2016

Adaptive Sensor Node Sleep Scheduling for Quality-of-Experience Enhancement

Mini Mathew
University of Illinois Springfield

Ning Weng
Southern Illinois University Carbondale, nweng@siu.edu

Recommended Citation

Mathew, Mini and Weng, Ning. "Adaptive Sensor Node Sleep Scheduling for Quality-of-Experience Enhancement." *International Journal of Distributed Sensor Networks* 2016 (Jan 2016). doi:10.1155/2016/6960876.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2016, Article ID 6960876, 11 pages
http://dx.doi.org/10.1155/2016/6960876

Research Article

Adaptive Sensor Node Sleep Scheduling for Quality-of-Experience Enhancement

Mini Mathew[1] **and Ning Weng**[2]

[1] *Department of Computer Science, University of Illinois, Springfield, IL 62703, USA*
[2] *Department of Electrical and Computer Engineering, Southern Illinois University, Carbondale, IL 62901, USA*

Correspondence should be addressed to Ning Weng; nweng@siu.edu

Received 4 December 2015; Revised 3 March 2016; Accepted 8 March 2016

Academic Editor: Michele Magno

Focusing on a user's quality-of-experience (QoE) has become important, because of the growing space of sensor-dependent applications and low-cost sensor design. QoE is typically affected by two quantities: the quality-of-information (QoI) received and the lifetime-of-service. Therefore, QoE is defined as a sensor network's ability to consistently offer assured QoI for an expected lifetime when operating on a limited energy resource, such as a battery. However, dynamic factors, such as varying user requirements, unpredictable sensor environment, unreliable network conditions, and limited energy resource, affecting both QoI and lifetime-of-service, make it challenging to achieve a good QoE. In our previous work, we presented a SNR-based QoI metric which addresses the impact of several of these factors on QoI. In this paper, we design a QoE metric that quantifies the relationship between energy conservation, QoI received by a user, and an application's quality expectation. Further, we develop an adaptive sleep schedule mechanism to demonstrate the usefulness of this metric. Finally, simulation results presented show the effectiveness of our mechanism in achieving QoE improvement.

1. Introduction

The advancement in low-cost sensor design has led to the growth in sensor applications [1, 2]. However, there are also various challenges involved in practically implementing these applications. Some of these challenges include short node lifetime, network unreliability, environmental unpredictability, and varying application requirements. Moreover, many of these applications are time-sensitive and can require that the users decide on a response action by assessing the obtained information. Therefore, in practically implementing sensor networks, one key objective to consider is the enhancement in users quality-of-experience (QoE). Considering the nature of the applications and the aforementioned challenges of sensor networks, the two quantities affecting users QoE are quality-of-information and service lifetime. Hence, we define QoE as the measure of *a network's ability to consistently provide assured quality-of-information (QoI) while being operational for an expected lifetime*, even with a limited energy resource, such as a battery.

So far, the extensive research carried out in sensor networks has considered a network-centric approach to address the various challenges. In particular, these works have focused on maximizing the network lifetime together with improving either the Quality-of-Service (QoS) or quality-of-information [3–6]. In recent past, however, the focus has been changing to more of a user-centric approach. For example, [7] presented an adaptive QoE-aware forward error correction mechanism for improving the video-data quality of event-driven multimedia applications, based on a user's experience. The scope of QoE can expand to other safety-critical applications, such as seismic monitoring [8] and target tracking [9]. We envision a sensor network dedicated to satisfying a user's quality-of-experience irrespective of its application. However, due to the multifaceted dynamic challenges mentioned in the paragraph above, defining QoE is nontrivial.

In this paper, we propose a QoE metric as a first step to characterize the interdependency of QoI and energy with respect to user experience. Our metric is defined as the product of energy-saving ratio and quality satisfaction estimating factor. Here, to estimate quality satisfaction, we utilize the QoI metric defined in our previous work [10] and seamlessly incorporate it into this new metric. Further, we propose an adaptive sleep scheduling mechanism that adjusts the sleep-wake schedule of nodes to reduce channel interference, packet delay, and loss. Finally, we present simulation results that validate the effectiveness of our adaptive mechanism in comparison to a typical static mechanism.

The main contributions of this paper are as follows:

(i) A QoE metric which integrates QoI with user-specific quality expectation and energy efficiency.

(ii) An adaptive sleep scheduling algorithm that uses the proposed metric to attain the objective of improving energy efficiency with guaranteed satisfaction of user-specific data quality.

(iii) Simulation results demonstrating the usefulness of the proposed mechanism by comparing it to a typical static sleep schedule mechanism.

2. Modeling Quality-of-Experience and Per Packet Latency

In a sensor network, the two quantities that affect a user's QoE are quality-of-information and service lifetime. Service lifetime, which we characterize as energy efficiency, is a critical requirement for battery-operated sensors. However, addressing the problem of maximizing energy efficiency independent of QoI assurance can prove detrimental to the decision-making process of quality-sensitive applications that directly affect public safety. For example, sleep scheduling is a technique widely utilized to improve energy efficiency [11, 12]. Although assigning a large sleep interval to a sensor node helps improve its lifetime, this also introduces additional delay which can degrade QoI. Modeling this relationship is important to attain efficient energy utilization and quality assurance. But characterizing such a QoE metric is challenging, because there are multiple parameters that affect both quality and energy such as sampling rate and application deadline. In this paper, we propose a simple QoE metric as a first step to address this challenge. The following sections describe our proposed QoE metric model, as well as packet latency model which addresses the impact of a sleep schedule on QoE.

2.1. Quality-of-Experience Model. Equation (1) quantifies our proposed QoE metric. It represents the relationship between energy saving $(1 - e^c)$ gained and quality-of-information satisfaction. Additionally, the quality obtained (q^r) and the expected quality (q^e) as specified by an application define the QoI satisfaction in our metric. This metric is defined based on the interdependency understanding obtained from our previously proposed energy efficiency metric [10]. The energy efficiency metric defined the quality-energy interdependency. The QoE metric proposed here addresses the interdependency in relation to improving the quality of user satisfaction, that is, the maximization of energy saving along with satisfying the minimum expected quality requirement:

$$q^{us} = \delta\left(q^r, q^e\right) \cdot \left(1 - e^c\right). \tag{1}$$

Here, e^c quantifies the energy consumption ratio, q^r calculates the average quality-of-information obtained from the sensors, and q^e represents the expected quality. In this paper, we assume the value of q^e to be application specific. Moreover, $\delta(q^r, q^e)$ in our metric is a binary value that evaluates the energy efficiency obtained in relation to q^e, as given below:

$$\delta\left(q^r, q^e\right) = \begin{cases} 1 & q^r \geq q^e \\ 0 & \text{else.} \end{cases} \tag{2}$$

This parameter provides a boundary to a mechanism focusing primarily on energy efficiency, so that the energy saving it gains is under the condition that the QoI obtained is satisfactory:

$$e^c = \frac{e^{\text{mech}}}{e^{\text{base}}}. \tag{3}$$

Equation (3) represents the energy reduction benefit of an energy-saving mechanism (e^{mech}) with respect to a baseline (e^{base}). In this paper, we focus on sleep scheduling mechanisms; hence, e^{mech} here is the energy consumed by a sleep mechanism (e^{mech}) and (e^{base}) gives the energy consumed by a node if it is not switched into a sleep state. Further, we briefly explain our previously proposed quality and energy models [3] for completeness.

Factors such as sensing quality and packet loss affect the QoI received by an application (q^r). But quantifying q^r, such that it addresses the impact of all these different factors on it, is difficult. In our previous work [3], we presented a fundamental metric for information quality based on signal-to-noise ratio (SNR). The following equation gives the information quality of an individual sensor:

$$q_i^r = \frac{\text{SNR}_i^r}{\text{SNR}_i^e}, \tag{4}$$

where SNR_i^r is the signal quality received by an application and SNR_i^e represents the expected SNR. SNR_i^r given by (5) computes the impact of sensing quality and loss rate (l_i) on the quality-of-information sent by a sensor i:

$$\text{SNR}_i^r = \text{SNR}_i^m - 10 \cdot \log_{10}\left(1 + l_i \cdot 10^{\text{SNR}_i^m/10}\right). \tag{5}$$

Here, sensing quality SNR_i^m is a function of the sensor sampling rate. Loss rate l_i represents the average loss for a node i caused due to its packets missing the application deadline. As aforementioned, if information is not received within a specified deadline, then that information may not hold much importance to a quality-sensitive application. Therefore, we analyze the impact such a loss has on the overall quality-of-information. Equation (6) calculates the loss rate of a node i when its packets miss their deadline:

$$l_i = \frac{\sum_{j=1}^{k} l_j^D}{\lambda \left(T^{s_p} + T^a\right)}, \tag{6}$$

where the numerator calculates the total number of packets with $l_j^D = 1$. The superscript "D" in l_j^D represents the delay introduced due to the node's sleep duration and queuing, which can cause the packets to miss their deadline. The denominator in (6) gives the total number of packets obtained from node i in one sleep-active cycle. λ represents the average rate at which the packets arrive at node i's queuing system and T^{s_p} is the per cycle sleep duration of node i. T^a represents the active duration of node i, which comprises switching, transmit, and receive durations. The assumption here is that all the packets buffered by a node in one cycle are out of the queuing system before the node switches back to sleep state. Furthermore, since this paper considers periodic applications, we use a $D/D/1$ queuing model, in which the packet arrival and service rates are deterministic. Based on this model, the per packet loss probability (l_j^D) related to the missing of deadline and presented in (7) is either 0 or 1. Therefore, the numerator of (6) calculates the sum of the number of packets with $l_j^D = 1$:

$$l_j^D = \begin{cases} 0 & \text{if } D_j \le T^d - \tau_j \\ 1 & \text{if } D_j > T^d - \tau_j. \end{cases} \tag{7}$$

Here, T^d is the application deadline and τ_j is the sampling time of a packet j of node i. D_j is the delay experienced by a packet j of node i in arriving at the base station. We quantify this delay in relation to a node's sleep duration, as explained in Section 2.2.

Equation (8) gives the energy consumed (e_i^{mech}) by a sensor node i when implemented with a sleep scheduling mechanism. The time periods of sleep $T_i^{s_p}$, transmit $T_i^{t_x}$, receive $T_i^{r_x}$, and transition $T_i^{s_w}$ are presented in Figure 1. Power consumed is the product of current and voltage and their typical values are given in Table 1:

$$e_i^{\text{mech}} = T_i^{s_p} \cdot P_i^{s_p} + T_i^{t_x} \cdot P_i^{t_x} + T_i^{s_w} \cdot P_i^{s_w} + T_i^{r_x} \cdot P_i^{r_x}, \tag{8}$$

where T stands for the time duration and P for the power consumed by sleep, active, and transition states. T^{t_x} is the per cycle duration for which a node transmits its packets. T^{s_w} represents the switching time for sleep, receive, and transmit modes. The sleep interval (T^{s_p}) of a node i is the key variable that our mechanism focuses upon to improve QoE. T^{r_x} in our mechanism represents the time spent by a node in receiving a feedback packet of the base station. This control packet is sent by the base station every time a node requests an update in its channel access schedule.

Table 1: Parameters and their values for the energy model.

Symbol	Description	Value
I^{t_x}	Current used in transmission	17.4 mA
I^{r_x}	Current used in reception	18.8 mA
I^{s_p}	Sleep current	0.021 μA
I^{idle}	Transceiver idle current	396 μA
t^{s_w}	Switching time:	
	(i) Sleep-idle (vice versa)	970 μs
	(ii) Idle-transmit (vice versa)	192 μs
	(iii) Sleep-transmit (vice versa)	1.792 ms
V_{sup}	Supply voltage	3.3 V

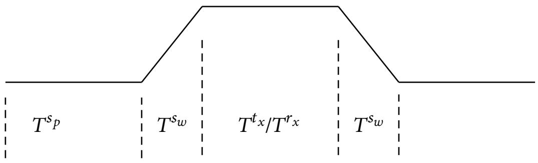


Figure 1: Modes of operation for a node in a single sleep-active cycle.

The energy consumed by a node i with a baseline model used in (3) is given as follows:

$$e_i^{\text{base}} = T_i^{\text{idle}} \cdot P_i^{\text{idle}} + T_i^{t_x} \cdot P_i^{t_x} + T_i^{s_w} \cdot P_i^{s_w}. \tag{9}$$

Here, the energy consumed by a node in an idle state is given as the product of idle interval (T_i^{idle}) and idle power consumption (P_i^{idle}). The transmit and switching energies in this equation are the same as described in (8). Since there is no sleep interval in the baseline model, there is no feedback packet received by the node and hence we do not consider the receive interval $T_i^{r_x}$ for this model. In this model, the transceiver is either in the transmit or in the idle mode. Further, we present an analytical model for calculating average latency introduced due to sleep scheduling.

2.2. Per Packet Latency Relative to Sleep Scheduling. This section analyzes the per packet latency introduced due to the sleep interval of a node. In this work, we assume that a sensor is capable of gathering data even when the node's transceiver is in sleep mode [13]. This introduces queuing delay for the packets being generated. Therefore, modeling this delay is important, because it ensures that the sleep schedule assigned to a node does not cause the quality; it provides for dropping below the application's expectation. Henceforth in this paper, when we mention sleep state, it refers to the transceiver sleep state. Based on the assumption that the sensor performs periodic sampling, we estimate per packet queuing delay using $D/D/1$ model. Figure 2 describes different variables used in the proposed model. The total number of incoming packets of one sleep (T^{s_p}) and active (T^a) cycle is given by λT^{s_p} and λT^a, respectively. Moreover, T^a includes the intervals of switching between sleep-active states, as well as the transmission and reception of packets, which are represented individually in Figure 1.

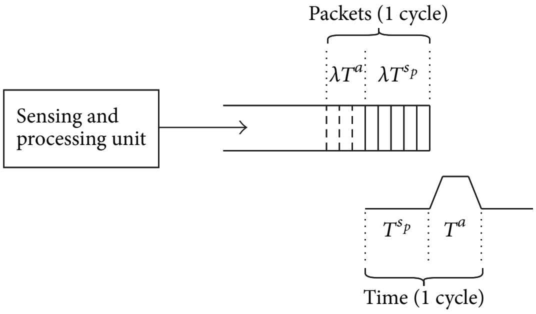


Figure 2: Packet queuing for a single sleep-active cycle.

Equation (10) represents the total queuing latency ($\overline{D}_i$) introduced per sleep-wake cycle for a node *i*. Here, D_j^s and D_j^a denote the queuing delay of a packet j arriving at the buffer during T^{s_p} and T^a, respectively:

$$\overline{D}_i = \sum_{j=1}^{\lambda \cdot T^{s_p}} D_j^s + \sum_{j=1}^{\lambda \cdot T^{a}} D_j^a. \quad (10)$$

Equation (11) gives D^s for a packet j, where the first part $(T^{s_p} - (j-1)/\lambda)$ represents packet j's waiting time and (j/μ) gives its transmission time. λ represents the average rate of packets arriving at the buffer and μ defines the average number of packets transmitted per second during time T^a. Here, we assume that a node's transceiver goes to sleep at time $t = 0$ and the first packet $j = 1$ arrives at the buffer at the same time instant $t = 0$:

$$D_j^s = T^{s_p} - \frac{j-1}{\lambda} + \frac{j}{\mu}; \quad j \in \{1, \ldots, (\lambda \cdot T^{s_p})\}. \quad (11)$$

Equation (12) presents the delay of a packet arriving (D_j^a) during a node's active cycle (T^a). Here, $((\lambda \cdot T^{s_p} + j)/\mu)$ gives the waiting time of a newly arrived packet j during interval T^a and $((j-1)/\lambda)$ provides its transmission time:

$$D_j^a = \frac{\lambda \cdot T^{s_p} + j}{\mu} - \frac{j-1}{\lambda}; \quad j \in \{1, \ldots, (\lambda \cdot T^{a})\}. \quad (12)$$

Using this latency model, our proposed scheduling mechanism determines QoI received by the base station as detailed in Section 3. Though we utilize an analytical model to determine packet delay, our proposed framework is not constrained by this model and can utilize delay obtained through direct measurement.

3. Adaptive Sleep Scheduling Framework

In this section, we present our proposed adaptive sleep scheduling mechanism that achieves maximum energy efficiency with assured QoI for quality-sensitive applications. Figure 3 gives a block diagram of the scheduling and forwarding operation performed by the sensors and the base station. It consists of three main blocks: sensor network, base station, and home station. The base station is the unit that executes our proposed sleep schedule mechanism. The following sections detail the functionalities of each of these blocks.

3.1. Sensor Network. Here, we have considered a single hop sensor network and assume that the sensor nodes are at a unit distance from the base station. Each sensor has two main blocks, namely, the control unit and the data unit. The control unit is responsible for sending meta-data such as sleep time, sampling rate, and measured and expected SNR to the base station. It is also responsible for updating the sleep interval of a node's transceiver unit when it receives a new sleep interval from the base station. It is assumed that, at network setup, when each node's control unit sends the meta-data information, there is no channel collision. Any updated sleep interval is being sent by the base station during the interval when that node has access to the channel. The data unit forwards the sensed information, through the transceiver unit, to the base station. As mentioned before, we assume that a node's sensing unit continues to gather information even while the transceiver unit is in its sleep state.

3.2. Home Station. The home station houses the applications utilizing sensor networks. The applications provide their deadline and expected quality requirements to the base station. In this work, we have considered a single application network. However, it is possible to extend our mechanism to a multiapplication network.

3.3. Base Station. The base station consists of a forwarding unit and a sleep scheduling unit. The forwarding unit receives packets from the sensors and forwards them to the home station. We assume that this unit uses a first-in-first-out approach to forward these packets. Furthermore, we have assumed that the base station is a single, highly powered node. The base station updates a sensor node's sleep schedule if it detects either a channel access conflict or a drop in the information quality below the expectation.

The energy-efficient sleep scheduler is the core part of this framework, which functions as further described. The sensors send their meta-data to the scheduling unit every time there is a change in its sleep-wake schedule. Upon receiving the meta-data from the application and each sensor, the scheduling unit verifies if the overall quality of the network satisfies the application's requirement. In case the QoI received does not satisfy the application's expectation, then the scheduler recomputes the node's sleep interval. Moreover, it also verifies if the requesting node would cause a channel access conflict with any other scheduled nodes. On identifying a conflict, the scheduler adjusts the sleep interval of the node currently requesting for channel access. After these two operations, the scheduling unit sends the newly calculated sleep interval to the sensor node, which then updates its sleep interval based on the obtained value. However, if with the node's initial sleep interval the overall quality is found to be acceptable and no conflict is detected, then the node obtains the channel access according to its requested duration. The base station will then store this node's channel access schedule for future reference. The following section explains the purpose, functioning, benefits, and limitations of our proposed algorithm.

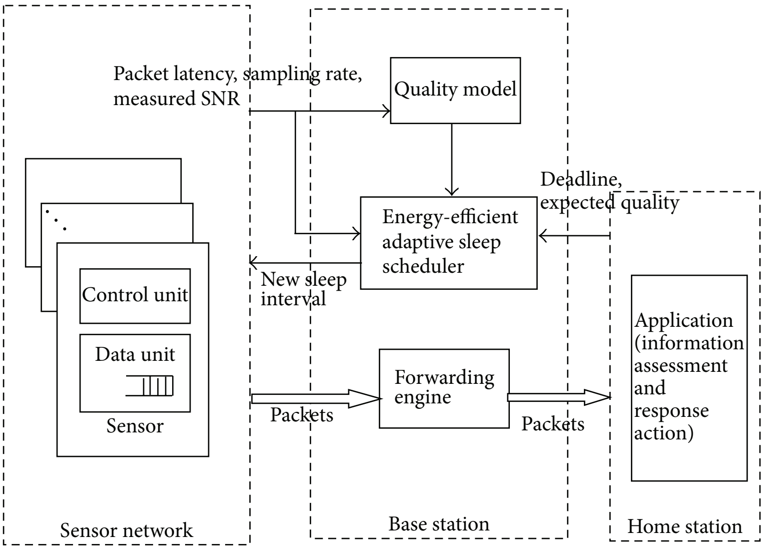

Packet latency, sampling rate, measured SNR
Quality model
Deadline, expected quality
Energy-efficient adaptive sleep scheduler
New sleep interval
Control unit
Data unit
Sensor
Forwarding engine
Packets
Packets
Application (information assessment and response action)
Sensor network
Base station
Home station

Figure 3: Adaptive sleep scheduling framework.

4. Adaptive Sleep Scheduling Algorithm

The objective in designing an adaptive sleep scheduling algorithm is to enable a base station with the capability for adapting a sensor node's sleep interval, to obtain a satisfactory QoI and reduced energy consumption. The proposed algorithm utilizes a simple mechanism of decreasing/increasing a node's sleep duration to ensure that an application receives good quality-of-experience defined in (1).

Algorithm 1 describes the pseudo-code of our proposed adaptive sleep scheduling mechanism. The algorithm has two main functions: quality_management and conflict_resolver. The quality management function calculates the information quality q^r of a node and verifies if it is within the expected quality variance v of an application. Based on the outcome of the verification, it will either increase or decrease the node's sleep interval such that satisfactory q^r is obtained. The conflict resolver function checks if the requesting node's active duration conflicts with any other scheduled nodes. The input to the algorithm is meta-data such as expected quality q^e, quality variance v, $T_i^{s_p}$, and SNR_i^m from sensor nodes and application. Using this information, the base station determines an appropriate sleep interval T^{s_p} and sends it to the node requesting channel access.

On receiving channel access request from a node i, the base station first calculates q^r using its initial $T_i^{s_p}$. It then verifies whether this q^r will satisfy the expectation (q^e). As given in lines (13)–(18) of Algorithm 1, if the received quality's (q^r) variance is greater than the application's tolerable variance v, then the sleep interval $T_i^{s_p}$ of node i is increased by a δ value. This increase benefits energy efficiency. On the other hand, if $q^e - q^r > v$, then T^{s_p} is reduced by δ until $q^e - q^r \leq v$ ensuring a good quality-of-information (lines (19)–(24)). δ in lines (15) and (21) can be either a constant or a variable that depends on v and q^r. In this paper, we have considered a constant δ value.

Following the quality_management operation, the base station checks if the newly calculated sleep interval $T_i^{s_p}$ would cause a channel access conflict with other scheduled nodes, as explained in the conflict_resolver function. On identifying an intersection, as shown in lines (33)–(38), node i's sleep interval is reduced by the interval of its overlap with other scheduled nodes. This will help resolve the conflict and packet loss that can occur if two nodes' active interval overlaps. This function reduces the sleep interval of node i, because the $T_i^{s_p}$ input to this function is $T_i^{s_p}$ updated by the quality_management function. By reducing this updated $T_i^{s_p}$, we can ensure that channel collision is prevented while satisfying the quality-of-information requirement. In line (37), t^a is the time instant when a node (i or m) becomes active and T_i^a is the active interval of node i that is requesting a channel access. This newly calculated sleep interval $T_i^{s_p}$ is then sent to node i which then schedules its sleep-active interval accordingly.

Our proposed algorithm will work effectively in both static and dynamic network setups. Although we have not considered a specific example of dynamic change, for a sensor network having limited energy resource and specific QoI requirement, the objective would still be to select an appropriate sleep interval for the nodes. This will help ensure that the overall QoI is satisfactory along with maximum possible energy reduction. Therefore, our algorithm would be applicable to specific scenarios. However, the current setup has a few shortcomings. Firstly, the network topology is one hop. Applying this algorithm to a cluster-based network is possible; however, for other multihop networks such as a hierarchical network, implementing our algorithm would require considering additional processing and scheduling complexities. Secondly, we assume that the meta-data packet is always received by the base station. Therefore, the impact on QoI if this packet is lost is not considered.

```
scheduling_operation( )
  Input: expected quality: $q^e$ quality variance: $v$
         proposed sleep interval: $T_i^{s_p}$ measured signal-to-noise ratio: $\mathrm{SNR}_i^m$
  Output: new sleep interval of node $i$: $T_i^{s_p}$
begin
    if a node $i$ requests channel access then
        $T_i^{s_p}$ = quality_management($T_i^{s_p}, q^e, v, \mathrm{SNR}_i^m$);
        conflict_resolver($T_i^{s_p}$);
        assign updated $T_i^{s_p}$ to node $i$;
    end
end
quality_management ($T_i^{s_p}, q^e, v, \mathrm{SNR}_i^m$)
begin
    calculate the initial $q^r$ via (4);
    while $|q^r - q^e| > v$ do
        if $q^r - q^e > v$ then
            //improving energy efficiency
            repeat
                $T_i^{s_p} = T_i^{s_p} + \delta$;
                recalculate $q^r$ via (4);
            until $0 < q^r - q^e \leq v$;
        else
            //improving information quality
            repeat
                $T_i^{s_p} = T_i^{s_p} - \delta$;
                recalculate $q^r$ via (4);
            until $0 < q^e - q^r \leq v$;
        end
    end
end
conflict_resolver ($T_i^{s_p}$)
begin
    foreach scheduled node $m$ do
        //Check if there is an active time overlap between node $i$
        //and other scheduled nodes
        while $(t_i^a, t_i^a + T_i^a) \cap (t_m^a, t_m^a + T_m^a)$ do
            //reduce sleep interval of $T_i^{s_p}$ by the overlapped
            //active interval of nodes $i$ and $m$, to avoid channel
            //conflict
            $T_i^{s_p} = T_i^{s_p} - (t_i^a + T_i^a - t_m^a)$;
        end
    end
end
```

Algorithm 1: Adaptive sleep scheduling algorithm.

There is processing overhead involved when performing on-line scheduling. However, this overhead is not significant in our case due to several reasons. Firstly, our proposed algorithm is event-driven reducing the frequency of performing the scheduling operation. Secondly, our algorithm is running on a control plane rather than a data plane; that is, it is not implemented for every packet. Finally, we assume the algorithm is operating at a high-powered base station. To evaluate the processing overhead of our algorithm, we considered a general case where a node's sleep interval causes both quality drop and channel conflict. The processing time required in running our algorithm was 2.57 secs, when implemented on a Linux operating system with Intel Core 2 Duo processor, model number E4500 @ 2.2 GHz, cache memory of 2048 KB, and RAM of 4 GB.

5. Simulation Results

In this section, we present results which demonstrate and validate the features of our adaptive sleeping scheduling framework presented in Section 3. We obtained these validation results by performing simulations using OMNET++ with parameter settings as presented in Table 1. As aforementioned, we assume that a node's sensing unit is always active and it periodically samples data, but its transceiver unit regularly switches between sleep and active modes. Hence, the results presented here analyze the effect on applying different scheduling mechanisms to the transceiver unit.

To achieve a fair evaluation of the adaptive mechanism, we compare it with two cases of the static mechanism, namely, *energy save preference* and *quality preference*. As the names suggest, the static mechanism chooses a sleep interval either to benefit energy (*energy save preference*) or in favor of satisfying quality (*quality preference*). These two cases give us an approximate boundary to verify the usefulness of an adaptive mechanism in saving energy and providing quality satisfaction.

The sections below are organized as follows. Section 5.1 compares the impact of sampling rate on the quality and energy efficiency of the two mechanisms. In Section 5.2, we present the improvement achieved in effective energy efficiency by our mechanism for different application deadlines. Finally, the effectiveness of our adaptive mechanism in providing the required quality-of-experience with different sampling rates is given in Section 5.3.

5.1. Impact of Sampling Rate on Effective Energy Efficiency. Figure 4 demonstrates the impact of sensor sampling rate on QoI, energy saving, and QoE, for both static and adaptive mechanisms. Sensor rate is one of the parameters that affects a sensor's sleep interval selection. For example, assigning a small sleep interval to a node with limited buffer space and a high sampling rate can result in buffer overflow. Moreover, small sleep intervals also reduce a node's lifetime. On the other hand, putting a node to sleep for large intervals can save energy, but it can cause packets to miss their deadline and reduce the user's QoE. Hence, selecting a node's sleep schedule with respect to its sampling rate will help satisfy the overall QoE for a user application.

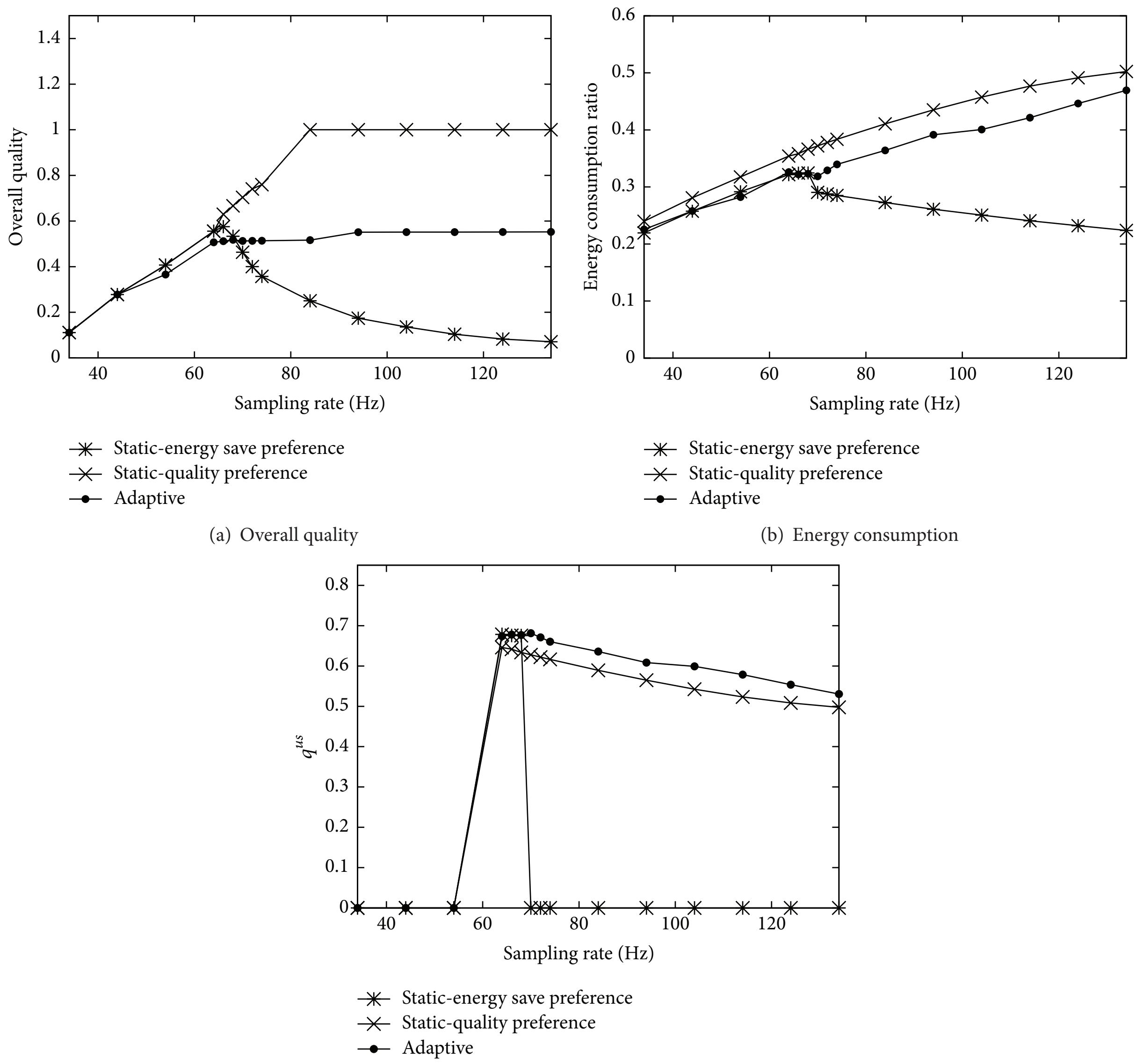
Figures 4(a) and 4(b) show the relationship of quality and energy consumption with sampling rate, respectively, as well as a comparison of adaptive and static sleep techniques. Here, we have assumed a quality requirement of 0.5. The measured SNR for a corresponding sampling rate is obtained from [14]. From Figure 4(a), we can see that at smaller rates (e.g., 30 Hz) the quality obtained is low, because the measurement SNR [14] at these rates are lower than the expected (27 dB) value. As the sampling rate increases, the quality starts improving, but energy consumption also increases, as shown in Figure 4(b).

Further, when the rates are higher than 60 Hz, the sleep scheduling techniques have different impact on quality and energy. With a *static-energy save preference* technique, since the sleep interval assigned is large, it achieves better energy saving, but it causes buffer overflow causing quality degradation. Once buffer overflow occurs, approximately same number of packets are transmitted by a node irrespective of its sampling rate. Hence, the energy consumption saturates, as shown in Figure 4(b). At the same time, quality degrades below the expectation, causing $\delta(q^r, q^e) = 1$ in our QoE metric, and hence q^{us} in Figure 4(c) plummets to 0. A *static-quality preference* technique, which favors quality, chooses a small sleep interval. Hence, we observe that the quality reaches an optimum, but the energy also increases, reducing the effective energy efficiency (Figure 4(c)). However, our proposed adaptive mechanism is able to increase energy saving while obtaining favorable quality, because it adapts to a node's sleep interval based on its sampling rate and an expected quality, thus improving the QoE (Figure 4(c)). Hence, we can conclude that although sleep scheduling is beneficial to conserving energy, the choice of a node's sleep interval as a function of its sampling rate will help improve application QoE.

5.2. Effect of Application Deadline on Effective Energy Efficiency. Figure 5 presents the relationship between sensor sleep scheduling and application deadline and their impact on a user application's quality-of-experience. Assigning a sleep-wake cycle with respect to an application's deadline is necessary to ensure that the delay in information reception at the application-end is within its acceptable range. To design such a schedule, we assume that the application specifies its delay tolerance range in terms of expected quality (q^e). Based on this application-specific q^e, our adaptive sleep mechanism determines the sleep interval of nodes such that the network provides assured quality while allowing higher energy saving.

Figure 5 shows the impact of an application's deadline on quality-of-information provided by the sensor nodes (Figure 5(a)), the energy consumed by the nodes in providing that quality (Figure 5(b)), and application's QoE (Figure 5(c)). As shown, a *static-quality preference* sleep technique satisfies the quality expectation for different deadlines (Figure 5(a)), but at the cost of higher energy consumption (Figure 5(b)). This is due to the fact that the sleep duration chosen will be small enough to ensure there is minimal or no loss of packets due to either delay or buffer overflow. With a *static-energy save preference* technique, the energy consumption ratio is comparatively small, but the QoI obtained is below the expectation ($q^e = 0.5$), especially for smaller deadlines (e.g., deadline < 100 sec), making its QoE equal to 0. However, our adaptive sleep mechanism adapts based on the application deadline and expected quality ($q^e = 0.5$); thus it is able to meet the quality expectation as well as achieving energy saving. At smaller deadlines (e.g., deadline < 32 sec), we observe that the quality (Figure 5(a)) is higher than the minimum expectation. Even a slight increase in sleep beyond the current selection caused quality to fall below the minimum due to the strictness of deadline. Therefore, we demonstrate the importance of incorporating an application deadline in the design criteria of sleep scheduling as well as the effectiveness of an adaptive mechanism in improving energy efficiency while providing quality assurance.

5.3. Influence of Quality Expectation on Effective Energy Efficiency. In Figure 6, we present the relationship between an application's quality requirement and the QoE provided by our adaptive sleep scheduler. As mentioned before, improving user experience especially for safety-critical applications is very important. Therefore, any energy-saving mechanism is useful for such applications only if it satisfies an application's expectation. From Figure 6(a), we find that a smaller



(a) Overall quality

(b) Energy consumption

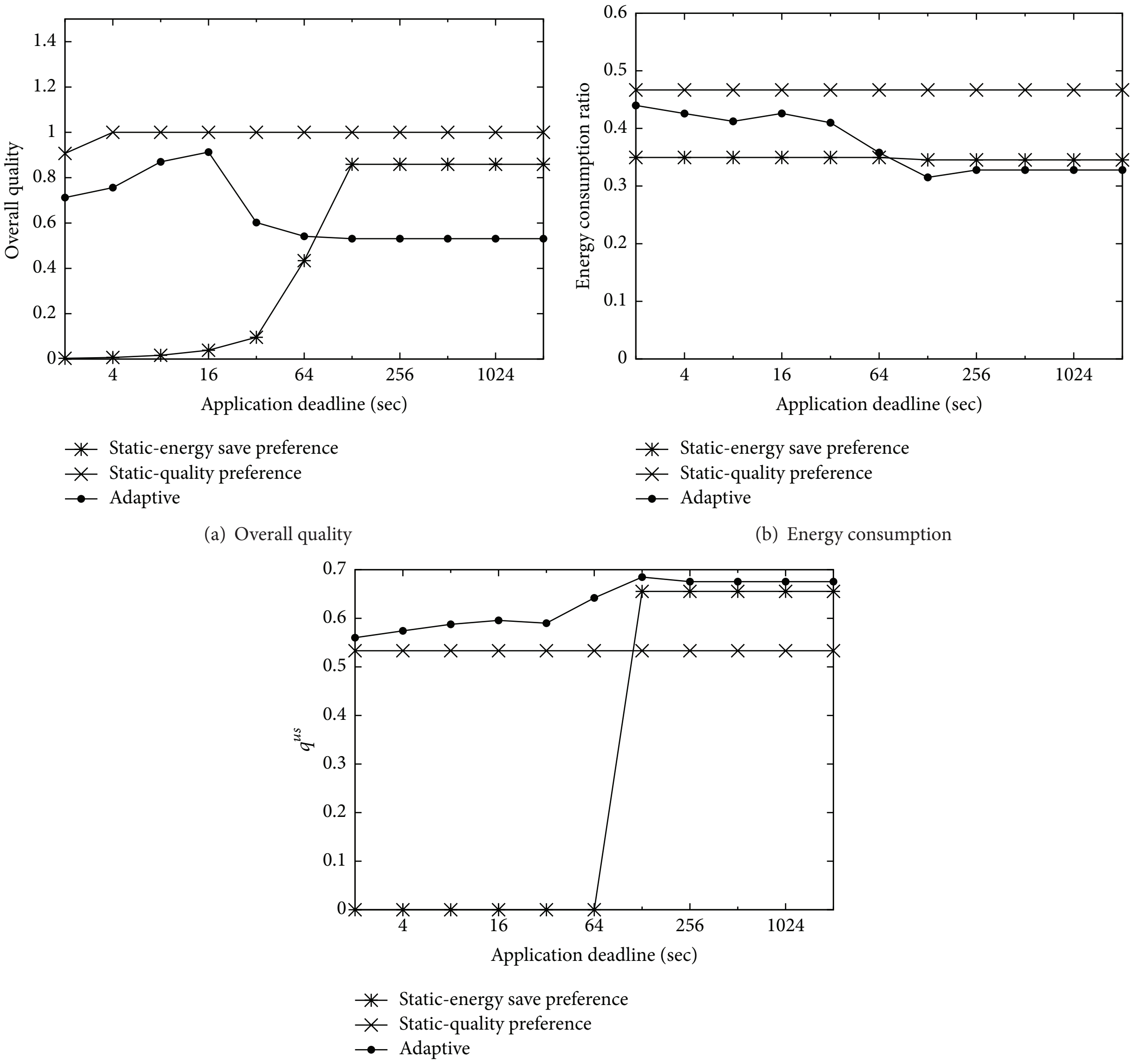
(c) QoE

Figure 4: Impact of sampling rate on overall quality and energy for varying sleep scheduling mechanisms. Here, expected quality (q^e) is set at 0.5 which is equivalent to loss tolerance of 1–3 percent.

sampling rate (e.g., 44 Hz) is sufficient to satisfy a low quality requirement (e.g., 0.25). However, meeting a higher quality expectation requires a higher sampling rate. Moreover, Figure 6(b) shows that, for each expected quality, the energy consumption increases as the sampling rate increases. Therefore, the QoE in Figure 6(c) rises for each q^e when the mechanism attains that expected value and decreases for the following rates. This is because with our mechanism the information quality saturates once q^e is satisfied, causing $\delta(q^r, q^e) = 1$ in (1). Any increase in energy consumption after this point will decrease QoE. Furthermore, the QoE of $q^e = 0.25$ is the highest, because a lower sensor rate that satisfies this quality expectation also provides the benefit of reduced energy consumption. Thus, this result displays the importance of considering application's requirements in determining appropriate values for sensor parameters such as sampling rate and sleep-wake intervals such that the overall QoE is improved.

6. Related Work

Sleep scheduling is a well-known mechanism used to enhance energy efficiency in wireless sensor networks. For instance, Zhen et al. [12] have proposed an on-demand sleep scheduling protocol to maximize energy saving and achieve better synchronization accuracy for reduced packet collisions. Aydin et al. [11] have addressed the problem of energy optimization by proposing an adaptive duty cycling algorithm which aims at reducing the switching energy consumption of sensor nodes. However, for quality-sensitive applications, achieving energy efficiency while satisfying the application's



(a) Overall quality

(b) Energy consumption

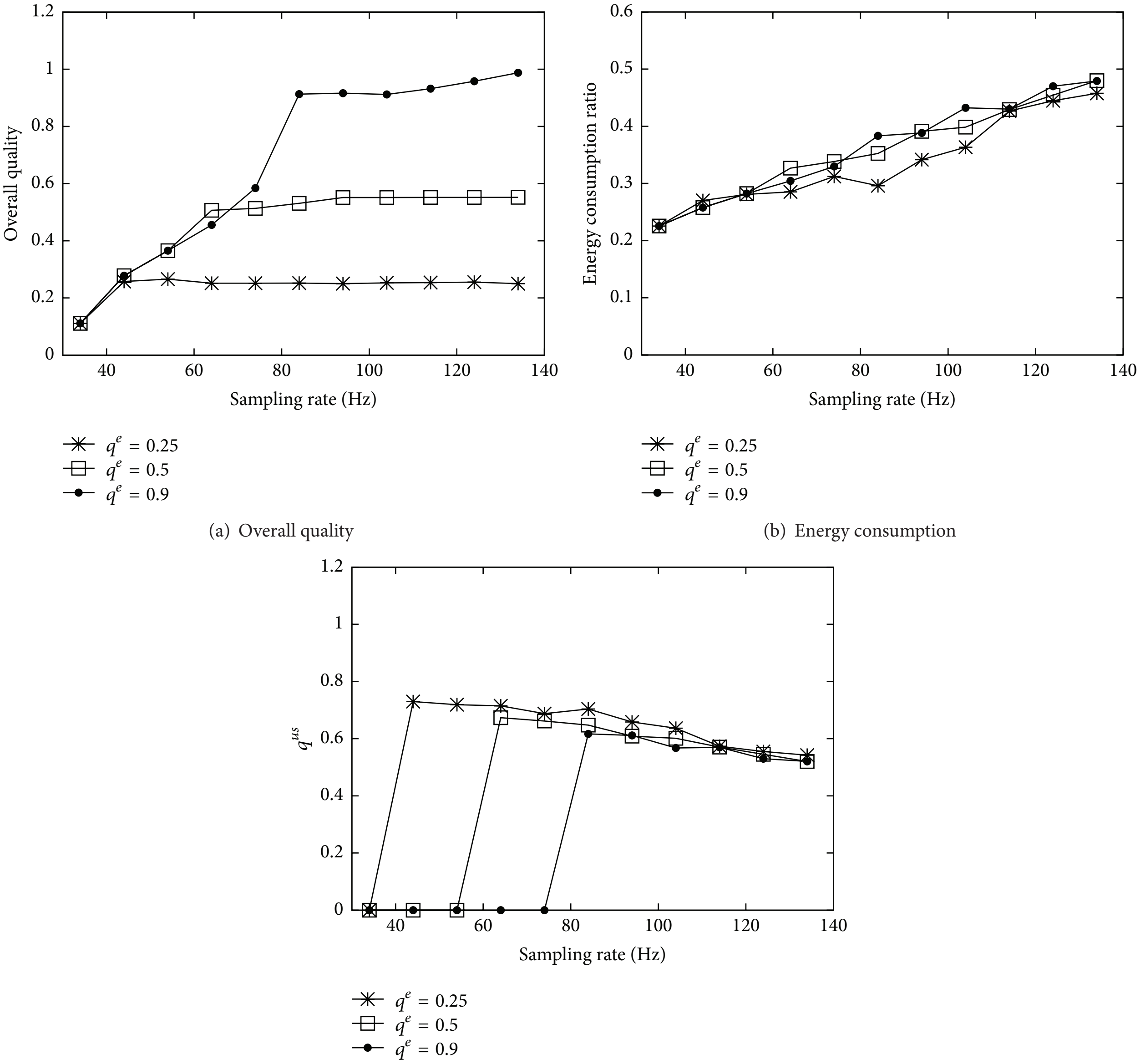
(c) QoE

Figure 5: Impact of application deadline on overall quality, energy for varying sleep scheduling mechanisms. Here, expected quality (q^e) is set at 0.5, which is equivalent to loss tolerance of 1–3 percent.

quality requirement is highly important. For these applications, obtaining cost reduction at the expense of information quality may affect its decision-making process, which can be detrimental depending on the application's criticality.

Most previous works quantify this as the energy-delay tradeoff problem. For example, Shi et al. [15] and Dao et al. [16] have proposed adaptive sleep schedule techniques that aim towards satisfying the end-to-end delay requirement of applications. However, we propose an adaptive sleep mechanism that aims at maximizing energy efficiency while providing assured quality. Spenza et al. [17] have designed a wake-up radio prototype and a cross-layer routing mechanism in order to improve the latency and network lifetime performance. In comparison, we aim at satisfying the application's quality requirement rather than aiming at a 100 percent packet delivery probability. In our proposed work, information quality is defined in terms of latency, buffer overflow, and channel interference. Our previous work [10] presented a novel energy efficiency model to reduce energy consumption and improve QoI. This work utilizes the QoI definition already proposed [10] and proposes an effective QoE metric to not only provide cooptimization, but also ensure that the cooptimization achieved is always meeting the user's QoI requirement as well as improving its energy saving.

In this paper, we propose an adaptive sleep scheduling mechanism that decides on the sleep schedule of sensor nodes based on our proposed QoE metric. The adaptive mechanism proposed here is similar to the idea presented by Liu et al. for VoIP applications over WLAN [18]. The major difference of our work is that we propose an adaptive sleep scheduler for



(a) Overall quality

(b) Energy consumption

(c) QoE

Figure 6: Impact of an application's quality expectation on effective energy efficiency achieved by our proposed sleep scheduling mechanism.

a resource-constrained (e.g., battery) network whereas [18] focuses on networks that are not limited in their resources. Therefore, though the idea is similar, the networks being addressed are different, and hence the solutions proposed vary. Finally, to validate the effectiveness of the adaptive mechanism, we performed OMNET++ based simulations by comparing our mechanism with a generalized static sleep mechanism.

7. Conclusion

This paper addresses the dual problem of energy efficiency enhancement and information quality assurance, in time-sensitive applications, in a paradigm of quality-of-experience. For this, an adaptive sleep-wake algorithm is presented and evaluated. Our results demonstrate that an effective scheduling of sensor sleep-wake cycle can benefit an application's QoE, characterized as achieving maximum energy efficiency and assured quality. We believe our QoE metric is a first step towards quantifying, designing, and evaluating different sensor networks to improve user experience.

Competing Interests

The authors declare that they have no competing interests.

References

[1] L. Rizzon, M. Rossi, R. Passerone, and D. Brunelli, "Wireless sensor networks for environmental monitoring powered by

microprocessors heat dissipation," in *Proceedings of the 1st ACM International Workshop on Energy Neutral Sensing Systems*, pp. 1–6, Rome, Italy, November 2013.

[2] M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. Wu, "Sensor placement with multiple objectives for structural health monitoring," *ACM Transactions on Sensor Networks*, vol. 10, no. 4, article 68, 45 pages, 2014.

[3] M. Mathew, N. Weng, and L. J. Vespa, "Quality-of-information modeling and adapting for delay-sensitive sensor network applications," in *Proceedings of the IEEE 31st International Performance Computing and Communications Conference (IPCCC '12)*, pp. 471–477, Austin, Tex, USA, December 2012.

[4] M. Z. Hasan and T.-C. Wan, "Optimized quality of service for real-time wireless sensor networks using a partitioning multipath routing approach," *Journal of Computer Networks and Communications*, vol. 2013, Article ID 497157, 18 pages, 2013.

[5] C. Bisdikian, L. M. Kaplan, and M. B. Srivastava, "On the quality and value of information in sensor networks," *ACM Transactions on Sensor Networks*, vol. 9, no. 4, article 48, 26 pages, 2013.

[6] C.-T. Cheng, H. Leung, and P. Maupin, "A delay-aware network structure for wireless sensor networks with in-network data fusion," *IEEE Sensors Journal*, vol. 13, no. 5, pp. 1622–1631, 2013.

[7] D. Rosário, Z. Zhao, T. Braun, and E. Cerqueira, "A cross-layer QoE-based approach for event-based multi-tier wireless multimedia sensor networks," *International Journal of Adaptive, Resilient and Autonomic Systems*, vol. 5, no. 1, pp. 1–18, 2014.

[8] L. Shi, W.-Z. Song, M. Xu, Q. Xiao, J. M. Lees, and G. Xing, "Imaging seismic tomography in sensor network," in *Proceedings of the 10th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '13)*, pp. 327–335, New Orleans, La, USA, June 2013.

[9] Y. Wang and D. Wang, "Energy-efficient node selection for target tracking in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 830950, 6 pages, 2013.

[10] M. Mathew and N. Weng, "Quality of information and energy efficiency optimization for sensor networks via adaptive sensing and transmitting," *IEEE Sensors Journal*, vol. 14, no. 2, pp. 341–348, 2014.

[11] N. Aydin, M. Karaca, and O. Ercetin, "Scheduling and power control for energy-optimality of low duty cycled sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 432978, 11 pages, 2015.

[12] C. Zhen, W. Liu, Y. Liu, and A. Yan, "Energy-efficient sleep/wake scheduling for acoustic localization wireless sensor network node," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 970524, 14 pages, 2014.

[13] I. Lazaridis, Q. Han, X. Yu et al., "QUASAR: quality aware sensing architecture," *SIGMOD Record*, vol. 33, no. 1, pp. 26–31, 2004.

[14] N. Weng, I.-H. Li, and L. Vespa, "Information quality model and optimization for 802.15.4-based wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1773–1783, 2011.

[15] P. Shi, Y. Wang, K. Li, and A. T. S. Chan, "Cross-layer adaptive end-to-end delay control for asynchronous duty-cycle wireless sensor networks," in *Pervasive Computing and the Networked World*, Q. Zu, M. Vargas-Vera, and B. Hu, Eds., vol. 8351 of *Lecture Notes in Computer Science*, pp. 520–531, 2014.

[16] T. Dao, S. Yoon, and J. Kim, "A deadline-aware scheduling and forwarding scheme in wireless sensor networks," *Sensors*, vol. 16, no. 1, article 59, 17 pages, 2016.

[17] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and C. Petrioli, "Beyond-duty cycling: wake-up radio with selective awakenings for longlived wireless sensing systems," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '15)*, pp. 522–530, April-May 2015.

[18] L. Liu, X. Cao, Y. Cheng, and Z. Niu, "Energy-efficient sleep scheduling for delay-constrained applications over WLANs," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2048–2058, 2014.

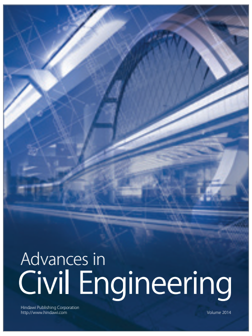
Journal of
Engineering

The Scientific World Journal

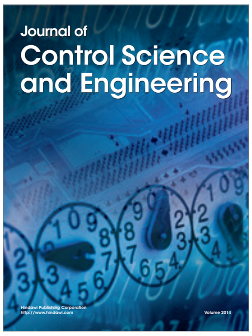
International Journal of
Rotating Machinery

Journal of
Sensors

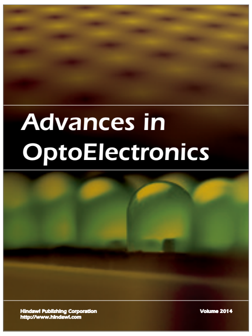
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering

Journal of
Robotics

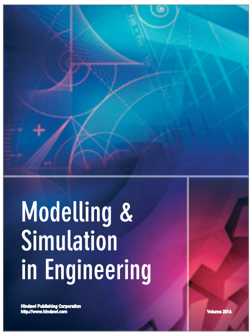

Journal of
Control Science and Engineering

Journal of
Electrical and Computer Engineering

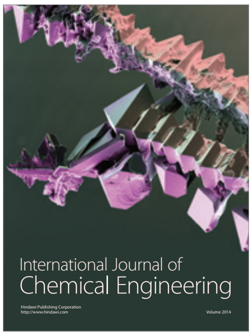

Advances in
OptoElectronics

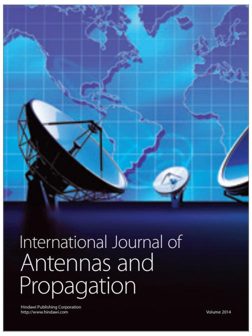
VLSI Design

International Journal of
Navigation and Observation

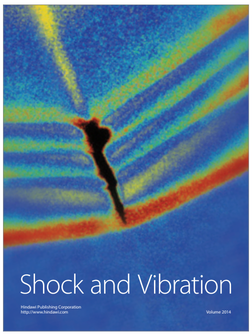

Modelling &
Simulation
in Engineering

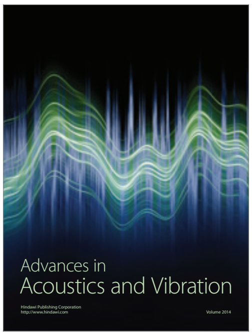
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration