QuARES: Quality-aware Data Collection in Energy Harvesting Sensor Networks

Nga Dang Elaheh Bozorgzadeh Nalini Venkatasubramanian

Computer Science Department
University of California, Irvine
Irvine, USA
email: ntdang, eli, nalini @ics.uci.edu

***Abstract*—Renewable energy technology has become a promising solution to reduce energy concerns due to limited battery in wireless sensor networks. While this enables us to prolong the lifetime of a sensor network (perpetually), unstable environmental energy sources bring challenges in the design of sustainable sensor networks. In this paper, we propose an adaptive energy harvesting management framework, QuARES, which exploits an application's tolerance to quality degradation to adjust application quality based on energy harvesting conditions. The proposed framework consists of two stages: an offline stage which uses prediction of harvested energy to allocate energy budget for time slots; and an online stage to tackle the fluctuation in time-varying energy harvesting profile. We implemented the application and our framework in a network simulator, QualNet. In comparison with other approaches (e.g., [9]), our system offers improved sustainability (low energy consumption, no node deaths) during operation with data quality improvement ranging from 30-70%. QuARES is currently being deployed in a campus-wide pervasive space at UCI called Responsphere[11].**

Keywords: energy harvesting, wireless sensor network

I. INTRODUCTION

In energy harvesting systems, energy is derived from environmental sources such as sunlight, wind and heat. Renewable energy sources allow us to continuously harvest energy from the environment, providing a constant and perpetual source of energy to the systems they drive. Despite its low efficiency, renewable energy technology is a viable and promising solution for low power wireless sensor network systems. Scavenging energy could enable smart sensors to be functional indefinitely and as a result, eliminating the cost for battery, enabling sustainable and manageable infrastructures.

Nevertheless, renewable energy depends heavily on environmental conditions (e.g., harvested solar energy on a sunny day is much higher than it is on a cloudy or rainy day). The time-varying characteristics of renewable energy sources creates a shift in research focus from energy-efficient to energy-neutral approaches, i.e. from optimizing energy consumption to *adapting systems to deal with unstable energy sources while meeting application quality constraints.* Renewable energy sources such as solar energy show predictable patterns that are exploited in several *coarse-grain (or slot-based) harvested energy prediction* methodologies [4, 9, 15]. Harvested energy prediction is utilized for planning activities of nodes in a sensor network to keep them functional, yet still powered.

Wireless sensor networks (WSN) are deployed in infrastructures, such as buildings or bridges and enable various data collection applications (e.g. structural monitoring). Sensors collect information about their surrounding environment; update a base station and respond to frequent or sporadic monitoring requests [14]. The base station stores such information in a cache and the bounded difference between the cached values and the actual instantaneous data values at the sensors is called the error margin. The energy cost of data collection applications relates heavily to the frequency of data requests and updates between sensors and the base station, which in turn affects accuracy of the collected data or the error margin. In systems with energy harvesting capabilities, we envision that sensors only communicate when there is sufficient harvested energy. There is therefore, a tight coupling between the ability of the system to harvest energy and the consequent data accuracy – intuitively better harvesting leads to better data quality; poor harvesting conditions imply loss of accuracy.

We propose an energy harvesting management framework called QuARES for data collection applications in wireless sensor networks. To the best of our knowledge, this work is the first attempt to jointly use both application data quality (expressed as error margins) and harvesting ability to manage the energy budget of such systems. Our framework includes two stages: an offline slot-based energy budget allocation algorithm and an online adaptation strategy. The offline stage exploits the slot-based harvested energy prediction and the relation between energy cost and data accuracy to allocate energy budget for each time slot in a given harvesting period (e.g., one day, one week). In the online stage, an online adaptation policy is proposed to guarantee timely responses to queries in spite of the time-varying characteristics of harvested energy.

Our contribution includes: (1) exploiting application tolerance to quality degradation to adapt the sensor data collection process under unstable energy harvesting conditions, (2) design of the QuARES framework, an energy harvesting management framework with 2 stages (online and offline) that utilizes energy harvesting prediction and knowledge of application tolerance–energy cost to maintain system sustainability and optimize data quality, (3) performance evaluation of the QuARES management framework as compared to other offline/online strategies (fixed-error-margin, minimum variance[9]) under different application and energy harvesting scenarios using the QualNet simulator (here, the battery model was modified to simulate energy harvesting) considering different sensor inputs, application constraints, weather conditions and battery capacities; (4) implementation, deployment and measurement in a real-world campus testbed, Responsphere at UCI. Our simulator is also a valuable tool for designers to tune system parameters, to check feasibility of application constraints under various energy harvesting conditions and to study system performance. Results show that our framework can tolerate lower error margins (i.e. higher data accuracy) of 30-70%, ensure a response to all queries; additionally, sensors do not have to shut down to replenish; Results also support the fact that while offline stage is needed to plan the energy budget to tolerate lower error margins, the online adaption is required for responsiveness to queries.

II. RELATED WORK

Recent research has enabled wireless sensor motes to harvest energy from the surrounding environments [1-3]. Energy-neutral approaches have been proposed to cope with the time-varying characteristics of energy harvesting profile. Most of these approaches are cross-layer, considering energy status at battery layer and adapting system at other layers. Hsu et al. [4] adapt duty cycling of systems to the changes in renewable sources. Hasenfratz et al. [10] modify routing protocol at MAC layer to exploit both temporal and spatial variations of renewable energy and maximize data delivery rate for sensors. Voigt et al. [7] adapt LEACH, a cluster-based routing protocol for sensor networks to take advantage of energy harvesting. At operating system layer, Liu et al. [5] and Moser et al. [6] propose task scheduling techniques for energy harvesting systems. At the application layer, Noh et al. [8] use an adaptive technique to turn on and off storage services based on different energy thresholds. Ravinagarajan et al. [12] adapt task utility of structural health monitoring applications to maximize accuracy of tasks.

Some approaches exploit patterns in renewable energy profile to predict future harvested energy and to plan energy budget accordingly. Wang et al. [9] propose a minimum variance slot-based energy budget allocation for systems which prefer steady level of operation. This solution is not suitable for systems whose level of operation is dictated by application and user constraints that vary. Moser et al. [13] allocate energy budget for time slots to maximize quality of service for general systems. However, given the fluctuation of harvested energy within each time slot, an offline budget allocation methodology cannot always guarantee the desired quality of service.

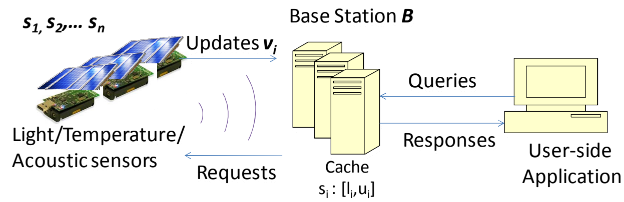


Fig. 1. Data Collection in a Wireless Sensor Network.

None of these works consider data collection applications with varying data accuracy needs, i.e. those that have flexibility of error margins. [21] and [22] attempt to optimize data rate for data collection in wireless sensor network without considering application quality needs as well as application's tolerance to degradation. In [14], Han et al. propose an adaptive data collection protocol which is aware of these data accuracy requirements and exploits error margin of many applications to minimize energy consumption and prolong battery life-time. This approach is designed for and suited to battery powered sensor systems. Our work, on the other hand, exploits error tolerance in both offline and online stages to adapt the system to the fluctuations of renewable energy sources.

III. DATA QUALITY IN ENERGY HARVESTING SYSTEMS

Our system consists of a wireless sensor network (WSN) deployed in an infrastructure for monitoring purposes. The components of our systems are a set of n sensor nodes $\{s_1, s_2, \ldots, s_n\}$ and a base station(s) B as shown in Fig. 1. Each sensor s_i collects information about its surrounding environment by reading value v_i from its embedded sensor and periodically sends an update to the base station(s). v_i is a property of the environment, e.g., temperature, humidity or sound, that the application needs to monitor through our WSN. In this study, we assume that all sensor nodes are equipped with a harvesting circuitry. Harvested energy is accumulated in an energy buffer that supplies power for sensor nodes' operation.

A base station B resides at a node with unlimited resources, e.g., power, storage, computation. It collects data from sensors and stores them in a cache. The cache contains an approximated value u_i for each sensor s_i*'s* true value v_i. Base station B is connected to a monitoring application on the user side. The application periodically polls sensor nodes through the base station(s) for the monitored information. When necessary, the application can ask for sporadic information. In particular, the application sends a query Q_j to the base station each time it needs data from a sensor node or a set of sensor nodes. Query Q_j contains data accuracy constraints specified as error margin. If the approximated value u_i for a sensor s_i satisfies these constraints, base station B returns u_i to the monitoring application. Otherwise, B sends an update request to sensor s_i, to retrieve the latest value v_i, and replies to query Q_j with the updated approximated value.

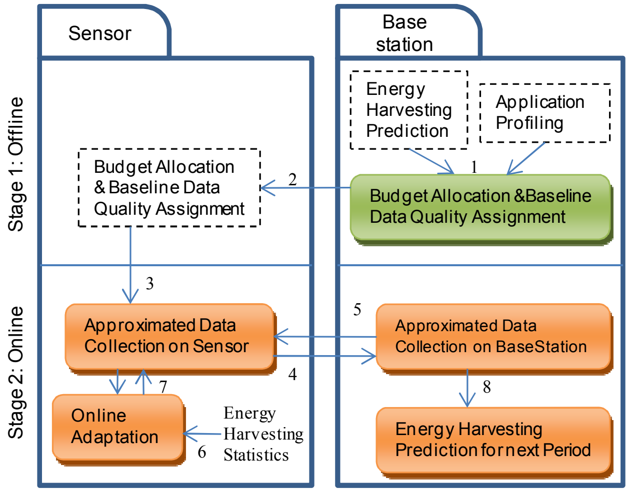


Figure 2. QuARES Framework

Data accuracy can be expressed as an error margin of the actual value v_i, e.g., $v_i \pm 10$ or $v_i \pm 10\%$. Such error tolerance can be exploited while tuning the system. Error margins can be increased or decreased to meet both data accuracy constraints and system constraints, such as varying energy supply. Our energy harvesting management framework exploits this error tolerance to adapt the system to the availability of renewable energy sources. However, there are several challenges in managing such dynamic systems as outlined in the next section.

A. Energy Harvesting Systems

Exploiting renewable energy patterns, such as daily and seasonal patterns, several prediction algorithms (see [4], [9], [15], [16]) have been proposed to predict future harvested energy. Prediction information is composed of the predicted total energy harvested in each time slot in a harvesting period rather than a series of instantaneous values. Typical systems keep track of the average rate of energy harvesting; they do not maintain specific statistics on variations in harvested energy as a factor of time.

Furthermore, in energy harvesting systems, dynamicity is not only from the fluctuation in energy supply but also from the changes in input data and application quality constraints. To make the application demand match the energy supply is a critical task for system sustainability, realizing additional criteria such as the optimization of application quality, specifically data accuracy under these constraints, is even more challenging.

The first challenge is to utilize the prediction information about future harvested energy to sustain the system and maximize overall data accuracy. If high data accuracy is assigned to an interval with predicted low energy, the energy supply will not meet the energy demand and the system might run out of battery and shut down, suspending monitoring activities. If low data accuracy is assigned to an interval with predicted high energy, the harvested energy is not utilized and might be wasted due to energy overflow. The second challenge is to adapt the assigned data accuracy to the actual rate of harvested energy that the system perceives at run-time. If the energy in the buffer is low and the harvested energy rate is lower than the predicted average rate, the system will not have enough energy to sustain itself, let alone maintain the assigned data accuracy.

A high level definition of the problem is as follows: Given predictions of energy harvesting and knowledge of error tolerance and energy cost, the goal is to guarantee maximum system lifetime while optimizing data accuracy.

B. Proposed Framework - QuARES

We propose a framework called **QuARES** – **Qu**ality **A**ware **R**enewable **E**nergy **S**ystem to address this problem. **QuARES** is a cross-layer energy management framework consisting of 2 stages, one offline stage and one online stage. Each stage addresses a challenge in section III.A respectively. Fig. 2 depicts our framework and its stages.

Stage 1 is executed offline in the current harvesting period. In this stage, the base station runs a prediction algorithm to extrapolate information about energy harvested in the next harvesting period. The predicted information and the knowledge of data accuracy-energy cost are inputs to an optimizing algorithm (step 1, Fig. 2). This algorithm allocates energy budget for each time slot in the next period and optimize overall data accuracy. The results are then sent to sensor nodes (step 2) and each sensor node stores the energy budget in its memory.

Stage 2 is executed online in the subsequent harvesting period. At the beginning of each time slot, the corresponding energy budget is retrieved from memory and a corresponding baseline error margin is looked up (step 3). Sensor nodes and the base station exchange messages according to their protocol to maintain the error margin (steps 4 & 5). Online adaptive policies monitor the energy buffer and the actual harvested energy (step 6) to adjust the baseline error margin (step 7). Adaptation allows the system to cope with variation in renewable energy source and maintain system sustainability. In addition, energy harvesting statistics are sent to the base station (together with messages in step 4) to update energy harvesting prediction of the next harvesting period (step 8).

IV. PROBLEM FORMULATION

In this section, we define system parameters, energy harvesting and data accuracy models. Next, we describe the algorithms used in each stage of our framework.

Each sensor has a battery, with capacity C, to store harvested energy. Energy accumulated beyond capacity C will be discarded (energy overflow). Let $E^0_{initial}$ be the available energy at the beginning of the harvesting period. Let $E_{\min}$ be the minimum energy to be reserved at the end of the harvesting period.

Energy Harvesting Model - We denote the length of a harvesting period as T. A harvesting period is then divided into equal-length intervals or time slots. Let N be the number of slots in a harvesting period T. The values of T and N are defined by the harvested energy prediction algorithm on the basis of renewable energy source, the nature of sensor input and query model. Our framework, however, is independent of these parameters. For each time slot i, the prediction algorithm provides $E^i_{harvest}$ which is the amount of harvested energy in that slot. We assume an ideal slot based harvested energy prediction algorithm.

Data Accuracy Model - We model data accuracy in terms of error margin. Error margin δ_i is the bounded difference between the sensed value v_i at the sensor node s_i and the approximated value u_i at the base station B, such that $|u_i - v_i| <= \delta_i$. The entry in the base station's cache is not a single value but an approximation range [l_i, u_i] where l_i is the lower bound and u_i is the upper bound for sensed value v_i and $u_i - l_i = 2\delta_i$. This error margin is the constraint of the application and is not the measurement error or sensitivity of physical sensors. We provide the following definitions:

Definition 1: *Baseline Average Error margin* denoted as $\overline{\delta}_{i,base}$ is the average of baseline error margin predicted in the offline stage, representing the predicted average data accuracy of data collected from sensor s_i.

Definition 2: *Actual Average Error Margin* denoted as $\overline{\delta}_i$ is the average error margin maintained by sensor s_i in a harvesting period T, representing the actual average data accuracy of data collected from sensor s_i.

Our framework, QuARES, optimizes $\overline{\delta}_{i,base}$ in the offline stage and uses online adaptation to maintain $\overline{\delta}_i$ close to $\overline{\delta}_{i,base}$ during online stage.

Definition 3: *Consistent state* refers to the state of cached entry on the base station B. If the approximation range $[l_i,u_i]$ satisfies $u_i - l_i = 2\delta_i$ and sensed value v_i satisfies $l_i \leq v_i \leq u_i$, the cache entry is in a consistent state.

Definition 4*: Inconsistent state* refers to the state of cached entry on the base station B when v_i falls outside the approximation range $[l_i,u_i]$ or when $u_i - l_i \neq 2\delta_i$.

When sensor node s_i reads a new value v_i, it checks if the cache is still consistent. If the cache is inconsistent, a new approximation range [vi-δ_i, vi+δ_i] is sent to the base station to update the cache. This process is called source update. If the sensor node does not update the base station, e.g., running out of battery, the cache entry is in inconsistent state. The number of source updates essentially reflects the physical characteristics of monitored phenomenon. We assume the sampling rate of sensors is sufficient to detect changes in the monitored environment and time to send update message is negligible.

When the base station B receives a new query Q_j, it first checks if the current cache entry satisfies the data accuracy constraint A_j, i.e. $\delta_i \leq A_j$. If it is satisfied, the base station immediately responses to the query with value *(u_i + l_i)/2.* Otherwise, the base station sends a request to s_i for current sensed value v_i. The sensor node replies with the updated approximation range [vi-δ_i, vi+δ_i], the base station updates its cache entry and sends vi to the application. This process is called *consumer update.* The number of consumer updates reflects the nature of query model both in term of query frequency and associated accuracy constraints.

The smaller the error margin, the more source updates and less consumer updates. On the other hand, the larger the error margin, the less source updates and more consumer updates. The two extremes are $\delta = 0$ (only source updates) and $\delta = \infty$ (only consumer updates). However, *consider both types of updates*, Olston et al. [17] find an error margin $\delta_{\max}$ where the total number of updates and energy consumption is minimum. Hence, we only need to consider error margin in the range [0, $\delta_{\max}$] as beyond this, both the error margin and energy consumption is higher.

We assume to have a vector ($\delta_1 .. \delta_K$) and a corresponding vector ($E^1_{\cos t},.. E^K_{\cos t}$) where $E^j_{\cos t}$ is the energy required to maintain error margin δ_j in a time slot. K is the number of error margin levels. This function from error margin to energy consumption is an abstraction of the relation between application quality and the nature of monitored physical phenomenon and query behavior.

A more formal characterization/formulation of the problem is given in Fig. 3 where our goal is to minimize the actual average error margin. In the next two sections, we describe our algorithms in each stage to achieve this.

Problem Formulation
Given input:
Harvesting period T, N time slots
Battery capacity $C \leq 0$ and Initial battery $E^0_{initial} \geq 0$
Minimum battery remained after T: $E_{\min} \geq 0$
Energy harvesting prediction of each time slot $E^i_{harvest}$ $0 \leq i < N$
Error margin and energy cost vectors: ($\delta_1 .. \delta_K$) and ($E^1_{\cos t},.. E^K_{\cos t}$)
Objective: Minimize $\overline{\delta}$, the average actual error margin, in a harvesting period T

Figure 3. Problem Formulation

A. Stage 1: Budget Allocation and Baseline Data Quality Assignment

In stage 1, we solve a linear optimization problem (see Fig. 4) to allocate energy budget and assign a corresponding baseline error margin for each time slot. The linear optimization problem for each sensor node is solved by a linear solver on the base station. System parameters, harvested energy prediction and data accuracy – energy cost in Fig. 3 are input to this optimization problem.

Let $E^i_{initial}$ denote the energy in the battery at the beginning of time slot i. Since the battery cannot store more than its capacity C, $E^i_{initial}$ must be constrained by this upper bound (constraint 1, Fig. 4). Any excess energy is discarded (energy overflow). Let E^i_{budget} denote the energy budget for slot i which could only be drawn from the available energy in the battery at the beginning of the slot and the energy harvested during this slot (constraint 2). Vice versa, the energy in the battery at the beginning of a slot is limited by the amount of energy available in the previous slot subtracted by its energy consumption (constraint 3).

The base line error margin for each slot is assigned based on the allocated energy budget. Let $q[i,j] = 1$ if error margin δ_j is assigned to slot i, $q[i,j] = 0$ otherwise. Each slot i could only be assigned one base line error margin (constraint 4) and this error margin is computed in constraint 5. In addition, its energy budget must be sufficient to maintain this error margin, i.e. at least $E^j_{\cos t}$ (constraint 6). Let $\delta^i_{\max}$ be the maximum tolerated error margin of the application in slot i and also the upper bound for the assigned base line error margin of slot i (constraint 7). Many applications could have time-based quality constraints such as monitoring closely during day-time than night-time or vice versa. Under this constraint, the QuARES framework makes sure the data accuracy is never degraded beyond application needs.

The objective of this optimization problem is to maximize the average base line error margin, $\overline{\delta}_{base}$. Next, we describe stage 2 of our framework, online dynamic adaptation whose task is to maintain the assigned baseline error margin and guarantee continuous system operation.

Stage 1 at Base Station B

1. Run a linear solver to solve this linear programming for each sensor s

Objective Minimize $\overline{\delta}_{base} = \dfrac{\sum_{i=0}^{i=N} \delta^i_{base}}{N}$

Subject to constraints:

Input and Constraints in Figure 3

$$E^i_{initial} <= C \quad (1)$$

$$E^i_{budget} \le E^i_{initial} + E^i_{harvest} \quad (2)$$

$$E^{i+1}_{initial} <= E^i_{initial} + E^i_{harvest} - E^i_{budget} \quad (3)$$

$$0 <= q[i,j] <= 1 \quad \forall 0 <= i < N, 0 <= j < K$$

$$\forall i < N : \sum_{j=0}^{K} q[i,j] = 1 \quad (4)$$

$$\forall i < N : \delta^i_{base} = \sum_{j=0}^{K} q[i,j] * \delta_j \quad (5)$$

$$\sum_{j=0}^{K} q[i,j] * E^j_{\cos t} \le E^i_{budget} \quad (6)$$

$$\forall i < N : \delta^i_{base} <= \delta^i_{\max} \quad (7)$$

2. Send energy budget allocation and baseline data quality assignment to sensor *s*

Stage 1 at Sensor node s

1. Receive energy budget allocation and baseline data quality from base station
2. Save in memory for run-time use in the next harvesting period

Figure 4. Stage 1 – Offline energy budget allocation

B. Stage 2: Online Dynamic Adaptation

During stage 2, data collection protocol runs on both sensor and base station to keep the cache in consistent state and respond to monitoring queries. However, the actual harvested energy rate could be lower than the predicted average rate. The system thus needs online adaptation policies to tackle energy supply fluctuations, maintain system operation and data accuracy constraints.

Our online adaption running on sensor node is a heuristic which keeps track of current harvesting rate and battery status to adjust the error margin, guarantee system operation and consistency of cached entry on base station. We develop 2 dynamic adaptation policies: inter-frame adaptation and intra-frame adaption (see Fig. 5).

1) Inter-frame adaption is triggered at the beginning of each time slot. The harvested energy often does not come at a constant rate; when harvested energy is abundant and the energy buffer is almost full, energy overflow happens. The harvested energy thus could be less than what expected and the system needs to adapt its energy budget plan and base line error margin. The inter-frame policy keeps track of this energy discrepancy and distributes the energy offset among current and future slots' energy budget. Our inter-frame adaptation algorithm is summarized in Policy 1, Fig. 5.

2) Intra-frame adaptation on the other hand is triggered more often every sub-slot within a time slot to quickly adapt to fluctuations of the renewable energy source. The length of a sub-slot depends on the energy source, which is the interval of time that the harvested energy rate remains fairly stable, e.g. 1-5 minutes. Every sub-slot, intra-frame policy will check if the current harvesting rate is significantly less than the predicted average harvesting rate and adjust error margin in the current sub-slot accordingly. In the next sub-slots, if the harvesting rate increases above the expected average rate, the policy restores the baseline error margin. Our inter-frame adaptation algorithm is summarized in Policy 2, Fig. 5.

V. EVALUATION

In this section, we first explain the experimental setup to evaluate the effectiveness of our proposed framework QuARES and then we compare the results with other existing policies in terms of error margin (data accuracy), responsiveness to queries, and energy consumption.

A. Experimental Setup

We initially implemented the approximated data collection application and QuARES framework in QualNet network simulator [18]. The simulator is configured to simulate a sensor network of Mica motes with ZigBee standard specification. Power consumption of sensor node is set accordingly to Mica-2 [19]. We are also developing a prototype test bed with harvesting capability (see sec. V.C).

Policy 1: Inter-frame online adaptation (slot i)

1. $buffer$ = current energy in the buffer
2. $offset = E^i_{initial} - buffer$
2. **if** ($offset > epsilon$) **then** #adjust budget of future slots
3. $E^i_{budget} = E^i_{budget} - offset$
4. j = find_quality_level(E^i_{budget})
5. $\delta = \delta_j$
6. update_server(s_i, $v - \delta$, $v + \delta$)

Policy 2: Intra-frame online adaptation (slot i)

1. h = current_harvesting_rate;
2. if ($|h - h_{old}| < epsilon$) then return
3. else $h_{old} = h$
4. $buffer$ = current energy in the buffer
5. $reserve$ = buffer energy reserved for future sub-slots
6. l = length of a sub_slot
7. $supply = h*l + (buffer-reserve)$ #energy supply for this sub-slot
8. **if** ($supply < E^i_{budget}/number\_of\_subslot$)
9. **then** δ = find_quality_level ($supply*number\ of\ subslot$)
10. **else** $\delta = \delta^i_{base}$
11. **if** δ changes **then**
12. update_server(s_i, $v - \delta$, $v + \delta$)

Figure 5: Stage 2 - Online Adaptation Policies

Sensor data are generated randomly from the range *[-150, 150]*. The sampling rate is *100Hz*, each sampling either increases or decreases the previous value by an amount randomly chosen in the range *[0.5, 1.5]*. Fig. 6b gives an example of randomly generated sensor data for simulated time of 6 hours. Periodic queries arrive every *100 ms*. Sporadic queries are modeled by Poisson distribution with mean interval = *100 ms*. Each query is associated with an error tolerance of mean = *20* and deviation = *1*.

Energy harvesting profile is retrieved from National Renewable Energy Lab website [20]. Solar profiling for a day is shown in Fig. 6a. The data is average solar irradiance (mW/m^2) at a specific location every 5 minutes. The irradiance is converted to harvested energy by linear conversion considering solar panel size *9.6cmx6.4cm*, solar cell efficiency *10%* and harvesting efficiency *80%*. We modify QualNet battery model to charge battery every 1 minute. We assume a perfect solar energy prediction algorithm which gives accurate slot-based prediction to our offline stage. We choose $T = 1$ day and $N = 48$ slots.

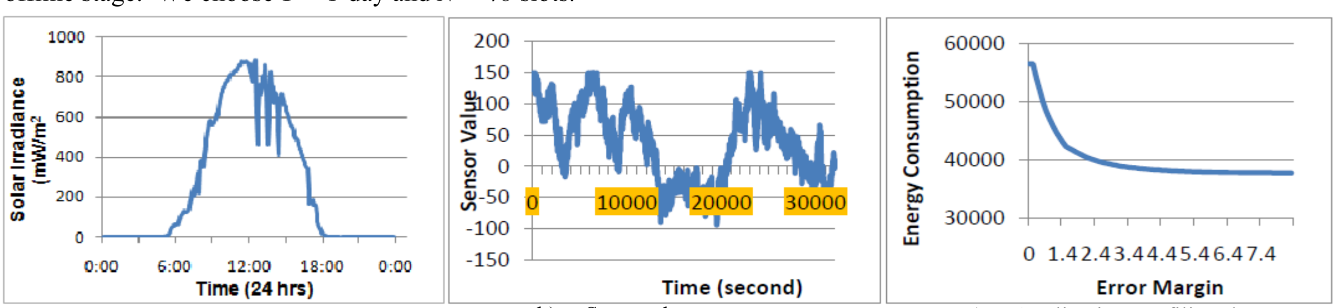
We profile error margin vs. energy cost by simulating application for different error margins in the range $[0, \delta_{\max}]$ (see Fig. 6c). Since it is impossible to profile every value in this range, we choose to profile values at a constant interval step, $r = 0.1$. For each error margin $\delta_{profile}$, we fix the baseline error margin for all slots ($\delta^i_{base} = \delta_{profile}$) and set energy buffer always full. We run the simulation for a simulated time of one day and divide the total energy consumption by the number of time slots N to obtain average energy cost per slot at error margin $\delta_{profile}$. For the given query model and sensor input, we find that $\delta_{\max} \approx 8.0$.

B. Experimental results

To evaluate the effectiveness of our proposed QuARES framework, we have implemented several offline or online policies for energy management during data collection. We evaluate the policies in terms of their data accuracy (error margin), system sustainability, and energy consumption. The implemented policies are as follows:

- FIX_ERROR($\delta = 8.0$) and FIX_ERROR($\delta = 0.5$): FIX_ERROR is an offline policy and has no online adaptation. A fixed baseline error margin is assigned to all time slots.
- GREEDY_ADAPT: Greedy adaptation is a completely online protocol without offline energy budget assignment. It sets an error margin at the beginning of each time slot according to the amount of available energy in the buffer.
- MIN_VAR: Adopted from [9], it allocates energy budget for slots in the offline stage with minimum variance. Its goal is to maintain steady operation for the system. It does not have online adaptation.
- QuARES: As presented in this paper, our quality-aware framework minimizes error margin both in offline and online stages.

Table 1 shows results of QuARES in comparison with other approaches. FIX_ERROR($\delta = 8.0$) has a very high error margin and consumes minimum energy. Due to high error margin, source updates is very low compared to consumer updates; this scheme is thus very close to push-based data collection. The system responds to all queries at the trade-off of higher error margin. This is the extreme case where energy saving is a dominant requirement



a) Solar data b) Sensor data c) Application Profiling data

Figure 6: Inputs to our simulator

TABLE 1. Comparison of 5 different approaches

	FIX_ERROR ($\delta = 8.0$)	FIX_ERROR ($\delta = 0.5$)	GREEDY_ ADAPT	MIN_ VAR	QuARES offline	QuARES offline + online
Average Error Margin	8.00	0.50	0.348	0.388	0.156	0.159
Total Energy Consumption (J)	1813	2686	2656	2641	2641	2641
Shut down time for harvesting (minute)	0	45	21	7	4	0
Failed responses to queries	0	570	420	196	64	0

TABLE 2. Varying application constraints

	App. Constr. 1		App. Constr. 2	
	MIN_ VAR	QuARES	MIN_ VAR	QuARES
Average Error Margin	3.09	0.891	1.067	0.783
Energy Cons. (J)	2481	2509	2481	2501
Shut down time (min.)	32	0	9	0
Failed responses to queries	521	0	227	0

compared to data accuracy and is suitable for traditional battery powered systems [16]. The second column FIX_ERROR ($\delta = 0.5$) is another fixed error rate policy which attempts to maintain a lower error margin $\delta = 0.5$ by exploiting energy harvesting. Due to very low error margin, source updates is very high compared to consumer updates; this scheme thus is very close to pull-based data collection. However, it fails since the sensor node cannot maintain this low error margin when the harvested energy is low. The battery is exhausted and system needs to shut down for 45 minutes to replenish energy. This leads to a very high number of failed responses to queries. FIX_ERROR approach in general cannot work in dynamic energy harvesting systems.

The third and the fourth column show results of GREEDY_ADAPT and MIN_VAR [9]. Both have comparable average error margin and energy consumption. However, both do not consider fluctuation of energy harvesting in its greedy online adaptation or offline budget allocation. Therefore, in both cases, sensor node runs out of battery and shuts down for harvesting, leading to failed responses to queries. It is thus necessary to have online adaptation to handle fluctuation of renewable energy. In columns 5 and 6 of table 1, we study the significance of each stage in the QuARES framework: offline budget allocation and online adaptation. We compare our QuARES offline and the whole QuARES with both online and offline stages. As seen from table 1, without online adaptation, QuARES (offline) must shut down the sensor node for 4 minutes and thus failed to response to 64 queries. The system often shuts down during the sunrise when both battery reserves and harvested energy rate is low.

Among all approaches, QuARES with both online and offline stages keeps the system alive for the whole harvesting period, responds to all queries, maintains low error margin from 30-70%, and enables a uniform energy usage across nodes. The results show that while our proposed offline stage in QuARES maximize data accuracy, the online adaption stage is required for successful query responsiveness.

Varying Application Constraints: We evaluated our framework for different application constraint profiles. Application Constraint profile 1 (AC1) maintains low error margin during day time $(\delta < 1.5)$, from 5am to 7pm. Application Constraint profile 2 (AC2) maintains low error margin during night time $(\delta < 1.5)$ from 7pm to 5am. QuARES (table 2) satisfies all application constraints at the trade-off of suboptimal energy budget allocation compared to no constraint case, higher error margin and less energy utilization. Interestingly, the QuARES offline stage also gives immediate feedback to system designers if the given data accuracy constraints are infeasible in the next harvesting period and needed to be adjusted. MIN_VAR, on the other hand distributes energy budget among slot with minimum variance regardless of application constraints, thus do not give any feedback to designers when application constraints are infeasible. Furthermore, the budget is not in accordance with energy demand in different slots, MIN_VAR spends more energy in slots where it should save energy for higher-demand slots in the future and soon runs out of battery and fails to response to queries.

Impact of Battery Capacity: We simulate the application with different battery capacities. Fig. 7a shows average error margin during day time and night time under different battery capacities on a summer day. As seen from Fig. 7a, battery capacity has negligible effect on error margin during the day as the energy supply is abundant. On the other hand, battery capacity has significant impact on error margin during the night. There is no energy harvested during this period and the capacity of battery limits energy saving to maintain data accuracy at night. In addition, we compare results on summer days and winter days (Fig. 7b). Results show that to obtain a same error margin, the battery capacity required on winter days is larger than the battery capacity on summer days as winter night time is longer than summer night time. For example, to achieve average error margin of *1.0* at night requires battery capacity *C = 950000 (mJ)* on a summer day but requires *C = 1400000 (mJ)* on a winter day.

C. Prototype System and Test Bed

We are deploying a test bed of energy harvesting wireless sensor network in Responsphere [11]. Fig. 8 shows our Responsphere infrastructure and prototype of our energy harvesting platform, including a Crossbow sensor temperature, light and acoustic sound), two solar panels and 2 AA batteries. Table 3 shows our measurement at noon at 4 different locations inside and outside a building in Responsphere infrastructure (Floor Plan, Fig. 8). Locations

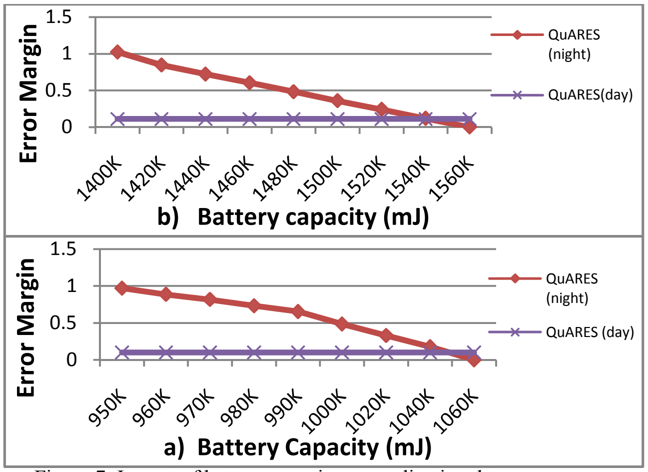


Figure 7. Impact of battery capacity on application data accuracy
a) on summer days b) on winter days

1-2 are indoor while locations 3-4 are outdoor. Under artificial light condition (location 2), solar panels provide low but stable power to sensors. Under natural light (loc. 1 next to a window and loc. 3-4) solar panels provide higher but fluctuating power as it subjects to several conditions such as time of the day, surrounding objects, cloud or wind which may create or shift shadow or direction of windows for indoor case. We plan to use our measurement of solar profile and real sensor data to input to our network simulator. The simulator of data collection application and QuARES framework plays an important role in designing and tuning real system parameters, studying the feasibility of application constraints and energy harvesting as well as system performance.

VI. CONCLUSION

In conclusion, this paper proposes a complete autonomous energy management framework in energy harvesting WSN whose goal is to optimize data accuracy for approximated data collection applications and sustain system operation. The offline stage explores energy harvesting prediction information to allocate energy budget among time slots in a harvesting period and maximize overall data accuracy. The online adaptation stage maintains the predicted data accuracy while coping with harvested energy fluctuation. Our framework is evaluated extensively in comparison with other approaches and considering different weather conditions, battery capacities and application constraints.

REFERENCES

[1] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava, "Heliomote: Enabling Long-Lived Sensor Networks Through Solar Energy Harvesting," in SenSys, 2005.

[2] X. Jiang, J. Polastre, and D. Culler , "Perpetual Environmentally Powered Sensor Networks," in IPSN, 2005.

[3] F. Simjee and P. H. Chou, "Everlast: Long life, Supercapacitor operated Wireless Sensor Node," in ISLPED, 2006.

[4] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava, V. Raghunathan, "Adaptive Duty Cycling for Energy Harvesting Systems," in ISLPED, 2006.

[5] S. Liu, Q. Wu, Q. Qiu, "An adaptive scheduling and voltage/frequency selection algorithm for real-time energy harvesting systems," in DAC, 2009.

TABLE 3: Measurement of Solar Panel output at various location in UCI

	Indoor (1: next to a window)	Indoor (2: inside a room)	Outdoor (3: under shadow)	Outdoor (4: sunny location)
Voltage (V)	16.24	7.17	15.2 - 15.7	17.2 - 17.7
Current (mA)	20.7	2.6	9 - 11	59.4 - 65.2
Availability	12 hours	8 hours	12 hours	12 hours

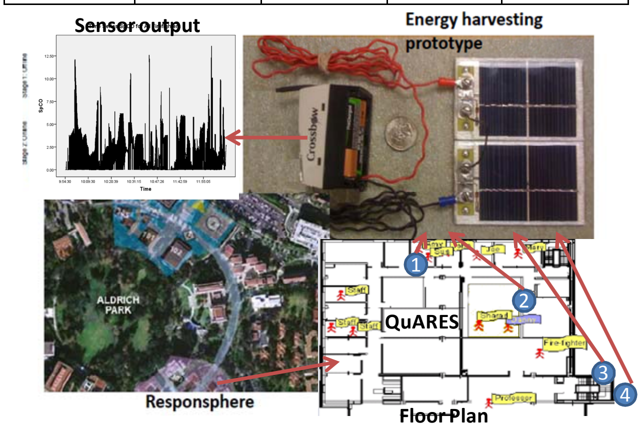


Figure 8: Responshpere infrastructure and solar test bed prototype

[6] C. Moser, D. Brunelli, L. Thiele, L. Benini, "Lazy Scheduling for Energy Harvesting Sensor Nodes," in DIPES, 2006.

[7] T. Voigt, A. Dunkels, J. Alonso, H. Ritter, J. Schiller, "Solar-aware clustering in wireless sensor networks," in ISCC, 2004.

[8] D. K. Noh, L. Wang, Y. Yang, H. K. Le, and T. Abdelzaher, "Minimum Variance Energy Allocation for a Solar-Powered Sensor System," in DCOSS, 2009.

[9] L. Wang, D. K. Noh, Y. Yang, H. K. Le, T. F. Abdelzaher and M. Ward, "AdaptSens: An Adaptive Data Collection and Storage Service for Solar-Powered Sensor networks," in RTSS, 2009.

[10] D. Hasenfratz, A. Meier, C. Moser, J. J. Chen, and L. Thiele , "Analysis, Comparison, and Optimization of Routing Protocols for Energy Harvesting Wireless Sensor Networks," in SUTC, 2010.

[11] Responsphere infrastructure test bed, 17 March 2011, http://www.responsphere.org/index.php

[12] A. Ravinagarajan, D. Dondi and T. S. Rosing, "DVFS Based Task Scheduling in a Harvesting WSN for Structural Health Monitoring," in DATE, 2010.

[13] C. Moser, J. J. Chen, L. Thiele, "Power Management in Energy Harvesting Embedded Systems with Discrete Service Levels," in ISLPED, 2009.

[14] Q. Han, S. Mehrotra, N. Venkatasubramanian, "Energy Efficient Data Collection in Distributed Sensor Environments," in ICDCS, 2003.

[15] C. Bergonzini, B. Lee, J. R. Piorno, T. S. Rosing, "Management of Solar Harvested Energy in Actuation-based and Event-triggered Systems," in Energy Harvesting Workshop, 2009.

[16] M. Ali, B. Al-Hashimi, J. Recas and D. Atienza, "Evaluation and Design Exploration of Solar Harvested-Energy Prediction Algorithm," in DATE, 2010.

[17] C. Olston, B. T. Loo, and J. Widom, "Adaptive precision setting for cached approximate values," in ACM SIGMOD, 2001.

[18] QualNet network simulator. [Online]. "http://www.scalable-networks.com/products/QualNet/"

[19] Crossbow Technology INC. Mica-2 Data Sheet. [Online] Crossbow, http://www.xbow.com.

[20] National Renewable Energy Lab. [Online]. http://www.nrel.gov

[21] B. Zhang, R. Simon and H. Aydin,"Energy Management for Time-Critical Energy HarvestingWireless Sensor Networks" in SSS, 2010.

[22] K Fan, Z. Zheng and P. Sinha ,"Steady and Fair Rate Allocation for Rechargeable Sensors in Perpetual Sensor Networks" in SenSys, 2008.