Energy and Bandwidth-Efficient Wireless Sensor Networks for Monitoring High-Frequency Events

Md Zakirul Alam Bhuiyan[†], Guojun Wang[†*], Jiannong Cao[‡], and Jie Wu[§]

[†]School of Information Science and Engineering, Central South University, Changsha, Hunan, P. R. China, 410083
[‡]Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong
[§]Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
[*]Correspondence to: csgjwang@mail.csu.edu.cn

Abstract—Wireless Sensor Networks (WSNs) are mostly deployed to detect events (i.e., objects or physical changes) at a high/low frequency sampling that is usually adapted by a central unit (or a sink), thus requiring additional resource usage in WSNs. However, the problem of autonomous adaptive sampling regarding the detection of events has not been studied before. In this paper, we propose a novel scheme, termed "event-sensitive adaptive `sampling` and low-cost monitoring (`e-Sampling`)" by addressing the problem in two stages, which lead to reduced resource usage (e.g., energy, radio bandwidth) in WSNs. First, `e-Sampling` provides a solution to adaptive sampling that automatically switches between high- and low-frequency intervals to reduce the resource usage while minimizing false negative detections. Second, by analyzing the frequency content, `e-Sampling` presents an event identification algorithm suitable for decentralized computing in resource-constrained WSNs. In the absence of an event, "uninteresting" data is not transmitted to the sink. We apply `e-Sampling` to structural health monitoring (SHM), which is a typical application of high frequency events. Evaluation via both simulations and experiments validates the advantages of `e-Sampling` in low-cost event monitoring, and in expanding the capacity of WSNs for high data rate applications.

Index Terms—Wireless sensor networks, decentralized signal processing, event monitoring, energy-efficiency

I. INTRODUCTION

Since sensor nodes in future wireless sensor networks (WSNs) are expected to work autonomously to support long-lived and inexpensive acquisition of data from the physical world, the problem of low-power consumption of the nodes is an important issue. Due to severe resource constraints, in particular, energy and bandwidth, a sound body of literature has centered on extending the lifetime of WSNs from different perspectives, e.g., aggressively reducing the spatial sampling rate, the rate assignment, utility-based sensing and communication [1], [2], [3], [4], [5], [6], [7]. Although these schemes all attempt to reduce the energy cost of sensors, they may not be feasible in practice, due to the following serious limitations:

- Sensors cannot take actions independently to adapt their rates and intervals. They all rely on a central unit (or a sink) that knows everything (signal complexities at all of the sensor positions) and periodically transmits suitable sampling rates to them.
- It is difficult to assign sampling rates or allocate bandwidth to sensors in a specific region where an interesting event occurs, especially in the case of 'emergency' alarming applications.

On the one hand, WSNs nowadays are being suggested for many high-rate data collection applications, such as physical activity monitoring, structural health monitoring (SHM), fire event monitoring, etc. The WSNs are expected to monitor events in these applications on a long-term basis. However, sensors generate too much data for their radios in these applications, especially in those that involve audio, seismometers, imaging, and vibration [2], [6], [8], [9]. In most cases, they cannot send that data even one hop in real-time, due to limited bandwidth. Frequent transmission of such raw data results in significant data loss, due to channel contention and congestion. All of these applications bring challenges to sensors' resource circumstances. Thus, sensors should be able to reduce data autonomously before transmission.

On the other hand, natural environments are often extremely dynamic, where the presence of an event is also dynamic. Some events may not appear once in hours, days, months, even years, e.g., damage, fire, snow level, etc. Thus, sensors are required to continuously adjust their activities to dynamic systems (e.g., cyber-physical systems [10]). The challenge is to represent an accurate picture of changes in the event process and environmental variables. This can only be achieved if the event is sensed or sampled from the environment at an accurate rate. Thus, sampling rate should be regarded as a function of both the dynamic phenomena and the application.

We design `e-Sampling` (an event-sensitive adaptive `sampling` and low-cost monitoring scheme), which leads to reduced resource usage in WSNs in two stages. In the first stage, each sensor has "short" and recurrent "bursts" of high-rate sampling, and samples at a much lower rate at any other time. Depending on the analysis of the frequency content of signals, whenever one of the short intervals of high-rate sampling is longer than normal, possibly due to the presence of an event, the frequency content of signals becomes important. Each sensor automatically switches (takes actions on) its rates and both the high- and low-rate intervals. Previously discussed limitations are overcome, as `e-Sampling` enables reliable analysis to estimate appropriate future sampling rates and net reduction in acquired samples.

In the second stage, `e-Sampling` enables sensors to compute a lightweight indication of the presence of an event by analyzing only the important frequency content in a decentralized manner. A significant change in the content (called

event-sensitive or *interesting data*) indicates that a possible event occurred in a given monitoring application. If the event has truly occurred, the sink who receives the indications may want detailed information from the sensors in specific regions (e.g., which are located around the event) and may ask queries; otherwise, in the absence of the event, sensors reduce data (called *uninteresting data*) transmission to the sink.

The major contributions of this paper are four-fold:

- We formulate the problem of *adaptive sampling and low-cost monitoring*, focusing on event-sensitive data, and we design `e-Sampling` to address the problem.
- To make the best use of WSNs for diverse applications, we present an algorithm, which is adaptive to adjust the sampling rate autonomously and is computationally inexpensive, and it does not require the sink's mediation.
- We present an event identification algorithm suitable for energy-constrained WSNs. For this reason, we study SHM as a typical application of high frequency events.
- We demonstrate the effectiveness of `e-Sampling` in simulations with data traces collected by a real systemunder an SHM project[1] of Hong Kong PolyU. Also, we implement a prototype WSN system by using TinyOS [11] and Imote2 sensors [12], and deploy it on a specially designed infrastructure to validate our algorithms.

This paper is organized as follows. Section II reviews related work. We formulate our problem in Section III. Section IV briefly explains the design of `e-Sampling`. Section V presents the adaptive sampling algorithms. Section VI provides a decentralized event indication algorithm. Evaluation through both simulations and experiments is conducted in Section VII. Finally, Section VIII concludes this paper.

II. RELATED WORK

Adaptive sampling has been a resource management issue in WSNs [1], [2], [3], [5], [6], [13], [14], [15]. Optimal sampling in WSNs focuses on how to assign the sampling rate under given bandwidth constraints [1]. The most closely related work, called FloodNet [14], provides the application-layer design of WSNs, where the sampling rate is adjusted according to the estimation error or regression accuracy of the physical phenomenon measured [14]. It requires a daily pattern, and the sampling rates of a specific hour, for estimating the sampling rates of the very same hour of the next day. It maintains an acceptable signal reconstruction at the sink to detect an event. The data collection at a fixed period of time (e.g., an hour), and then adjustment of sampling rates based on the analysis of a large amount of data, consumes significant energy. Such adjustment may not be applied to many dynamic or high-frequency events (e.g., an event of fire, damage/crack, or a physical activity), which may occur for a short period (e.g., 5 seconds). However, it has serious limitations on the quality of signal frequency, which varies largely from time to time, and also changes suddenly with dynamic events [7].

In another study [13], an optimization problem corresponding to maximum entropy sampling is formulated, which proposes a solution to the problem, using factor masks. Utilization of application-specific filtering for data reduction before transmission is effective for high rate applications [15]. However, the filter behavior is hard to predict, and applications perform poorly if filtering is too aggressive or poorly calibrated. Backcasting is a prominent method that operates by activating only a small subset of nodes that communicate their information to the sink [4]. This provides an initial estimate of the sensed environment, and guides the allocation of additional WSN resources. A camera-based WSN provides two main concerns of visual monitoring [6]: (i) it increases robustness to unpredictable events by adding redundancy; (ii) it reduces energy cost on each node by distributing image sampling tasks among neighbors. However, the nodes are allocated a fixed bandwidth and sampling rate, where they should keep long-range communication links. Waiting for the allocation from the sink from time to time, and/or periodically, is expensive.

A recent work proposes a quality-aware adaptive sampling ('QAS' for short) algorithm, which improves the performance of an existing algorithm proposed in [10] in terms of energy consumption and data quality. In the algorithm, when a node experiences stability in its environmental condition, it reduces its sampling frequency. By doing so, the number of data transmissions between the nodes and the cluster head (CHs) are assumed to be reduced. In fact, the sampling rate adaptation requires a lot of data transmission in each cluster and brings a burden on a CH. Moreover, it is also difficult to provide a specific sampling rate to a region of interest, where an event occurs. In the evaluation, we found that, once the sampling rate becomes high, in the absence of an event, achieving the lowest sampling rate is not possible in practice in these schemes.

Applying the methods above, if one wishes to analyze event information at the sink, this mechanism is not suitable for some real-time applications, especially for high-rate and emergency alarming applications (e.g., the event of a fire or damage, intrusion detection, road patrolling, etc.), where the alarm is stringently required to be announced with very low latency. The drawbacks of most of the existing sampling methods are that the real-time requirement is not taken into account. Enabling nodes to rely on the neighbors, the CH, or the sink to assign or estimate their rates requires a lot of packet transmission separately, thus inducing network latency and energy cost in each round of monitoring.

The key aspect that differentiates `e-Sampling` from the prior efforts lies in both data acquisition and decision making. `e-Sampling` allows an ongoing estimate of frequency content. Sensors adjust sampling rates independently, and do not wait for the sink or neighbors' interruption.

III. PROBLEM FORMULATION AND MODELS

Consider a set S of N sensor nodes given to monitor events in a data-intensive WSN application. Assume that they are deployed by some generic deployment strategy (such as uniform, random, or deterministic [16]) at feasible locations

TABLE I
SYMBOL/VARIABLE DESCRIPTION

Symbol	Description
F_h	high frequency content of input signals
R_h	sampling at a high rate
R_l	sampling at a low or adjusted rate
R_m	minimum required sampling rate
R_c	current sampling rate (i.e., continue activities at this rate)
d	time interval index, $d = 1, 2, \ldots$
D_h	the duration of each burst of high-rate sampling
D_l	the duration (short interval) of low-rate or adjusted sampling
I	given a whole monitoring round or time interval

Fig. 1. Illustration of one dimensional signal indicating, sampling interval, sampling points, and how to sample it.

$L = \{l_1, l_2, \ldots, l_N\}$, where sensor i is placed at location l_i, and l_0 is a suitable location of the sink. Let X be the communication range, where the maximum and minimum communication ranges of a sensor are X_{max} and X_{min}, respectively. X_{min} is used to maintain local topology, where every pair of sensors within X_{min} are allowed to share and compare their decision with their neighbors. Sensor i corresponds to a node, and any two nodes are connected if their corresponding nodes in X_{min} can communicate directly. The intention of adopting adjustable X is to reduce energy costs for frequent long distance transmission. Advanced sensor platforms, such as Imote2, support discrete power levels [12].

A. Energy Cost Model (E_i)

One objective is to minimize network energy cost, hence to maximize the network lifetime. We achieve it by reducing the total energy cost, denoted by E_i, on a sensor i in different aspects, including sampling, analog to digital conversion (ADC), computation, and communication. At first, we briefly describe how energy is consumed by a sensor communication component in packet transmission/reception. The maximum energy cost of a sensor depends on a routing protocol used by the data collection application. This falls into the domain of power aware routing [17], [18]. Consider a routing algorithm [18]: we define $q[i]$ as the ith hop sensor on the path q, and γ_q as the amount of traffic flowing along path q within each round of monitoring data collection. Then, $\overline{q[i]q[i+1]}$ is the distance between any two sensors $q[i]$ and $q[i+1]$. $\overline{q[i]q[i+1]} \leq R_{min}$ is used for data delivery to the neighbors and $\overline{q[i]q[i+1]} \leq R_{max}$ is used for data delivery to the sink. Let e_s and e_r be the energy cost for receiving and transmitting data, respectively. Thus, E_i is decomposed into the following parts:

$$E_i = e_t + e_{comp} + e_{ADC} \tag{1}$$

(i) e_t is the total energy cost per bit for transmission over a link between a transmitter and a receiver. Hence,

$$e_t = \sum_{\forall q, \exists i, q[i] = l_i} \gamma_q \cdot e_s \overline{(q[i]q[i+1])} + \sum_{\forall q, \exists i, q[i] = l_i} e_r(q[i]) \tag{2}$$

We do not consider the distance between two nodes when calculating energy cost for receiving data.

(ii) e_{comp} is the energy consumed by the computation that is mainly due to the onboard processor, such as a micro-controller, DSP chip, or FPGA [19]. These devices consume energy proportional to the number of processing cycles, as well as the maximum processor frequency f, switching capacitance μ, and hardware specific constants k and β, respectively
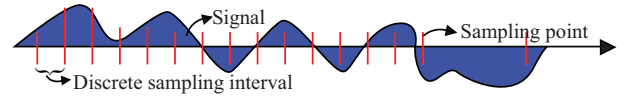
[19]. The number of cycles required to perform a task on the amount of samples (m) are estimated according to the computational complexity $O(m)$, which describes how many basic operations, i.e., averages, additions, multiplications, etc., must be performed in executing the task. The computational energy to complete a task can be calculated according to:

$$e_{comp} = O(m) \cdot \mu(\frac{f}{k} + \beta) \tag{3}$$

(iii) e_{ADC} is the energy consumed by the ADC. In the sampling, two of the modules are the most important, namely the ADC and the sensor itself, when they need energy. As in most cases, if the samples come at fixed time intervals, the average energy can be related to the energy per sample and the number of samples acquired. However, in the case of event-sensitive adaptive sampling in `e-Sampling`, the energy cost can vary due to m and R_c (for symbol description, refer to TABLE 1). Thus, e_{ADC} is proportional to m and the sampling rate used [20], [19].

We calculate the remaining energy (E_i^{rem}) reserved for sampling on sensor i at the beginning of a given monitoring round I. Let E_i^{req} be the maximum energy required on i (which is equivalent to E_i) for a set of a actions in I. We define the system lifetime T to be the total number of rounds of monitoring data collection before any battery runs out of energy:

$$T = E_i^{rem}/E_i^{req} \tag{4}$$

B. Problem in `e-Sampling`

Given a set S of N sensors for monitoring events, $S = \{s_1, s_2 \ldots s_N\}$. Each sensor s_i has a actions it can perform, and these actions are denoted as $A_i = \{R_i^1, R_i^2, \ldots, R_i^a\}$. In the adaptive sampling context, these actions represent sampling rates that i opts to perform at any particular point of time within T, and adjusts its D_l and D_h. This means that s_i is allowed to adjust its actions by analyzing its recent samples of F_h and the samples it believes it will observe.

The following are the key constraints: (i) *action selection constraint*—a sensor i can only select one sampling action at any particular point of time; (ii) *energy constraint*—a sensor i requires a certain amount of energy E_i^{req} to take a sampling action in I that must not exceed E_i^{rem}, i.e., $\sum_{o=1}^{a} R_i^o E_i^{req} \leq E_i^{rem}$; (iii) *computational complexity*, $O(m)$.

Key objectives are to minimize $\sum_{i=1}^{N} E_i$ and overall monitoring latency, and to maximize T.

IV. DESIGN OF OUR SCHEME

We design `e-Sampling` as follows. In the beginning of an interval, sensors start short and recurrent bursts of sampling at a high-rate (R_h), and examine these samples to analyze F_h. The sensor probes the entire bandwidth that is opposite of

```
1 DecentralizedControl{         //1st stage data reduction
2 While (True) {
3   S.RateComp in D_h = True {  // start Sampling Rate Computation at the
                                beginning of the system or at a certain interval
4   Run the Algorithm 1;      // Sampling rate and interval adaptation
5   Compute new R_c;    //set a new sampling rate
6   Compute D_l;}}}     //set the duration for the new rate
7 ComputeEventIndication{       //2nd stage data reduction
8   Run the Event Indication Algorithm
9     If indication.Strength ≥ 40%
10       transmit the indication;
11     else transmit an acknowledgment;}
```

Fig. 2. Overview of `e-Sampling` showing two stages data reduction.

Algorithm 1: Autonomous Adaptive Sampling
(Rate and Interval Adaptation**)**

checking only the bandwidth visible at some sampling point (see Fig. 1) when sampling at another time at a low/adjusted rate (R_l). D_h is the duration of each burst of sampling at R_h that is followed by D_l, where the sampling rate used is calculated based on findings in F_h. The time between two neighborhood sampling points is called a 'discrete sampling interval.'

Whenever an event occurs, F_h becomes important (i.e., the changes in this F_h is large, which implies that there may be an event) and D_h is long enough. Thus, the sampling rate is kept at R_h until F_h is unimportant. In this case, D_l shortens in this interval. Before F_h is known, D_h is followed by a relatively long D_l.

Once F_h becomes unimportant, the sampling condition is again relaxed: D_h shortens and D_l lengthens. Thus, D_h and D_l, and R_h and R_l are automatically switched, depending on F_h. Using this technique in event detection, an analysis of F_h is preserved in each discrete interval and, subsequently, a better sampling rate is selected, while reducing false negative detections. This reduces the energy cost of sensors in almost all aspects (e.g., sampling, ADC, computation, and transmission).

The procedures of `e-Sampling` are simply shown in Fig. 2, and are executed by each sensor node individually. `e-Sampling` reduces the amount of data in two stages: during the period of sampling (lines 1-6), and during the period of decision-making on an event (lines 7-11). Both stages are performed at individual sensors. The ADC task is performed within an interrupt routine, and the main program performs the sampling rate selection computation. When enough samples are acquired, "DecentralizedControl" is executed to select a sampling rate. All of the sets of F_h are stored in the sensor local memory (or flash memory). A sensor may keep them within the memory until it receives a confirmation message from the sink, or the memory is full. After each complete sampling period I, a sensor calculates an event indication.

V. ADAPTIVE SAMPLING: 1ST STAGE DATA REDUCTION

In this section, we propose the autonomous adaptive sampling algorithm (Algorithm 1). In the algorithm, in step 1, a sensor first starts acquiring samples at a high rate, and stores them into the buffer. We need to discuss the sampling interval: how to adjust the interval for a duration in which a sensor takes samples at a high or low sampling rate. For both sampling rate and interval adaptation, each sensor acquires data in d-th
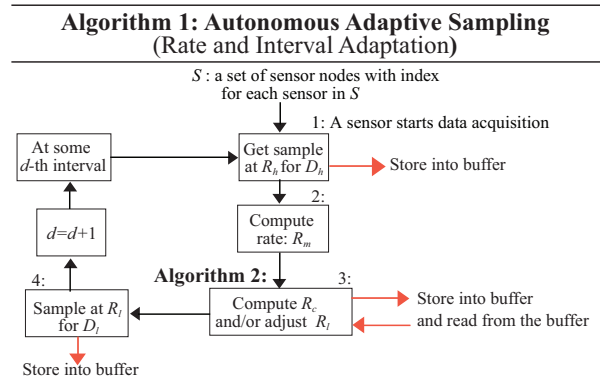
given time interval. Each of such intervals consists of two sub-intervals: (i) D_h starts with a short investigative sub-interval in which an R_h is adopted; (ii) D_l (the remainder of the interval in each time interval) starts when the sampling rate is adjusted to a lower rate, based on the frequency content. Thus, the adjusted R_l for D_l is adopted based on the required rate, R_m.

An analysis is performed on the data points acquired during D_h to estimate the highest frequency content F_h. After analyzing, the sensor executes step 2. If the sensor discovers that F_h is important, then the minimum required rate is the high rate, i.e., $R_m = R_h$. It is determined as:

$$R_m = c \cdot F_h \qquad (5)$$

where $c \geq 2$ is a confidence factor chosen to satisfy Shannon's sampling theorem [3]. However, F_h is generally unknown before sampling, due to the absence/presence of an event. To know this, step 3 is executed and contains Algorithm 2, which helps to get the current rate, R_c, and to make a decision whether or not the current rate should be continued or adjusted. If F_h is still important (there is possibly an event), the sensor continues sampling at the current rate, $R_m = R_c$; otherwise, it adjusts the rate to a lower rate, R_l. After getting the lower rate, the sensor continues sampling at this rate in step 4, until the next d-th interval.

For increased robustness, R_h or R_l is estimated by taking into account the sampling rate in the k previous intervals:

$$R_c = \max(R_m(d - j)), \quad j = 0, ..., k - 1, d \in I \qquad (6)$$

For simplicity, the sequence of operations in adaptive sampling, intervals, sub-intervals, and resulting dynamic sampling rate are presented in Algorithm 1. In the d-th interval, either D_h or D_l can be zero (see the right plot of Algorithm 1).

A. Sampling Rate Selection

In Algorithm 1, the analysis on the samples of F_h helps compute a choice of sampling rate in a fast and efficient manner. We can achieve it through splitting the samples.

1) Wavelet Packet Decomposition for Signal Splitting: We split the acquisition of samples into frequency bands at a sensor's current sampling rate to estimate F_h. The wavelet packet decomposition (WPD) technique is used for this purpose, which is leveraged to break the signal into frequency bands (see Fig. 3). The wavelet packet transform decomposes the set of samples into separate frequency bands by recursively applying high-pass ($H(z)$) and low-pass filters
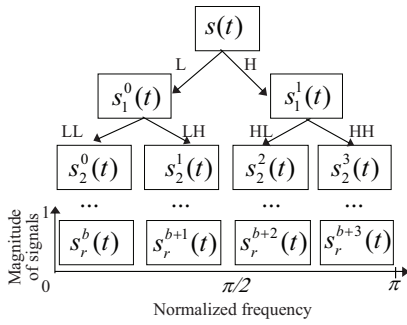
Fig. 3. The two-dimensional wavelet packet decomposition.

$(L(z))$, where $H(z)$ and $L(z)$ are the z transforms of finite impulse response (FIR) filters $h[n]$ and $l[n]$ [21]. A complete sub-band decomposition can be viewed as a decomposition of the acquired signal, using an analysis tree of depth, $\log r$. A tree-based WPD is illustrated in Fig. 3. The filters are desired to be power complementary, meaning that the summation of their responses is equal to one over the frequency domain. This is why we choose wavelets that have as few vectors as possible. The effect of scaling functions in wavelets is a kind of filtering operation on signals, which acts as a low-pass filter, screening out F_h. This ensures that all signal information is being examined equally.

In Fig. 3, the decomposition can continue downward multiple levels r, generating $b = 2^r$ frequency sub-bands. These indexes r and b are exploited to distinguish each sub-band in the wavelet packet tree, $s_r^b(t)$. The low-pass and high-pass filtering operations, which generate the sub-bands, are explained mathematically as discrete convolutions:

$$s_{r+1}^{2b}[n] = \sum_p h[2n - p]\, s_r^b[p]$$
$$s_{r+1}^{2b+1}[n] = \sum_p g[2n - p]\, s_r^b[p] \tag{7}$$

Notice that the signal is decimated by a factor of two after each filtering operation, indicated by $2n$ in each equation. This means that half of the coefficients are discarded after filtering, and the total number of coefficients in all bands at each level r is about the same for each level. This decimated wavelet packet formulation enables greater computational efficiency by keeping the number of coefficients about the same for each level of decomposition, although the number of bands increases by a factor of two. However, the filters are never perfect half-band filters. Therefore, proper construction of the filters is critical to truly represent the signal in these bands, due to the aliasing effect induced by decimation, and the slightly overlapping frequency responses as illustrated in in Fig. 3. Specifically, filters should be selected as the decomposition filters of a quadrature mirror filter (QMF) bank [22]. These are two out of a set of four filters: two decomposition and two reconstruction filters, which meet the criterion of being power complementary, as well as cancelling out any aliasing due to decimation [21]. For reconstruction, at each level the series $s[n]$ is transformed into another of the same length, i.e.,

$$\{s[n]\} \to y[n]$$
$$= \{\sum_p s[p]h[n - p]; \sum_p s[p]l[n - p];\} \tag{8}$$

Algorithm 2: **Sampling Rate Selection**

$U=0$ //There is an update of sampling rate;
$s_0^0(t) = S(t)$;
While $r < R$ && $U = 0$ **do** :
  {
    Compute Eqn. (7);
    if (there is F_h in $s_{r+1}^{2b+1}(t)) > \omega$ for any t
      $s_r^b(t) = s_{r+1}^{2b+1}(t)$;
      //there is possibly a situation of the presence of a physical event)
    elseif $s_{r+1}^{2b}(t) > \omega$ for any t
      $s_r^b(t) = s_{r+1}^{2b}(t)$;
      //there is possibly a situation of the presence of a physical event)
    else $U=1$;
  }
  $b = b_h$;
  Compute R_m //using Eqn. (10);
  Set $R_c = R_m$;
end while

B. Threshold and Estimation of Sampling Rate

The filters break the signals into 2^r frequency bands, as previously discussed using the WPD. Once the bands are generated for some r level decomposition, where there will be 2^r bands, coefficients in all bands are subjected to a threshold ω which is a function of the data size m, noise variance σ, and error constant τ, taken from the first high-pass band as:

$$\omega = \sigma\sqrt{2\log m}$$
$$\sigma = \frac{median(\{s_1^1(W)|:\forall W \in s_1^1\})}{\tau} \tag{9}$$

Any coefficient not above ω is set to zero and $\tau \in [0.5, 1]$ [23]. Now, the highest band index b with any non-zero coefficient, denoted by b_h, is identified, indicating an estimate of F_h of signals. We utilize this index and compute an appropriate sampling rate, called a control rate, for the signals as follows:

$$R_m = c \cdot \frac{R_h}{2^r} \cdot (b_h + 1) \tag{10}$$

We only identify one band, i.e., the highest band with any content in it, b_h, as the computational load induced by determining a sampling rate. Algorithm 2 completes this task by selectively computing bands; it does this by computing only two bands at each r instead of 2^r bands at each r. If there is F_h in the high-pass band, the high-pass band is passed on and split again. If there is no such F_h, the low-pass band is further split. As a result, the WPD tree is allowed to be traversed; only computing two bands at r, and completely avoiding computing unnecessary branches, the computation time is $O(m)$.

By using Algorithm 2, an appropriate sampling is identified, i.e., R_c is either R_h or R_l in each discrete interval. By implementing this technique, computational complexity is now linearly related to the number of samples needed. The computational complexity in actual filtering becomes only that of the discrete wavelet transform, as approximated by:

$$O(m) \approx \sum_{r=0}^{R} 2\left[\frac{m}{2^{r-1}}Q + Q(Q - 1)\right] \tag{11}$$

Here, Q is the length of the QMF filter. This result is obtained by counting the operations needed to perform the filtering. This compares favorably with the FFT (fast Fourier transform) algorithm, which requires $O(m\log m)$.

VI. EVENT INDICATION: 2ND STAGE DATA REDUCTION

In this section, we describe sensor decentralized computing for providing an indication of the absence/presence of events. In `e-Sampling`, sensors do not need to know the types of events during rate. However, in different applications, they need to know application-specific adaptation, since detection of different types of events requires different sampling rates.

In flood monitoring, the detection of a flood warning level is a rare event. In snow monitoring, status of snow coverage also requires event-sensitive sampling. Snow is a kind of composite event (i.e., ice, water, and air) measured at different frequencies in $\{0.5, 100\}$kHz [3]. Fire is also a composite event that requires very high frequency sampling. Physical activity monitoring is another high-rate application that may require switching components into sleep mode between samples, or adjusting the sampling rate to a lower rate during non-physical activity. Any WSN applications, like above, that acquire data at high rates, will clearly use a lot of resources. We consider monitoring events in a data-intensive SHM application.

A. Structural Health Monitoring (SHM)

Wired sensor networks have long been used by civil or structural engineering domains for SHM. In a typical SHM system, the objective is to monitor structures (e.g., buildings, bridges, aircrafts, nuclear plants, etc.) and to detect possible events (e.g., damage, crack) of them at an early stage, which prevails throughout the engineering domains. SHM algorithms work on data acquired at a high fixed-rate and let sensors work for a fixed period of time, say from 10 minutes to hours or days [10]. The algorithms are carried out globally by the sink to identify events. This global monitoring, with handling a large amount of data transmission, brings difficulties to WSNs.

Based on our experience obtained from collaborations with civil researchers[1], the general concern is whether the WSN-based SHM system can replicate the data delivery functionality of the original wire-based counterpart, have less interest in addressing the constraints of the WSNs, and embed in-network processing algorithms. Therefore, the method of data acquisition in traditional SHM [16], [24] can be improved by our sampling algorithms. `e-Sampling` presents a WSN in which the nodes are equipped with some sensing units for SHM. We consider a 3-axis accelerometer that acquires vibration data. After data reduction in the 1st stage, it is still infeasible for a resource-constrained sensor to transmit a set of F_h. We use the sensor decentralized computing to provide an indication shortly based on F_h, i.e., analyzing those ranges of samples.

A sensor has all of the acquired samples sequenced in the database. Each sensor is designed to maintain a local database. We utilize a query-interface-based database for wireless sensor [25], by which all of the samples are aligned. A sensor first distinguishes the samples in F_h and then reads them from the database. Thus, only the samples of F_h are processed to extract an indication. The indication decision is normalized and graded as the percentage of the strength of the event detection, as follows: $0\% \leq \text{VLOW} \leq 20\%$, $21\% \leq \text{LOW} \leq 40\%$, $41\% \leq \text{MED} \leq 60\%$, $61\% \leq \text{HIGH} \leq 80\%$, and $81\% \leq \text{VHIGH}$. If the strength is more than expected (e.g., "MED" (medium)), a sensor makes a pairwise comparison by sharing it with the neighbors in R_{min}. If the strength is less than that, a sensor does not transmit the indication to the sink.

Let q^{Cur} be the current condition of monitoring the environment (e.g., physical structure). The following three equations are iteratively executed by each sensor during the monitoring:

$$\begin{cases} q^{Cur} = \eta \cdot q_1 - q_2 + \sum_{h=h_1}^{h_n} F(h) \\ q_2 = q_1 \\ q_1 = q_0 \end{cases} \quad (12)$$

where $F(h)$ is the recent set of samples of F_h, $h = 1, 2, \ldots n$; thus, the range of frequencies acquired at R_h is $[h_1, h_n]$. q_1 and q_2 store the results of the two previous iterations, which have been acquired at either R_h or R_l. η is used in the iterations as the application-specific data collection-coefficient. We define the embedded indication function denoted as $\Delta s_u(e)$ in order to ensure that there is absence/presence of an event in the vicinity of a sensor u. Let Ref be the reference sample set of F_h that has been acquired when there is absence of an event in the application. $\Delta s_u(e)$ is calculated as follows:

$$\Delta s_u(e) = \frac{|q^{Cur} - q^{Ref}|}{q^{Ref}} \quad (13)$$

Assuming that a sensor u has $n \in N$ neighbors in R_{min}, we normalize the output of $\Delta s_u(e)$ and put the output into the grade. In this work, a sensor u shares its $\Delta s_u(e)$ with the n sensors if the strength is MED. It can be set as needed by the user. By comparing the strength among all pairs $(u \leftrightarrow v)$ of sensors in R_{min}, it is possible to correctly locate the event.

To allow a sensor to make a decision whether to transmit the indication to the sink or not, we define a transmissibility function given in (14). The function is calculated by:

$$F_{s_u}^{s_v} = \frac{\sum_{v=1}^{n} (\Delta s_v(e))^2}{\sum_{u=1}^{n} (\Delta s_u(e))^2}, \quad n \in N \quad (14)$$

We also normalize the output of $F_{s_u}^{s_v}$ to 100% as the strength of the event indication, which is a confirming decision. If the strength is MED, a node transmits the indication to the sink. This also can be set by the user. The use of these decentralized computations prevents the nodes from transmitting a long sequence of acceleration signals (usually up to 60KB of data for each sensor node) to the sink node for off-line analysis.

VII. EVALUATION

A. Simulation Studies

1) Methods and System Parameters: To evaluate `e-Sampling` in realistic simulations, we adopt real datasets. The datasets consist of the vibration signals collected from a sophisticated SHM system, deployed on a high-rise Guangzhou National TV tower (GNTVT)[1]. These datasets are collected by a set of 200 and a set of 800 sensors. Its basic abstraction and the sample sets efficiently provide for sample processing and application-specific format extensions. In the 200-sensor case, the sensors are deployed in a deterministic manner [16]. We use the datasets for the
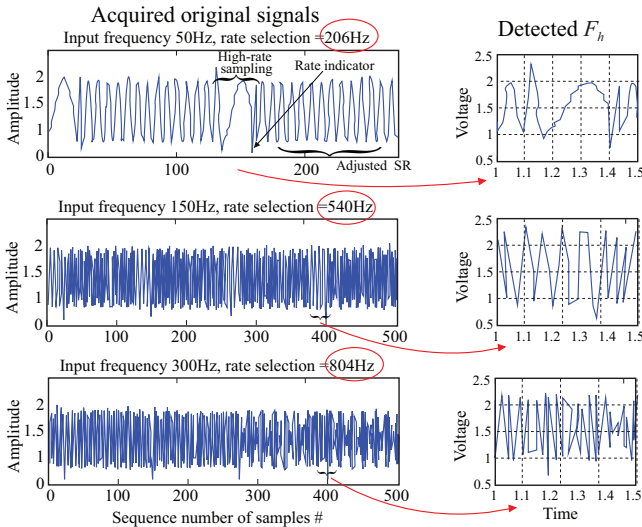
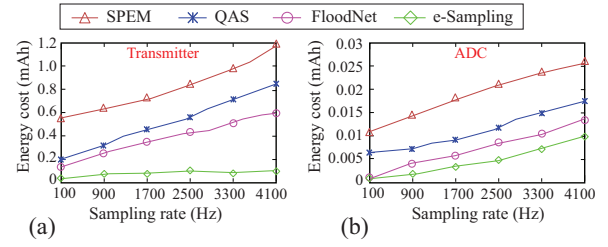Fig. 4. Autonomous sampling rate adaptation of the 4th sensor node.

Fig. 5. Energy cost (of (a) transmitter and (b) ADC) of the 6th sensor at different sampling rates calculated over a monitoring round.
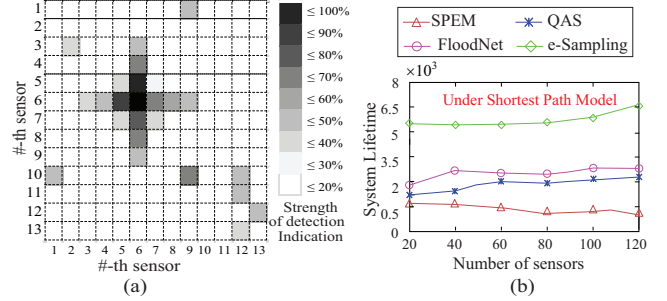

Fig. 6. (a) The strength of indications obtained by the first 13 sensors; (b) network lifetime.

100-sensor case in our simulations. The core of our simulation relies on the sample set data structure that contains data collected at a low to high sampling rate. Filter coefficients tabulated in [22] are the quadrature mirror filters (QMF) used for the sampling rate selection, and of a length up to 16.

Simulations are performed with Matlab Toolbox using a finite element model (FEM) [16] of the structure, and within a $50m \times 400m$ rectangular field, taking into account the area of structural environment, e.g., a high-rise building, bridge, aircraft, etc. We inject different levels of physical event (such as damage) information at 10 sensor locations (by modifying input signals randomly in the datasets).

The hardware constants for the processor and transceiver are from the Intel Xscale PXA271 [12], and the ADC is of a Maxim converter defined for multiple sampling rates [20]. The Imote2 uses a CC2420 radio chip for communication. We model each sensor with six discrete power levels in the interval {-10dBm, 0dBm}, considering the Imote2's power settings, which can be tuned within the IEEE 802.15.4. The study of different routing protocols is out of the scope of this paper; for evaluation, we use the shortest path (SP) routing model. A detailed description can be found in [17], [18]. The constants and parameters are set as follows: processor speed is 13MHz; $\beta = 0.83V$; payload size = 32 bytes; $\overline{q[i]q[i+1]} = 30m$; $\tau = 0.7$; $c = 2.5$; $r \leq 16$.

Comparison: For rigorous comparisons, we implement five schemes as follows. (i) SPEM [16]: this is a SHM scheme validated on the GNTVT and simulated with fixed-rate sampling. (ii) FloodNet [14]: a decentralized control algorithm for information-based adaptive sampling; (iii) QAS [7]: a quality aware adaptive sampling (an improved algorithm given in [26]). (iv) e-Sampling: this proposed scheme. (v) e-Sampling-C: this is a *semi-centralized* version of e-Sampling that excludes the 2nd stage data reduction, i.e., sensors transmit the set of F_h data to the sink directly.

2) Simulation Results: We first study the sampling rate adaptation of a sensor, based on F_h identification in Fig. 4.

For example, we analyze the results on the acquired vibration signals by the 4th sensor, each of which is a sine wave in the range of F_h (the left-hand plots). It is seen that for each signal, the correct F_h and sampling rate is selected by the algorithm. After analyzing the set of F_h at different selected rates (circle marked), important F_h in the periodic stretched out portions indicates D_h (the right-hand plot). This indicates that such high frequency content may convey event information.

We next study the energy cost and energy saving of different hardware components of a sensor, as shown in Fig. 5. By using the datasets with varying sampling rates (between 250 and 4100 Hz), SPEM consumes energy at the rates about 0.54mAh. It is interesting to mention that when the sampling rate is about 250 Hz (=R_c) and a burst of sampling rate reaches 4000 Hz (= R_h) and gets back again to 250 Hz ($R_c \rightarrow R_l$), the energy cost reduces to about 0.07mAh in each interval; hence, saving about 87%. In the same situation, we found that R_l =1110 Hz in QAS and R_l =960 Hz in FloodNet. This indicates that the selected minimum sampling rate is still high in these schemes, although there may be no event, i.e., the lowest sampling rate cannot be selected in these schemes. The energy cost reduces to about 0.02mAh in FloodNet and about 0.015mAh in QAS, which is very small, compared to e-Sampling. SPEM monitors events at fixed rates, which consume higher energy than both FloodNet and QAS, while e-Sampling outperforms all of the other schemes. The components (e.g., transmitter, ADC) in e-Sampling consume less baseline energy than others.

We next examine the performance of the event indication. Fig. 6(a) demonstrates the results under event injection, obtained from the first 13 sensors. We see the highest strength of event indication at the 6th sensor location among those sensors. In Fig. 6(a), the strength of the 6th sensor is VHIGH (i.e., more than 81%) and its neighboring sensors also show a

(a) An integrated SHM mote   (b) Test building structure   (c)
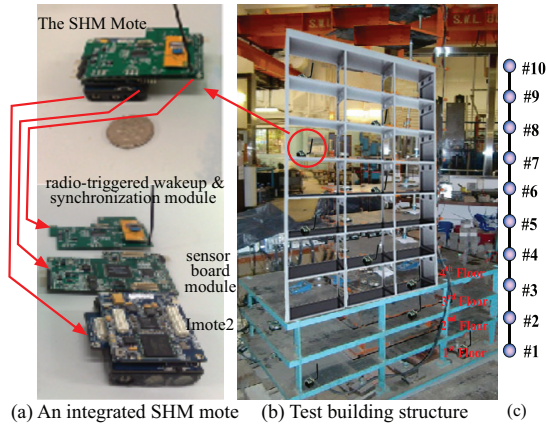
Fig. 7. (a) The SHM mote integrated by Imote2; (b) twelve-story test infrastructure and the placement of 10 SHM motes on it; (c) network topology.

high degree of strength. The strengths of the neighbors hints that the presence of an event is highly possible in the vicinity of the 6th sensor. This is why the 3th, 4rd, 6th, and 7th sensors obtain an extent of strength, say, more than 40%. Thus, they also transmit the indication to the sink, while 1st, 2nd, and 10th to 13th sensors do not transmit, even if they do not participate in pair-wise decision sharing.

Looking for more details on the WSN performance, Fig. 6(b) depicts that the lifetime increases with the number of sensors. We can see, as expected, e-Sampling performs much better (from 55% to 70%) than the other four schemes. FloodNet slightly outperforms QAS. Their poor performance hints that data collection under event-sensitive sampling and decentralized computing is oblivious to the demand from high-rate applications.

B. Proof-of-Concept System Implementation

1) WSN Deployment for SHM Applications : we implement a proof-of-concept system using the TinyOS [11] on Imote2 sensor platforms [12]. We specially design a test infrastructure in our lab and deploy 10 SHM motes (Integrated Imote2) on it, as shown in Fig. 7(a). Three abilities of e-Sampling are justified experimentally, whether or not (i) lowering the sampling rate can lead to a maximized system lifetime, (ii) e-Sampling can identify the correct sampling rate in the presence of an event, and (iii) a node is able to automatically and autonomously adjust its sampling rates and intervals.

An additional Imote2 as the sink node is deployed 30 meters away. A PC as a command center is used for the sink and data visualization. Each mote runs a program (implemented in the nesC language) to process the acceleration signal acquired from on-board accelerometers (LIS3L02DQ). The digital acceleration signal is acquired within frames of 4,096 data points and is then stored in the local memory for each round of monitoring. The LIS3L02DQ is with a built-in ADC, which is followed by digital filters with selectable cutoff frequencies. According to civil engineering, SHM can also be performed by acquiring data within frames of more or less than 560 data points. We consider 4,096 data points to meet higher requirements posed by diverse WSN applications.
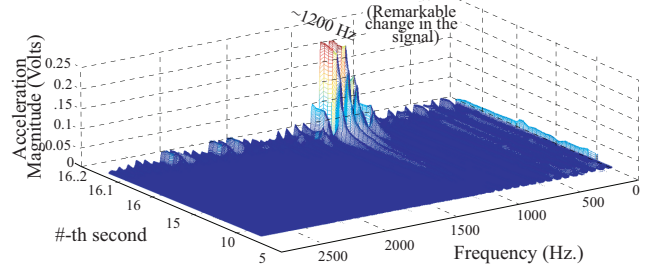

Fig. 8. Waterfall plot of frequency content vs. time.

TABLE 2
Performance of Energy Saving in the WSN-based SHM

Enegy Saving	Deployed sensor no. #									
	1	2	3	4	5	6	7	8	9	10
In the 1st stage	42.3%	27.2%	15.4%	12.1%	21.9%	25.7%	33.8%	62.1%	62.2%	75.1%
In the 2nd stage	53.8%	48.2%	22.5%	13.2%	17.3%	23.5%	43.1%	59%	83.1%	85.2%
Energy Wasting	(-) 4.2% (on average)									

We inject events at the 4th sensor location by removing the plate from the 4th floor of the structure to observe the sampling rate adaptation and event indication. The sensor attached on the 4th floor and its neighbors are expected to provide an indication. We also inject a high-magnitude manual excitation on the structure at some point of time by using a hammer.

2) Experiment Results: when we inject an artificial event on the structure by the hammer strike, e-Sampling can successfully identify F_h at the point of time. The sampling rate and corresponding D_l is also adjusted accordingly. This leads to an increase in the adjusted sampling rate (1100 to 1400 Hz). For example, this can be best observed by examining the sampling rate waterfall plot in Fig. 8. The plot is depicted according to the rate adjusted at a point of time. When the sampling rate is minimum (from 0 to 150 Hz), D_l is relatively long. The data set (m) analyzed during D_h is taken to be 4,096 data points. By using (3), the Imote2 at 13 MHz completes the task in an estimated 0.2 ms.

As with the structural event injection, the sampling rate varies with F_h. The adaptive sampling is performed on all of the 10 sensors' signals, and the results of energy saving caused by data transmission reduction are shown in TABLE 2. Considering the cases like in Fig. 8 and TABLE 2, it is evident that the amount of data is significantly reduced, based on the changing bandwidth requirements of the sensors. A reduction of up to 82% of data is seen at some sensors, in comparison to the 560Hz fixed sampling rate in SPEM. Adaptive sampling enables a net data reduction of 37.4% for the entire acquisition signal, translating to a predicted 34.7% energy saving. This reduction is due to decreased hardware activities in the 1st stage. This translates to a reduction in the energy cost from 22.6mAh to 17.5mAh. The event indication enables another net data reduction of 44.6% in the 2nd stage, translating to a 41.38% energy saving. About 4.2% energy wasting is estimated from errors, query, and data copy from/to the buffer.
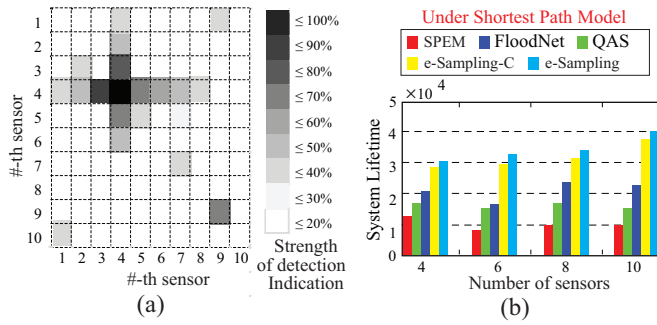
Fig. 9. (a) The strength of event indication by the first 10 sensors; (b) the system lifetime vs. the number of sensors..

Fig. 9(a) depicts the strength of the event (i.e., damage) indication at each node, with a comparison between pairs of nodes. For monitoring, the sampling rate is adapted up to 2500 Hz in the experiment. The obtained strength of the 4th sensor is VHIGH (i.e., more than 80%) and its neighboring sensors show some degree of strength between MED and HIGH.

Analyzing the results of energy saving in TABLE 2 and energy cost, the WSN lifetime is depicted in Fig. 9(b), which is evident that this event-sensitive adaptive sampling scheme, `e-Sampling`, has high energy-efficiency in the WSN, compared to all of the schemes. In SPEM, while collecting data at a fixed rate and transmitting the data in each round, frequent retransmissions for the packet losses decrease energy saving. QAS requires a slightly higher energy cost for all aspects other than FloodNet, except computation.

VIII. Conclusion

We have designed `e-Sampling`, a novel scheme of adaptive data acquisition and low-cost monitoring in WSNs, as an alternative to the traditional event-insensitive schemes. `e-Sampling` is capable of high-rate data acquisition and multi-hop wireless transmission in an energy-efficient way. It is quite flexible, as it supports diverse WSN applications—all the while it is able to run on small and low power micro-controller-based sensor nodes. Evaluation results show that, when both algorithms of adaptive sampling and decentralized event indication are used, `e-Sampling` saves up to 87% of the energy consumed by Imote2 sensors. There are some limitations in this paper that will be improved in the future: (i) analyzing the performance of the current scheme for monitoring different high-frequency events and the event detection accuracy; (ii) a detailed analysis of the proposed algorithms, a comparison of their performance with more related schemes, under a sophisticated energy model.

Acknowledgment

References

[1] W. Shu, X. Liu, Z. Gu, and S. Gopalakrishnan, "Optimal sampling rate assignment with dynamic route selection for real-time wireless sensor networks," in *Proc. of IEEE RTSS*, 2008.

[2] H. Luo, J. Wang, Y. Sun, H. Ma, and X.-Y. Li, "Adaptive sampling and diversity reception in multi-hop wireless audio sensor networks," in *Proc. of IEEE ICDCS*, 2010.

[3] C. Alippi, G. Anastasi, M. Francesco, and M. Roveri, "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 2, pp. 335–344, 2010.

[4] R. Willet, A. Martin, and R. Nowak, "Backcasting: Adaptive sampling for sensor networks," in *Proc. of ACM/IEEE IPSN*, 2004.

[5] J. Wang, Y. Liu, and S. K. Das, "Energy-efficient data gathering in wireless sensor networks with asynchronous sampling," *ACM Transactions on Sensor Network*, vol. 6, no. 3, 2010.

[6] Z. Chen, G. Barrenetxea, and M. Vetterli, "Share risk and energy: Sampling and communication strategies for multi-camera wireless monitoring," in *Proc. of IEEE INFOCOM*, 2012.

[7] A. Masoum, N. Meratnia, and P. J. M. Havinga, "A decentralized quality aware adaptive sampling strategy in wireless sensor networks," in *Proc. of UIC/ATC*, 2012.

[8] M. Z. A. Bhuiyan, J. Cao, G. Wang, and X. Liu, "Energy-efficient and fault-tolerant structural health monitoring in wireless sensor networks," in *Proc. of IEEE SRDS*, 2012, pp. 301–310.

[9] G. Wang, M. Z. A. Bhuiyan, and Z. Li, "Two-level cooperative and energy-efficient tracking algorithm in wireless sensor networks," *Wileys Concurrency and Computation: Practice & Experience*, vol. 22, no. 4, pp. 518–537, 2010.

[10] G. Hackmann, W. Guo, G. Yan, C. Lu, and S. Dykei, "Cyber-physical codesign of distributed structural health monitoring with wireless sensor networks," in *Proc. of ACM/IEEE ICCPS*, 2010.

[11] TinyOS, http://www.tinyos.net.

[12] Crossbow Technology Inc. Imote2 Hardware Reference Manual, 2007.

[13] S. Burer and J. Andlee, "Solving maximum-entropy sampling problems using factored masks," *Mathematical Programming*, vol. 109, no. 2, pp. 263–281, 2007.

[14] J. Kho, A. Rogers, and N. Jennings, "Decentralized control of adaptive sampling in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, 2009.

[15] B. Greenstein, C. Mar, A. Pesterev, S. Farshchi, E. Kohler, J. Judy, and D. Estrin, "Capturing high-frequency phenomena using a bandwidth-limited sensor network," in *Proc. of ACM SenSys*, 2006.

[16] B. Li, D. Wang, and Y. Q. N. F. Wang, "High quality sensor placement for SHM systems: Refocusing on application demands," in *Proc. of IEEE INFOCOM*, 2010.

[17] S. Olariu and I. Stojmenovic, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting," in *Proc. of IEEE INFOCOM*, 2006.

[18] S. Singh, M. Woo, and C. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. of ACM MobiCom*, 1998.

[19] V. Gutnik and A. Chandrakasan, "Embedded power supply for low-power DSP," *IEEE Transactions on VLSI System*, vol. 12, no. 4, pp. 425–434, 1997.

[20] Application Notes Maxim Corporation, 2-Bit Sampling Analog to Digital Converter Conserves Power. [Online]: http://www.datasheetarchive.com/AN43-datasheet.html.

[21] S. Mitra, *Digital Signal Processing: A Computer Based Approach*. McGraw-Hill College, 3rd edition, 2005.

[22] M. Jairaj and S. Subbaraman, "QMF implementation using xilinx sysgen (XSG)," in *Proc. of IEEE ICCSIT*, 2010.

[23] D. Donoho and I. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrica*, vol. 81, no. 3, pp. 425–455, 1994.

[24] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proc. of ACM/IEEE IPSN*, 2007.

[25] N. Tsiftes and A. Dunkels, "A database in every sensor," in *Proc. of ACM SenSys*, 2011.

[26] S. Chatterjea and P. Havinga, "An adaptive and autonomous sensor sampling frequency control scheme for energy-efficient data acquisition in wireless sensor networks," in *Proc. of IEEE DCOSS*, 2009.