misses are commonly acceptable to many monitoring and aggregation estimation applications in sensor networks. Similarly, it is possible for this sensing approach to make a measurement when there is no actual state change. We refer to such redundant sensing events as false positives or false hits, as they correspond to a false expectation of observing a state change when there is none.

Note that a sensing scheme should strive to minimize both false negatives as well as false positives: while false negatives result in degraded data quality, false positives result in energy wastage. We define two

quantities — the miss ratio μ and the false hit ratio ρ — to quantify the degradation in data quality and wasteful sampling respectively:

$$\mu = \frac{n_f}{n}, \rho = \frac{n_p}{n}$$

where, n_f and n_p denote the number of misses and false hits respectively, and n denotes the total number of sampling points (corresponding to the base sampling frequency).

2.2 Data Stream Prediction Model

PSS employs a data stream model to predict future sensor readings from historical data. Statistical models are particularly suitable for sensor network applications (Deshpande,2004). While several sophisticated statistical models (Vilalta,2002; Deligiannakis,2004) can be used, we used a biased random walk model in our experiments. This model is a type of first-order Markov model that we chose for its computational efficiency and compact representation. This simple model can readily capture the intrinsic correlations in data streams and is sufficient to evaluate the effectiveness of our scheduling algorithm. We now summarize this model here. More details can be found in a technical report (Liu,2005). In this model, a k -step prediction is given by:

$$X_{i+k} = X_i + N(\mu_k, \sigma_k)$$

where, X_i denotes the data value at time instance i , X_{i+k} denotes the predicted data value at time $i + k$, i.e., k time steps forward from step i , and $N(\mu, \sigma)$ denotes a normal distribution with mean μ and standard deviation σ . The possibly non-zero mean value μ or the bias, captures the systematic trend in the data stream, while σ captures the process random noise and non-linear error components. Given the data quality model with a resolution threshold δ , computing the state change probability is as simple as looking up the value of probability in a locally stored unit normal distribution table after appropriate transformations. The model can be constructed from a training data stream and updated with new data samples. The initial construction and subsequent updates could be carried out at base stations, similar to (Deshpande,2004), taking advantage of the storage and computing power of base stations in addition to more complete view of measurement data streams. Since the prediction models take small number of parameters and are updated infrequently, the amortized communication cost of model updates on a sensor node is expected to be negligible.

3. STOCHASTIC SCHEDULING ALGORITHM

3.1 Overview

In this section, we present a stochastic scheduling algorithm that employs the underlying data stream model and the data quality requirement to determine sampling instants for the sensor. The goal of this algorithm is to minimize the sensor energy consumption while meeting the desired data quality requirements. The intuition behind our algorithm is to sample with high probability at instants when state change probabilities are expected to be high. The sensor scheduling algorithm must satisfy several important requirements while determining the sensing points: • The overall energy consumption of a sensing process is the sum of energy spent in the sensors (for making measurements), the radio (for sending required value updates when necessary), and keeping the CPU in power-on state (for sensing, radio transmission and scheduling operations). The scheduling algorithm must try to minimize the sensing energy consumption of all these components. • Since the scheduler tries to save energy by not sampling at some time instants, it is possible to miss some of the state change events. In this case, the scheduling algorithm must ensure that the overall sampled data meets an application-specified data quality requirement such as the miss ratio μ , defined in Section II-A. As sensing decisions must be made for future instants Based on predicted information, uncertainty is inherent in the decisions. For such probabilistic events, deterministic scheduling would result in poor data quality or energy wastage. Therefore, scheduling decisions must be stochastic to account for the uncertainty. Overall, the scheduling algorithm must minimize the energy consumption while meeting the desired data quality requirements in a stochastic manner. Note that our stochastic scheduling algorithm does not depend on the specific prediction model and data quality model being used, and can be used in conjunction with any kind of models as long as they can estimate state change probabilities at future time instants. Next, we formalize the scheduling problem and present our solution.

3.2 Problem formulation

We formulate the stochastic scheduling problem as an optimization problem that minimizes the total

energy consumption while providing statistical guarantees on data sampling quality. Let us assume that the baseline data sequence consists of N data samples, and the probability of state change at a sampling instant i (determined using the underlying data stream model and the application's resolution threshold δ) is q_i . Further assume that the average energy spent for each measurement is e_{avg} (this includes the average energy spent by the sensor, CPU, and the radio). Finally, let the application's data quality requirement be expressed as a tolerance level $F_N \in [0,1]$, such that its miss ratio $\mu \leq F_N$. Then, the goal of the stochastic scheduling algorithm is to determine a probability of sensing $p_i \in [0,1]$ for each sampling instant such that it minimizes the total energy

$$E = \sum_{i=1}^N p_i \cdot e_{avg} \quad (1)$$

under the constraint

$$\frac{\sum_{i=1}^N (1 - p_i) \cdot q_i}{N} \leq F_n \quad (2)$$

The constraint given by Inequality 2 satisfies the statistical data quality requirement of the application as we require the expected miss ratio to be less than the application-specified tolerance level. Recall from Section II-A that the expected miss ratio, μ , is evaluated as the expected number of false negatives divided by the total number of data samples. Thus, given the false negative probabilities f_{ni} over all sampling instances, we have,

$$\mu = \frac{\sum_{i=1}^N f_{ni}}{N} \quad (3)$$

To catch state changes more effectively, the higher the probability of state change q_i , the higher the probability of sensing p_i should be. Therefore, we assume that p_i is, by design, positively correlated to the probability of state change q_i at each sampling instant. Hence, the false negative probability f_{ni} at each scheduling instant is less than what would be obtained by assuming independence between p_i and q_i . In other words, $f_{ni} \leq (1 - p_i) \cdot q_i$. Thus, the miss ratio (Equation 3) reduces to

$$\mu = \frac{\sum_{i=1}^N f_{ni}}{N} \leq \frac{\sum_{i=1}^N (1 - p_i) \cdot q_i}{N}$$

Thus, the constraint (Inequality 2) satisfies the data quality requirement $\mu \leq F_n$. In fact, the constraint is a

conservative bound on the data quality requirement, such that if a schedule satisfies the constraint, it must also satisfy the data quality requirement.

3.3 Scheduling Algorithm

Having presented the problem formulation, we now present a stochastic scheduling algorithm that closely approximates the optimization problem. The goal of the scheduling algorithm is to determine the sensing probability p_i for each sampling instance given the state change probability q_i for that scheduling point. Given q_i , solving for the precise value of p_i would require the joint distribution of the random processes of sampling and state changing. This distribution is neither available nor desirable due to its high storage and computational overhead. Instead, we simplify the computation of p_i as follows: we first determine the upper and lower bounds for p_i , and the scheduling algorithm then chooses a value from this range based on a heuristic we describe later. Intuitively, the upper bound of p_i specifies a limit such that selecting values higher than it would only waste energy for providing unnecessary data quality improvement. On the other hand, the lower bound of p_i corresponds to a limit, such that going below it would always result in violation of the application's tolerance level.

1) Determining the Upper Bound of Sensing Probability: To determine the upper bound on the value of p_i for a given q_i value, our scheduling algorithm performs local Optimization instead of global optimization. Note that local optimization meets a stricter requirement since satisfying the constraint at each sampling instance automatically satisfies the constraint over all sampling instances. In other words, the optimization problem is reduced to minimizing p_i at each scheduling instant under the constraint

$$(1 - p_i) \cdot q_i \leq F_n,$$

which yields the solution

$$P_i^{ub} = 1 - \frac{F_n}{q_i} \text{ if } F_n < q_i \leq 1; = 0 \text{ otherwise.}$$

In other words, p_i^{ub} is the minimum value of p_i that guarantees the satisfaction of the data quality requirement for each sensing instance. This value of p_i is an upper bound on the value of the sensing probability, because, any sensing probability value higher than p_i^{ub} , while always satisfying the local

optimization constraint, would be more wasteful of energy.

2) Determining the Lower Bound of Sensing Probability: To determine the lower bound on the value of p_i , we consider the most optimistic scenario where every sample catches a real state change, i.e., there are no false positives. In this scenario, the data quality requirement can be satisfied only if $q_i - p_i \leq F_n$, which provides us with the following lower bound:

$$P_i^{lb} = q_i - F_n \text{ if } F_n < q_i \leq 1; = 0 \text{ otherwise.}$$

This value of p_i is the lower bound because any sensing probability value smaller than p_i^{lb} would always result in violating the data quality requirement. Thus, the value p_i^{lb} corresponds to the smallest value of p_i given q_i , such that the data quality constraint could be met.

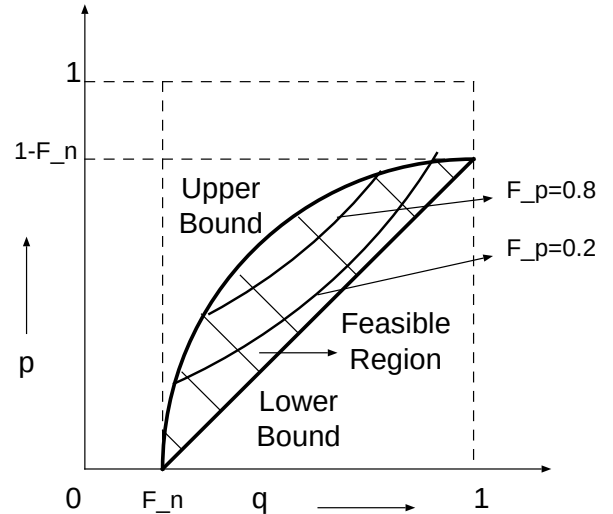


Figure 2: Relation between sensing probability and state change probability

3) Selecting the Sensing Probability Value: Given the upper and lower bounds on the value of p_i given q_i , we present a heuristic to select the actual value of p_i . Note that the application uses a miss ratio bound F_n to limit the data quality degradation. Analogously, we can bound the energy wastage by using a false hit ratio limit F_p . Our heuristic uses this limit F_p as the tuning parameter to determine the p_i value from the region bounded by p_i^{lb} and p_i^{ub} . Lower values of F_p correspond to more aggressive energy saving, while higher values of F_p provide better data quality at the expense of higher power consumption. F_p can be approximated as $F_p = p_i \cdot (1 - q_i)$, which yields

$$p_i = \frac{F_p}{(1 - q_i)}$$

subject to the two bounds derived above. The stochastic scheduling algorithm then uses this p_i value to probabilistically schedule a sensing event at a sampling point. Note that this formula also satisfy the design principle that the higher state change probability is, the higher sampling probability would be. Figure 2 shows the relation between the sensing probability p_i and the state change probability q_i for a given value of the data quality threshold F_n . The figure also shows the intermediate values that p_i would take based on the value of the tuning parameter F_p .

4) Dynamic Adaptation: While major trend change of the measurement data stream can be captured by model updates, local fluctuations and inaccuracy in estimation of q_i may lead to poor scheduling decisions affecting the sample data quality. Thus, it is important to ensure that our scheduling scheme adapts to sudden or unforeseen data variations. While it is not possible to directly observe false negatives (corresponding to missed state changes), we can measure false positive rates to estimate the dynamism in the underlying data. Intuitively, a low rate of false positives implies that most of the sensing events result in state changes, suggesting the possibility of missing other significant changes. Thus, a low false positive rate could be taken as an indication of more dynamic data values, and the number of sampling events should be increased in this case to catch possibly significant state changes. On the other hand, if we observe a high rate of false positives, it means that we are taking large number of redundant samples, many of which are non-informative. Such a high rate indicates a relatively stable data process, and the sampling probability should be decreased in this case to save energy. We use the tuning parameter F_p to achieve this dynamic adaptation of the sampling probability p_i . A Multiplicative Increment Additive Decrement algorithm is utilized in hope of fast responding to sudden events.

5) Practical Considerations: While sampling decision must be made for each time instant, it is inefficient to compute at each instant. Instead, at each scheduled sensing instant (when the CPU is turned on anyway for the sensing operation), the stochastic scheduler determines the next sampling instant using a pre-generated random number sequence and the sequence of sampling probabilities.

4. EXPERIMENTAL EVALUATIONS

4.1 Experimental Setup

The prototype of this framework was implemented on Telos platform running TinyOS. We used real-world temperature readings to test the effectiveness of our prototype.

The temperature data was sampled in an air-conditioned storage room at sampling frequency of 0.1Hz for two days (illustrated in Figure 3). The simulation time period starts at the data point corresponding to about 6 am on the second day in the trace, when air conditioning is configured to turn on in the room. This choice of data set results in richer variations in the test data. A simulation period of 10,000 seconds (corresponding to 1000 sample points) was selected for each run. In order to reduce the artifact of pseudo randomness, each simulation run was repeated multiple times with different random seeds and the arithmetic mean is reported.

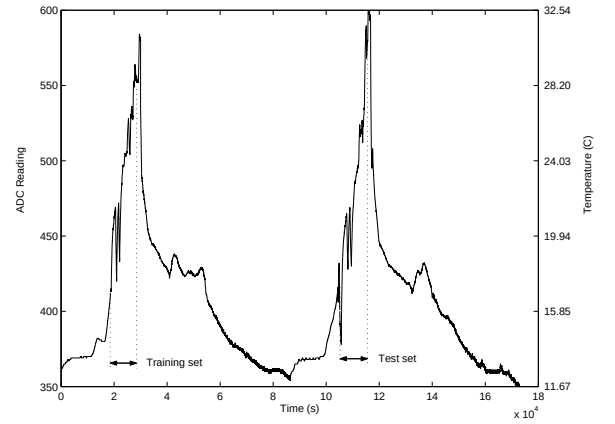


Figure 3: Sampled Temperature Data Trace

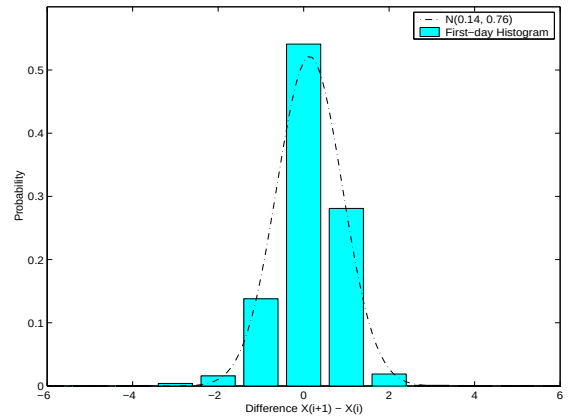


Figure 4: Gaussian approximation to the histogram of a single-step predictor ($k=1$)

The parameters of the data stream model are derived from the histogram over a training period of

10,000 seconds. Figure 4 shows the data histogram and the corresponding Gaussian distribution approximation for single-step prediction ($k=1$). The close approximation illustrated in this figure validates our selection of the biased random walk model. Details about this model can be found in (Liu,2005).

4.2 Experimental Results

In our experiments, multiple sampling strategies were compared. Base sampling corresponds to the original user-specified schedule requirement. Ideal sampling denotes the most energy-efficient schedule assuming Oracle knowledge. Dynamic sampling corresponds to the actual schedule. The upper bound and lower bound samplings are variations of the dynamic sampling.

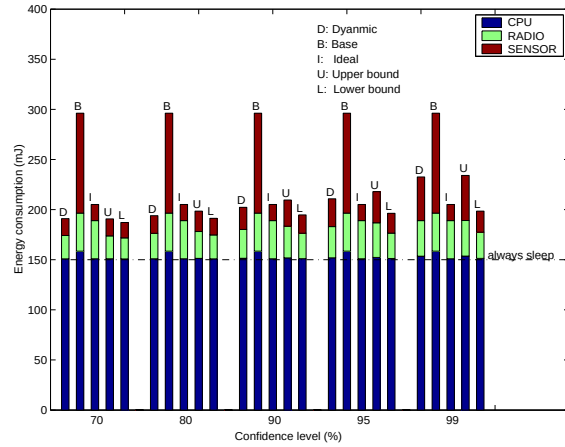


Figure 5: Energy consumption for a fixed resolution threshold = 5

Figure 5 shows up to 30% overall energy savings through energy-efficient sampling. Even for high data quality requirement of a confidence level of 99%, an 18% saving can be achieved. Excluding the idle energy posed by the hardware limit, denoted by the dashed line with annotation of “always sleep” in the graph, more than 60% energy saving can be obtained. In addition the dynamic sampling achieves an energy saving close to the ideal sampling. Note that it is possible the dynamic sampling costs less energy than ideal sampling as the former is allowed false negative misses while the latter catches all state changes.

Figure 6 shows the performance of the stream select operation over the temperature stream with a predicate of ($T > 450$, equivalent to about 20°C). About 35% energy saving were obtained.

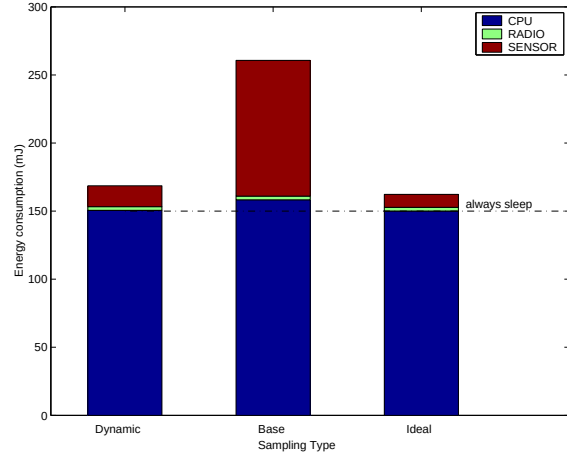


Figure 6: Energy Saving over for Stream Select (absolute resolution threshold = 450 and confidence level = 99%)

5. RELATED WORK

Recently, several research efforts have focused on energy-efficient operations on wireless sensor platforms. These efforts include minimizing data transmission through data compressions (Deligiannakis,2004) and saving energy by turning redundant nodes off while maintaining required field coverage (Abrams,2004). (Boulis,2003) introduces a node-level energy allocation scheme to maximize the overall gain to multiple applications running on a node. (Deshpande,2004) uses a statistical model at the base station to optimize query plans by picking the optimum set of attributes and nodes for data acquisition. However, it considers only communication cost in its cost model and does not provide real time scheduling service in response to sudden events. PSS is complementary to these approaches in that our scheduling framework is a node-level real time approach that primarily targets sensing energy conservation. PSS thus can be used in conjunction with existing work to further reduce energy consumption. Dynamic Power Management (DPM) (Sinha,2001) and multiple sensing unit scheduling (MSUS) (Cam,2005) also attempt to control the operating modes of sensor node components in response to different workloads. Prediction-based dynamic power control has also been used in energy management in mobile systems (Liu,2004; Lorch,2003) and embedded systems (Li,2002; Srivastava,1996). However, unlike our approach, they are completely unaware of application semantics. Stochastic sensor scheduling has also been used in target-tracking applications [23] for

maximizing estimation accuracy. It is fundamentally different from our goal of minimizing energy consumption given an accuracy requirement.

6. CONCLUDING REMARKS

In this paper, we presented PSS: an energy-efficient sensing framework for wireless sensor platforms, that can achieve significant energy savings in response to dynamic data quality requirements. Our scheduling framework dynamically controls the operating modes of the sensor node components using a stochastic scheduling algorithm coupled with data stream prediction models and data quality models. Using experimental results obtained on PowerTOSSIM with a real world data trace, we showed that our approach reduces energy consumption by 29-36% while providing strong statistical guarantees on data quality.

As part of future work, we intend to extend our work to schedule data transmissions on the network by modeling the radio component as a pseudo-sensor and to multiple sensors by exploring their correlations. The traffic irregularity caused by this type of sensing scheduling will also be studied to improve the communication protocol stack.

REFERENCES

- Abrams, Z. etc. 2004: Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks,” in *Proceedings of IPSN’04*
- Boulis A. etc. 2003: Node-level Energy Management for Sensor Networks in the Presence of Multiple Applications, in *Proceedings of PerCom’03*
- Boulis, A. etc. 2003: Aggregation in sensor networks: a energy-accuracy trade-off, in *Proceedings of SNPA’03*
- Cam, H. etc. 2005: Energy-efficient Task Scheduling for Wireless Sensor Nodes with Multiple Sensing Units,” in *Proceedings of IPCCC’05*
- Deligiannakis, A. etc. 2004: Compressing Historical Information in Sensor Networks, in *Proceedings of SIGMOD’04*
- Deshpande, A. etc., 2004: Model Driven Data Acquisition in Sensor Networks, in *Proceedings of VLDB’04*.
- Deshpande, A. etc. 2005: Resource-aware Wireless Sensor-Actuator Networks,” *Data Engineering Bulletin*, vol. 28, no. 1, pp. 40–47
- Estrin, D. etc., 2002: Wireless Sensor Networks, tutorial at *MobiCom’02*.
- Gupta, V etc. 2004: Sensor Scheduling Algorithms Requiring Limited Computation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*
- Li, D. etc. 2002: Mode Selection and Mode-Dependency Modeling for Power-Aware Embedded Systems, in *Proceedings of the 7th Asia and South Pacific Design Automation Conference*, pp. 697–704.
- Liu, X. etc. 2004: A Time Series-based Approach for Power Management in Mobile Processors and Disks, in *Proceedings of NOSSDAV’04*, pp. 74–81.
- Liu, H. etc. 2005: dSENSE: Data-driven Stochastic Energy Management for Wireless Sensor Platforms, *Dept. of CSE, Univ. of Minnesota, Tech. Rep. TR 05-018*
- Lorch J. etc. 2003: Operating System Modifications for task-based speed and voltage scheduling, in *Proceedings of MobiSys’03*, pp. 215–229
- Madden, S. etc. 2002: TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks, in *Proceedings of OSDI’02*
- Polastre, J. etc. 2005: Telos: Enabling Ultra-Low Power Wireless Research,” in *Proceedings of IPSN’05*.
- Shnayder, V. etc. 2004: Simulating the Power Consumption of Large-Scale Sensor Network Applications, in *Proceedings of SenSys’04*
- SHT1x/SHT7x Humidity and Temperature Sensor Datasheet, 2004: http://www.sensirion.com/en/pdf/Datasheet_SHT1x_SHT7x.pdf.
- Sinha A. etc. 2001: Dynamic Power management in Wireless Sensor Networks,” *IEEE Design and Test*, vol. 18, no. 2, pp. 62–74
- Srivastava, M. etc. 1996: Predictive System shutdown and other architectural techniques for energy efficient programmable computation, *IEEE Transactions on VLSI Systems (TVLSI)*, vol. 4, no. 1, pp. 42–55
- Tatbul, N. etc. 2003: Load Shedding in a Data Stream Manager, in *Proceedings of VLDB’03*
- Telos/Tmote Datasheet, 2004: <http://www.moteiv.com/products/docs/tmote-sky-datasheet.pdf>.
- Vilalta, R. etc. 2002: Predictive algorithms in the management of computer systems,” *IBM Systems Journal*, vol. 41, no. 3, pp. 461–474, 2002.
- xBow Technology Inc. 2004: Heading Sensor Datasheet, http://www.xbow.com/Products/Product_pdf_files/Mag_pdf/6020-0030-01_AC%HS110.pdf.