

Optimal Sleep/Wake Scheduling for Time-Synchronized Sensor Networks with QoS Guarantees

Yan Wu, Sonia Fahmy, Ness B. Shroff

Abstract— We study sleep/wake scheduling for low-duty cycle sensor networks. Our work explicitly considers the effect of synchronization error. We focus on a widely used synchronization scheme and show that its synchronization error is non-negligible, and using a conservative guard time is energy wasteful. We formulate an optimization problem that aims to set the capture probability threshold for messages from each individual node such that the expected energy consumption is minimized, and the collective Quality of Service (QoS) over the nodes is guaranteed. The problem is non-convex. Nonetheless, we are able to obtain a solution with energy consumption that is provably at most 37% larger than the optimal solution. Simulations demonstrate the efficacy of our solution.

I. INTRODUCTION

Continuous monitoring systems constitute an important class of sensor network applications, where a large number of sensor nodes monitor the environment and *periodically* report to a single (or a few) base station(s). This application class includes many practical sensor network applications such as habitat monitoring [1], [2], civil structure monitoring [3], and factory maintenance [4].

To manage the large number of sensor nodes in such applications, a scalable method is to periodically group sensors within a geographical region into a cluster. The sensors can be managed locally by a cluster head (CH) – a node elected to coordinate the nodes within the cluster and to be responsible for communication between the cluster and the base station or other cluster heads. Clustering provides a convenient framework for resource management, data fusion, and local decision making [5]–[7]. One problem with clustering is that the cluster head is heavily utilized for both intra-cluster coordination and inter-cluster communications. Therefore, the cluster head will quickly deplete its energy. To address this concern, periodic re-clustering is performed to distribute the energy consumption among sensor nodes.

Sleep/wake scheduling has also been proposed to reduce energy consumption in sensor networks. The basic idea is to put the radio to sleep during idle times and wake it up right

before message transmission/reception. This requires fine-grained synchronization between the sender and the receiver, so that they can wake up at the same time to communicate. Prior work on sleep/wake scheduling assumes that the underlying synchronization protocol can provide nearly perfect (e.g., micro-second level) synchronization, or assumes an upper bound on the clock disagreement, and uses it as a guard time to compensate for the synchronization error. The awake period is lengthened by the guard time to combat synchronization errors. In practice, due to non-deterministic errors in time synchronization, as time progresses, clock disagreement becomes increasingly significant. Periodic re-synchronization can prevent the clocks from drifting, but for low duty cycle sensor networks, frequent re-synchronization would consume a significant amount of energy compared to communication/sensing. Using an upper bound on the clock disagreement as guard time will also significantly waste energy, since the synchronization error is non-deterministic.

In this work, we study the sleep/wake scheduling problem in clustered low duty cycle sensor networks. The nodes in the cluster are assumed to continuously monitor their environment and *periodically* report to the cluster head. Because the traffic is highly regular and the load is very low, the cluster head can go to sleep when no activity is present, and only wake up intermittently to send and receive messages. The following question hence becomes critical: *When should the cluster head wake up and how long should it stay active?*

With perfect synchronization, the cluster head and the cluster members simply agree upon a time and wake up simultaneously. We investigate a widely used synchronization scheme, proposed in the well-known Reference Broadcast Synchronization protocol [8]. We find that this scheme, although it achieves precise synchronization *immediately after* the exchange of synchronization messages, *has non-negligible clock disagreement as time progresses*. This, in fact, is true for most practical synchronization schemes, i.e., due to non-deterministic factors, the synchronization error will grow with time until the next exchange of synchronization messages. We conclude that the design of an effective sleep/wake scheduling algorithm must take into account the impact of synchronization error, and study the optimal sleep/wake scheduling scheme with consideration of the synchronization error. Our work includes two parts. In the first part, we show that there is an inherent tradeoff between energy consumption and message delivery performance (defined as the message capture probability in this work). In order to reduce energy consumption

This research has been sponsored in part by NSF grants 0238294 and 0207728, ARO MURI grant W911NF-07-10376, an Indiana 21st century grant, and a Tellabs foundation fellowship. Yan Wu and Sonia Fahmy are with the Department of Computer Science, Purdue University. E-mail: {wu26, fahmy}@cs.purdue.edu. Ness B. Shroff is with the Department of Electrical and Computer Engineering, The Ohio State University. E-mail: shroff@ece.osu.edu.

–A short version of this paper appeared in Proc. of IWQoS 2006.

but still guarantee high message delivery performance, we formulate an optimization problem to minimize the expected energy consumption, with the constraint that the message capture probability should be no less than a *threshold*. In the first part, we assume the threshold is *already given*. We find the problem to be non-convex, and cannot be directly solved by conventional convex optimization techniques. By investigating the unique structure of the problem, we transform the non-convex problem into a convex equivalent, and solve it using an efficient search method.

We then remove the assumption that the capture probability threshold is *given*, and instead determine a threshold to meet the Quality of Service (QoS) requirements of the application. In continuous monitoring systems, member nodes periodically report to the CH. Each message represents a certain amount of “information” about the environment. The CH uses the collected information to analyze interesting properties, e.g., the chemical contaminant in the area covered by this cluster. The accuracy of the analysis is determined by the total amount of information collected from all the member nodes, i.e., the collective information. Previous studies illustrate that information collected by nearby nodes is often correlated. This means that the CH does not need to receive all information from the member nodes. We thus define Quality of Service (QoS) to be that the CH collects a desired fraction of the total information. To achieve this QoS guarantee with minimum energy consumption, we must exploit the heterogeneity among the sensor nodes and favor more important ones. We thus formulate an optimization problem which aims to set the threshold for messages from each individual node such that the expected energy consumption is minimized, and yet the QoS is guaranteed. The problem turns out to be non-convex and hard to solve exactly. Therefore, we use approximation techniques to obtain a suboptimal solution that approximates the optimum.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the system model. Section IV discusses the optimal sleep/wake scheduling problem and presents the solution. Section V studies how to assign the threshold for messages from each individual node. Section VI concludes the paper.

II. BACKGROUND AND RELATED WORK

We first discuss time synchronization, then review previous work on sleep/wake scheduling.

A. Time Synchronization for Sensor Networks

Time synchronization has been studied in the context of wireless sensor networks [8]–[16]. Clock disagreement among sensor nodes is essentially due to two effects: *phase offset* and *clock skew*. Phase offset corresponds to clock disagreement between nodes at a given instant. Clock skew is because the crystal oscillators used on sensor nodes are imperfect, i.e., there is a difference between the expected frequency and the actual frequency. Further, the frequency may be time-varying due to environmental factors, including variations in

temperature and pressure [17]. Due to clock skew, clocks diverge over time.

Several synchronization protocols have been proposed to estimate the phase offset and clock skew. Elson et al. [8] proposed a receiver-receiver synchronization scheme called Reference-Broadcast Synchronization (RBS). In RBS, a node sends beacons to its neighbors using physical-layer broadcast. The recipients use the arrival time of the broadcast as a reference point to compare their times. RBS removes the non-determinism in the transmission time, channel access time, and propagation delay. The only non-determinism is in the packet reception time. To estimate the clock skew and phase offset, least square linear regression is used. Ganeriwal et al. [10] propose a sender-receiver synchronization approach called Timing-sync Protocol for Sensor Networks (TPSN). The TPSN approach time-stamps synchronization messages at the Medium Access Control (MAC) layer. It eliminates errors caused by access time and propagation delay via a two way message exchange. A shortcoming of TPSN is that it does not estimate the clock skew of the nodes. Maroti et al. [13] combine the MAC layer time-stamping of TPSN with clock skew estimation using linear regression, and demonstrate improved performance. In both RBS and TPSN, measurements show that the synchronization error follows a well-behaved *normal distribution with zero mean*. We will use this observation to model the error distribution in our work.

Both RBS and TPSN give high-precision synchronization, i.e., the clock disagreement *immediately after* the exchange of synchronization messages is on the order of several tens of microseconds. However, due to the estimation error in the clock skew, the clock disagreement becomes *more significant* as time progresses.

B. Sleep/Wake Scheduling for Sensor Networks

Sleep/wake scheduling for sensor networks has been extensively investigated [18]–[26]. The basic idea is to put the radio to sleep during idle times, and wake it up right before message transmission/reception. Existing sleep/wake scheduling schemes for wireless sensor networks can be synchronization-based, where nodes synchronize each other to coordinate their wake up schedules, or asynchronous/random which do not involve explicit synchronization. For continuous monitoring systems, synchronization-based sleep/wake scheduling schemes are often used because the traffic pattern is periodic. Fine-grained synchronization is required between the sender and the receiver, so that they can wake up at the same time to communicate. Prior work either assumes that the underlying synchronization protocol can provide nearly perfect (e.g., micro-second level) synchronization, or assumes an upper bound on the clock disagreement, and uses it as a guard time to compensate for the synchronization error. However, as pointed out in Section II-A, existing synchronization protocols like RBS or TPSN achieve micro-second level synchronization at the time instant *immediately following* the exchange of synchronization messages. Due to estimation errors in the clock skew, the clocks will gradually drift as time progresses, until the next exchange of synchronization messages. To see

how significant the clock disagreement can be, consider two nodes that have agreed to rendezvous on the radio channel once every 100 seconds to exchange a 20-byte message. Using a 19.2 kbps radio such as RF Monolithics [27], 20 bytes can be transmitted in about 8 ms. The radio must be awakened early to account for clock disagreement. Let the estimation error of the clock skew be 10 parts-per-million (ppm)¹, i.e., the clocks of the two nodes drift away from each other 10 μ s each second. After 100 seconds, the clocks will drift by 10 μ s \times 100 = 1 ms, which is non-negligible compared to the actual message transmission time. Ye and Heidemann [29] considered the effect of synchronization error in the design of a polling-based MAC protocol called Scheduled Channel Polling (SCP). In SCP, the receivers periodically poll the channel for network activity, and the sender uses a preamble to wake up the receiver before sending the actual message. To accommodate the clock disagreement they extend the preamble by a guard time, which is equal to the product of the maximum clock skew and the time elapsed since last synchronization. Using this worst case clock disagreement as the guard time can compensate for the synchronization error. However, energy efficiency can be further improved by exploiting the non-deterministic nature of the synchronization error.

III. SYSTEM MODEL

For our system model, we consider a cluster that has been constructed using an existing clustering protocol (e.g., [5]–[7]). The cluster consists of a single cluster head (CH) and M cluster member nodes n_1, n_2, \dots, n_M (Fig. 1). Time is divided into recurring *epochs* with constant duration T_e . As with many MAC protocols for sensor networks [19]–[21], each epoch begins with a synchronization interval, T_s , followed by a transmission interval (Fig. 1). During the synchronization interval, the cluster members synchronize with the CH, and no transmissions are allowed. During the transmission interval, each member continuously monitors its environment and sends one message to the CH every T seconds. Each transmission interval contains one or more rounds of transmissions, i.e., $T_e = T_s + NT$, $N \geq 1$. The transmissions from different members are equispaced, i.e., transmissions from node n_i and n_{i+1} are separated by $\frac{T}{M}$. Re-clustering of the network may occur at a lower frequency than synchronization, i.e., the time between re-clustering the network consists of one or more epochs.²

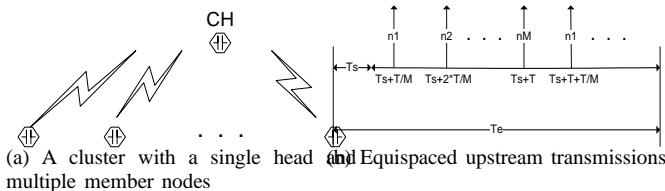


Fig. 1. System model

¹From the datasheet of Mica Motes [28], the clock skew with respect to the standard clock is up to 50 ppm, thus the relative clock skew between two sensor nodes can be 100 ppm in the worst case.

²We summarize all the symbols used in Table III in the appendix.

We make the following assumptions about our system:

(1) Communication pattern: In this work, we focus on intra-cluster communications. Another important question is: after a CH receives messages from its members, how can it transmit messages to the base station (possibly via other cluster heads), and how can it wake up to relay messages from other cluster heads? This can be achieved, for example, by further dividing each transmission interval into two subintervals. One subinterval is for intra-cluster communications, and the other is for inter-cluster communications when the CH is always active. In the remainder of the paper, we only focus on intra-cluster communications. Our future work plans include considering the problem of energy-efficient sleep/wake scheduling for inter-cluster communications.

We further assume that neighboring clusters use orthogonal frequency channels and do not interfere with each other. This assumption is reasonable since the data rate requirements of sensor networks are usually low, typically around 10–40 kbps. If we assume the radios operate in the ISM-900 bands (902–928 MHz), then we have more than a thousand frequency bands to choose from.

(2) Clock skew: Vig [17] discussed the behavior of general off-the-shelf crystal oscillators. Because of imprecision in the manufacturing process and aging effects, the frequency of a crystal oscillator may be different from its desirable value. The maximum clock skew is usually specified by the manufacturer and is no larger than 100 ppm. Besides manufacturing imprecision and aging, the frequency is also affected by environmental factors including variations in temperature, pressure, voltage, radiation, and magnetic fields. Among these environmental factors, temperature has the most significant effect. For general off-the-shelf crystal oscillators, when temperature significantly changes, the variation in the clock skew can be up to several tens of ppm, while the variation caused by other factors is far below 1 ppm. Observe, however, that temperature does not change dramatically within a few minutes in typical sensor environments. If the epoch duration T_e is chosen according to the temperature change properties of the environment, we can assume that the clock skew for each node is constant over each epoch. This is consistent with the observations in [16].

The crystal oscillator used by Mica Motes [28] is one type of off-the-shelf crystal oscillator, with similar characteristics to those discussed above. Specifically, its maximum clock skew can be up to 50 ppm.

(3) Radio hardware: For the transmitter circuit, we assume that the sender can precisely control when the message is sent out onto the channel using its *own* clock. This is consistent with the measurements in [10]. Therein, it is observed that non-determinism at the sender is negligible compared to non-determinism at the receiver, i.e., there are minor random effects at the sender.

For the receiver circuit, we assume that if there is an incoming message, the receiver circuit can detect the signal immediately. This is a close approximation of the real situation, since modern transceivers can detect the incoming signal within several microseconds [30]. We further assume that once the receiver circuit detects an incoming message, it can let the processor know, so that the processor will keep the radio active

until the reception is completed. This can be easily achieved using a 1-bit status register.

(4) Sleep/wake transition time: Research shows that with recent advances in hardware technology, the transition time between sleep and wake states can be reduced to a few clock cycles [31], [32]. Thus, we consider the transition time to be negligible.

(5) Collisions: We assume that the separation between transmissions from different members, namely $\frac{T}{M}$, is large enough so that the collision probability between transmissions from different members is negligible. This assumption is reasonable for low duty cycle sensor networks. Consider a large cluster of $M = 100$ members and each member transmits to the CH every $T = 60$ seconds. The separation is $\frac{T}{M} = 600$ ms, which is much larger than the message transmission time in sensor networks. In addition, the cluster nodes will be re-synchronized before the clock disagreement becomes large enough to cause significant collision probability.

(6) Energy expenditure: Measurements show that among all the sensor node components, the radio consumes the most significant amount of energy. In Section IV-B, we will show that the computational complexity of our scheduling algorithm is very low. Therefore, in this work, we only account for the energy consumption of the radio.

(7) Propagation delay: Finally, because the communication range for sensor nodes is typically < 100 meters, the propagation delay is below $1\mu s$. Hence, we consider the propagation delay to be negligible and assume it to be zero for simplicity.

A. Synchronization Algorithm

We adopt a *widely used* synchronization scheme, and study the sleep/wake scheduling problem under this scheme³. The scheme was first proposed in RBS [8], and was later adopted by several protocols and system implementations [11]–[15]. The scheme includes two steps: (1) Exchange synchronization messages to obtain multiple pairs of corresponding time instants; (2) Use linear regression to estimate the clock skew and phase offset.

Either a receiver-receiver approach or a sender-receiver approach can be used in the synchronization protocol. For the purpose of intra-cluster communication, the members only need to synchronize locally with the CH. Thus, at the start of each epoch j , each cluster member n_i will exchange several synchronization messages with the CH and obtain N_s pairs of corresponding time instants $(C(j, k), t_i(j, k))$, $k = 1 \dots N_s$, where $C(j, k)$, $t_i(j, k)$ denote the k^{th} time instant of the CH and of node n_i in epoch j respectively.

Under the assumption that the clock skew of each node does not change over the epoch, the clock time t_i of node n_i during an epoch is a linear function of the CH clock time C , i.e., $t_i(C) = a_i(j)C + b_i(j)$, where $a_i(j)$, $b_i(j)$ denote the relative clock skew and phase offset (respectively) between n_i and CH in epoch j .

Due to the non-determinism in the synchronization protocols, the time correspondence obtained via exchange of

synchronization messages is not exactly accurate and contains an error, i.e.,

$$t_i(j, k) = a_i(j)C(j, k) + b_i(j) + e_i(j, k), \quad (1)$$

where $e_i(j, k)$ is the random error caused by non-determinism in the system. In [8], measurements show that $e_i(j, k)$ follows a normal distribution with zero mean $N(0, \sigma_0^2)$, and σ_0 is on the order of several tens of microseconds. Specifically, the chi-square test shows a 99.8% confidence, which strongly indicates the validity of this model.

In each epoch j , pairs $(C(j, k), t_i(j, k))$, $k = 1 \dots N_s$ are obtained during the synchronization interval. Then, linear regression is performed on these N_s pairs to obtain estimates of $a_i(j)$, $b_i(j)$, denoted by $\hat{a}_i(j)$, $\hat{b}_i(j)$.

IV. PART I: OPTIMAL SLEEP/WAKE SCHEDULING WITH A GIVEN THRESHOLD

A. Problem Definition

Suppose that during epoch j , node n_i has a packet (message) p to send at CH clock time τ_p , where $jT_e \leq \tau_p \leq (j+1)T_e$. The node first translates τ_p into its own time using the estimates $(\hat{a}_i(j), \hat{b}_i(j))$, i.e., $\hat{t}_i(\tau_p) = \hat{a}_i(j)\tau_p + \hat{b}_i(j)$. Then, it sends out the message at $\hat{t}_i(\tau_p)$ according to its own clock.

The CH clock time corresponding to $\hat{t}_i(\tau_p)$ is:

$$\tau'_p = \frac{\hat{t}_i(\tau_p) - b_i(j)}{a_i(j)} = \tau_p + \frac{(\hat{a}_i(j) - a_i(j))\tau_p + (\hat{b}_i(j) - b_i(j))}{a_i(j)}. \quad (2)$$

If the estimation is exact, i.e., $(\hat{a}_i(j), \hat{b}_i(j)) = (a_i(j), b_i(j))$, then from Equation (2), $\tau'_p = \tau_p$, i.e., n_i will transmit precisely at τ_p . Under our assumption of negligible propagation delay, τ_p is equal to the time at which p *should* arrive if the synchronization is perfect, i.e., the *scheduled arrival time*, while τ'_p is equal to the time that p *actually* arrives, i.e., the *actual arrival time*. Hence, $\tau'_p = \tau_p$ means that the actual arrival time is exactly the same as the scheduled arrival time. If this is true, the CH simply wakes up at τ_p to receive the message.

However, as given in Equation (1), random errors exist in the measurements. Therefore, $(\hat{a}_i(j), \hat{b}_i(j))$ is also random. As a result, the actual arrival time τ'_p will deviate from the scheduled arrival time τ_p . To compensate for this random deviation and to “capture” (receive) the message, the CH needs to wake up earlier than τ_p and stay active for some time (Fig. 2). This leads to the following question: *When should the cluster head wake up and how long should it stay active?*

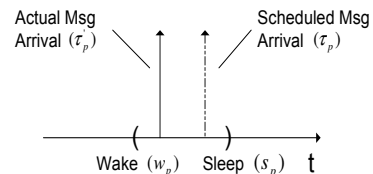


Fig. 2. Wake up interval to capture the message

Intuitively, if the CH wakes up much earlier than τ_p and stays active for a long time, it has a high probability

³We select this scheme for illustration purposes, but our sleep/wake scheduling solution works with most synchronization schemes.

of ‘‘capturing’’ the message; however, waking up early and staying active for a long time wastes energy. In order to reduce energy consumption and yet guarantee high message delivery performance, we formulate the following optimal sleep/wake scheduling problem which attempts to minimize the expected energy consumption with constraints on the ‘‘capture’’ probability.

Let p be a message from n_i to arrive during epoch j , i.e., scheduled arrival time $\tau_p \in (jT_e, jT_e + T_e)$. Let τ'_p be the actual arrival time at which p arrives at the CH, as defined in Equation (2). To capture p , the CH wakes up at w_p . If the message does not arrive until s_p , the CH goes back to sleep at s_p ; if the message arrives between w_p and s_p , the CH remains active until the message is received, which may be earlier or later than s_p depending upon the actual arrival time and message length. Our goal is to determine w_p and s_p to minimize the expected energy consumption as described by the following optimization problem:

$$(A) \text{ Min } E = (s_p - w_p)\alpha_I \text{Prob}\{\tau'_p \notin (w_p, s_p)\} + \int_{w_p}^{s_p} \{(x - w_p)\alpha_I + \frac{L_p}{R}\alpha_r\} f_{\tau'_p}(x) dx$$

such that $\text{Pr}\{\tau'_p \in (w_p, s_p)\} \geq th$,

where:

- α_I and α_r are the idle power and the receiving power, respectively;
- L_p is the length of the message;
- R is the data rate;
- $f_{\tau'_p}(\cdot)$ is the probability density function (pdf) of τ'_p ;
- th is the threshold on the capture probability, $0 < th < 1$.

Its value should be decided by the QoS requirements of the application. In this section, we assume that the value of th is *already given* and is the same for messages from different cluster members, i.e., all members are treated ‘‘uniformly.’’ Later in Section V, we will study how to set the value of th to meet the QoS requirements of the application.

In problem (A), the first term corresponds to the expected energy consumption when the message is missed, i.e., $\tau'_p \notin (w_p, s_p)$. In this case, the CH stays active during the time interval (w_p, s_p) and consumes $(s_p - w_p)\alpha_I$ amount of idle energy. The second term corresponds to the expected energy consumption when the message is received. Suppose the message arrives at $x \in (w_p, s_p)$, then, in addition to the reception energy, the CH will consume $(x - w_p)\alpha_I$ amount of idle energy, i.e., the energy needed to remain idle for (w_p, x) .

B. Solution

We first compute the PDF $f_{\tau'_p}(x)$, transform the problem, and then solve the equivalent formulation.

By linear regression analysis [33], we find that τ'_p is normally distributed and

$$E(\tau'_p) = \tau_p, \quad (3)$$

$$\sigma_p^2 \equiv \text{VAR}(\tau'_p) = \frac{\sigma_0^2}{a_{ij}^2} \left[\frac{1}{N_s} + \frac{1}{N_s} \frac{(\tau_p - \overline{C(j)})^2}{\overline{C^2(j)} - (\overline{C(j)})^2} \right].$$

where $\overline{C(j)} = \frac{\sum_{k=1}^{N_s} C(j,k)}{N_s}$, $\overline{C^2(j)} = \frac{\sum_{k=1}^{N_s} C^2(j,k)}{N_s}$.

Substituting Equation (3) into problem (A), and letting $\hat{\tau} = \frac{\tau'_p - \tau_p}{\sigma_p}$, then $w = \frac{w_p - \tau_p}{\sigma_p}$, $s = \frac{s_p - \tau_p}{\sigma_p}$, i.e., $\hat{\tau}, (w, s)$ are the normalized arrival time and normalized wake up interval respectively. With simple algebraic operations, problem (A) becomes:

$$(A1) \text{ Min } F(w, s) = (s - w)\sigma_p\alpha_I - [Q(w) - Q(s)]s\sigma_p\alpha_I + [g(w) - g(s)]\sigma_p\alpha_I + [Q(w) - Q(s)]\frac{L_p}{R}\alpha_r,$$

such that $Q(w) - Q(s) \geq th$,

where $g(x)$ is the probability density function for the standard normal distribution, and $Q(x)$ is the complementary cumulative distribution function.

The main difficulty in solving (A1) is that the problem is not a convex optimization problem (this can be easily shown by computing the Hessian matrix). Due to the non-convexity, we cannot use conventional convex optimization techniques [34] to find the optimal solution. Hence, we look into the structure of problem (A1) and show that it has certain unique properties that enable us to transform it into a convex equivalent, and solve the equivalent using an efficient search method.

We start by showing the following lemma.

Lemma 1: $\frac{\partial F}{\partial s} > 0$

Proof: We compute $\frac{\partial F}{\partial s}$ as follows.

$$\frac{\partial F}{\partial s} = [1 - Q(w) + Q(s)]\sigma_p\alpha_I + Q'(s)(s - w)\sigma_p\alpha_I + Q'(s)w\sigma_p\alpha_I + \frac{1}{\sqrt{2\pi}}se^{-\frac{s^2}{2}}\sigma_p\alpha_I + (-Q'(s))\frac{L_p}{R}\alpha_r.$$

Since $Q'(s) = -\frac{1}{\sqrt{2\pi}}e^{-\frac{s^2}{2}}$, put it in and we get

$$\frac{\partial F}{\partial s} = [1 - Q(w) + Q(s)]\sigma_p\alpha_I + \frac{1}{\sqrt{2\pi}}e^{-\frac{s^2}{2}}\frac{L_p}{R}\alpha_r.$$

Since $\forall x \in R, 0 \leq Q(x) \leq 1$, therefore $1 - Q(w) + Q(s) \geq 0$. Consequently,

$$\frac{\partial F}{\partial s} = [1 - Q(w) + Q(s)]\sigma_p\alpha_I + \frac{1}{\sqrt{2\pi}}e^{-\frac{s^2}{2}}\frac{L_p}{R}\alpha_r > 0. \quad \blacksquare$$

Here is an intuitive explanation of $\frac{\partial F}{\partial s} > 0$. As in (B), we write

$$F(w, s) = \sigma_p\alpha_I(s - w)\text{Prob}\{\hat{\tau} \notin (w, s)\} + \int_w^s (x - w)\sigma_p\alpha_I g(x) dx + \int_w^s \frac{L_p}{R}\alpha_r g(x) dx.$$

We note that the first two terms correspond to the expected idle energy consumption, while the third term corresponds to the expected energy used to receive the message. Suppose the normalized wake up interval is changed from (w, s) to $(w, s + \Delta)$, we observe that:

- The expected energy for receiving the message increases because the capture probability is larger;
- The change in the idle energy consumption is illustrated in Fig. 3. In the figure, t_1, t_2 , and t_3 are three possible message arrivals, where $t_1 \in [w, s]$, $t_2 \in (s, s + \Delta)$, $t_3 \notin [w, s + \Delta)$. I'_i, I_i $i = 1, 2, 3$ are the idle time for the message arrival at t_i before and after s is increased to $s + \Delta$, respectively.

– If the message arrival is in $[w, s]$, e.g., t_1 , the idle energy consumption does not change;

- If the message arrival is in $(s, s + \Delta)$, e.g., t_2 , the idle energy increases;
- If the message arrival is at another time, e.g., t_3 , the idle energy increases.

Therefore, as the normalized wake up interval changes from (w, s) to $(w, s + \Delta)$, the idle energy does not decrease, while the expected receiving energy always increases. This explains why the total energy consumption increases with s .

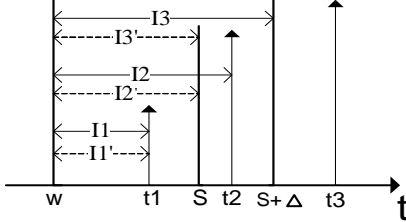


Fig. 3. Changes in the idle time when s increases

The next proposition shows that the optimal solution always appears at the boundary of the region $Q(w) - Q(s) \geq th$.

Proposition 1: Let (w^*, s^*) be the optimal solution to (B), then $Q(w^*) - Q(s^*) = th$.

Proof: We prove this by contradiction. Suppose $Q(w^*) - Q(s^*) > th$.

Because $Q(x)$ is continuous, $\exists r_1 > 0$ s. t.

$$Q(w^*) - Q(s) \geq th, \forall s^* - r_1 \leq s \leq s^* + r_1.$$

Meanwhile, we have

$$F(w^*, s^* - \Delta) - F(w^*, s^*) = -\frac{\partial}{\partial s} F(w^*, s^*) \Delta + O(\Delta^2).$$

As shown in Lemma 1, $\frac{\partial}{\partial s} F(w^*, s^*) > 0$, so $\exists r_2 > 0$ s. t.

$$F(w^*, s^* - \Delta) - F(w^*, s^*) < 0, \forall 0 < \Delta \leq r_2.$$

Pick $r = \min(r_1, r_2)$, then it satisfies $Q(w^*) - Q(s^* - r) \geq th$ and $F(w^*, s^* - r) - F(w^*, s^*) < 0$, which means that $(w^*, s^* - r)$ is a feasible point and $F(w^*, s^* - r)$ is lower than the minimum. This is contradictory to the fact that (w^*, s^*) is the optimal solution. ■

The physical meaning of $Q(w^*) - Q(s^*) = th$ is that under the optimal scheduling policy, the capture probability is always equal to the threshold th . This can easily be understood from Lemma 1. If $Q(w) - Q(s) > th$, then we reduce s by a small amount (go to sleep earlier by) Δ . From Lemma 1, the total energy consumption decreases, yet the capture probability still exceeds the threshold. Thus, $(w, s - \Delta)$ is a better solution than (w, s) . Hence, the optimal solution must satisfy $Q(w^*) - Q(s^*) = th$.

Substituting $Q(w^*) - Q(s^*) = th$, formulation (A1) becomes:

$$(A2) \text{ Min } F(w, s) = [(1 - th)s - w + g(w) - g(s)]\sigma_p\alpha_I + th\frac{L_p}{R}\alpha_r, \\ \text{such that } Q(w) - Q(s) = th.$$

We further simplify the formulation as follows. First, because $th\frac{L_p}{R}\alpha_r$ does not depend on w and s , we remove it from $F(w, s)$. Second, all the remaining terms of $F(w, s)$

have $\sigma_p\alpha_I$, so we can extract $\sigma_p\alpha_I$. Finally, we express s as a function of w , $s(w) = Q^{-1}(Q(w) - th)$, $w < Q^{-1}(th)$. Now, the formulation becomes:

$$(A3) \text{ Min } G(w) = (1 - th)s(w) - w + g(w) - g(s(w)), \\ \text{such that } s(w) = Q^{-1}(Q(w) - th), w < Q^{-1}(th).$$

So far, we have transformed the original formulation (A) into an equivalent formulation (A3). We notice that from (A2) and (A3), the minimum expected energy to receive the message can be expressed as

$$\sigma_p\alpha_I H(th) + \frac{L_p}{R}\alpha_r th, \quad (4)$$

where

$$H(th) = \min\{G(w) : s(w) = Q^{-1}(Q(w) - th), \\ w < Q^{-1}(th)\} \quad (5)$$

is the minimum value of the objective function in (A3). Equations (4) and (5) will be used later in Section V.

Next, we solve (A3). We first show that $G(w)$ is a convex function of w .

Proposition 2: $G''(w) > 0$.

We give the proof in the appendix. Since $G(w)$ is convex, and the region $w \in (-\infty, Q^{-1}(th))$ is a convex region, then the local minimum is in fact a global minimum. The next proposition gives the position of the global minimum.

Proposition 3: Let w_0 be the global minimum, $w_l = Q^{-1}(\frac{1+th}{2})$, $w_u = \min(0, Q^{-1}(th))$, then $w_0 \in (w_l, w_u)$, and is the unique minimum on this interval.

We give the proof in our technical report [35]. Since w_0 is the unique minimum on (w_l, w_u) , we can use the Golden Search method to find w_0 [36]. The Golden Search method has logarithmic complexity of $O(\log(\frac{1}{\delta}))$, where δ is the required precision. Hence, it can be efficiently implemented.

C. Example Implementation

We now describe an example implementation of our approach.

Cluster Initialization. After the clusters are established, the cluster head (CH) broadcasts the epoch duration T_e , synchronization interval T_s , and message frequency T to the cluster members, and lets the members know when they should transmit (according to the CH clock). In the following discussion, we will assume that the parameters and the transmission schedules for the members will not change in the system. In cases when these need to be changed, the CH simply makes the change and informs the members.

Synchronization. As we indicated in Section III-A, the synchronization scheme can utilize either a receiver-receiver approach (RBS) or a sender-receiver approach (TPSN). Our example implementation uses RBS. We first review how RBS works. In RBS, when two nodes A and B want to synchronize with each other, they need a separate beacon node. The beacon node broadcasts a beacon, which is received at T_1 and T_2 by A and B respectively. Specifically, let the relationship between node A's clock and node B's clock be $t_B = at_A + b$. Then, $T_2 = aT_1 + b + e$, where e is the non-deterministic factor, which follows a normal distribution $N(0, \sigma_0^2)$. Hence, one pair

of corresponding times (T_1, T_2) is obtained. Additional pairs can be obtained using multiple broadcast beacons.

To use RBS in the cluster, the CH selects a member as the beacon node. This member sends reference beacons using a sufficiently high power level⁴ for the beacons to be received by all other members and the CH. The cluster members then exchange the arrival times of the beacons with the CH and obtain multiple pairs of corresponding time instants. The cluster members will use these pairs to estimate $(a_i(j), b_i(j))$ as described in Section III-A. At this stage, all members will have synchronized with the CH except the beacon node. The CH then selects another member to send reference beacons, so that the original beacon node can synchronize with the CH.

To further conserve energy, the cluster members and the CH do not always stay active during the synchronization interval. Instead, they go to sleep if there are no beacons and wake up right before beacons arrive. Observe, however, that because of clock disagreement, the CH and the cluster members may not rendezvous with the beacon node. This leads to the loopback problem: the synchronization message itself cannot be successfully exchanged because of synchronization error. To solve this problem, the cluster members and the CH use a guard time to compensate for synchronization error. This guard time is chosen to be $3 \times$ message transmission time, while the clock disagreement is controlled such that it cannot go far beyond the message transmission time with high probability. Using this mechanism, in all our simulations, beacons and subsequent messages can be successfully communicated.

Determining the Wake up Schedule. To determine the wake up interval, the CH first computes (w^*, s^*) using the Golden Search method. The CH needs to do this computation only once. Next, the CH computes for each message p the value of σ_p using Equation (3). In Equation (3), σ_0 can be obtained from measurements which have already been taken, e.g., in RBS and TPSN. The difficulty is that we do not know $a_i(j)$. However, we can bound $a_i(j)$ in the following manner. According to [17], the maximum clock skew of most off-the-shelf crystal oscillators is no larger than 100 ppm (specifically for Mica Motes, the clock skew is no larger than 50 ppm and the bounds below still hold). Therefore, $\frac{1-10^{-4}}{1+10^{-4}} \leq a_i(j) \leq \frac{1+10^{-4}}{1-10^{-4}}$. Substituting into Equation (3), we have

$$0.9998^2 \leq \sigma_p^2 / [\sigma_0^2 (\frac{1}{N_s} + \frac{1}{N_s} \frac{(\tau_p - \overline{C(j)})^2}{\overline{C^2(j)} - \overline{C(j)}^2})] \leq 1.00021^2.$$

We choose $\sigma_p^2 = 1.00021^2 \sigma_0^2 [\frac{1}{N_s} + \frac{1}{N_s} \frac{(\tau_p - \overline{C(j)})^2}{\overline{C^2(j)} - \overline{C(j)}^2}]$. This value is no less than the actual σ_p , so the wake up interval will be larger than necessary and the capture probability will be slightly higher than th ; yet the wake up interval is no more than 0.04% larger than necessary, which causes little degradation in the energy consumption. After σ_p is obtained,

⁴We assume that each node has a fixed number of transmission power levels (as in Mica2 motes) and can transmit to the CH and all other cluster members using one of these power levels. This assumption is reasonable since in many clustering techniques, the transmission power level used by the members to communicate with the CH is much lower than the maximum. Therefore, a member node can increase the transmission power level to ensure its message can be received by the CH as well as other cluster members.

the CH computes for each message p the wake up interval $(w_p, s_p) = (\tau_p + \sigma_p w^*, \tau_p + \sigma_p s^*)$.

D. Simulations

We now verify the performance of our scheduling policy via simulations. We follow the implementation described in Section IV-C. We utilize MATLAB [37] for our simulations since our assumptions do not require models (e.g., interference models, 802.11, etc.) in network simulators.

Our scheduling policy intelligently compensates for the synchronization error through dynamic computation of the wake up interval. Another scheme that was previously used for compensating the synchronization error assumed an upper bound on the synchronization and used it as a *fixed* guard time. To evaluate the performance gain of dynamic adjustment of wake up intervals, we compare the performance of our scheme with that of the following fixed wake up interval scheme: The CH wakes up $\frac{L}{2}$ seconds earlier than the scheduled message arrival time (recall from Section IV-A that the scheduled arrival time is the time that the message *should* arrive). If the message does not arrive until $\frac{L}{2}$ after the scheduled arrival time, the CH goes back to sleep again; otherwise, it stays active until the message is received. To make the comparison fair, we use same message arrivals for both schemes.

We set the synchronization interval to 60 seconds. During each synchronization interval, the CH transmits to each cluster member in an equispaced manner, and obtains 2 pairs of corresponding times. We adopt the model used in RBS and TPSN to characterize the synchronization error. Specifically, the synchronization error is normally distributed with zero mean, $N(0, \sigma_0^2)$. In our simulations, we choose $\sigma_0 = 36.5 \mu s$, which is derived from [10]⁵. The clock skew of each node is chosen uniformly from $[1 - 50 \times 10^{-6}, 1 + 50 \times 10^{-6}]$ [28].

Table I summarizes the simulation parameters and other system constants. Unless otherwise specified, all the simulation results are averaged over 1000 runs.

TABLE I
SIMULATION PARAMETERS AND SYSTEM CONSTANTS

Threshold th	0.9
Idle power α_f (mW)	13
Receiving power α_r (mW)	13
Data rate R (kbps)	19.2
Message length L_p (byte)	8
Number of cluster member nodes M	10
Epoch duration T_e (minute)	20
Synchronization interval T_s (second)	60
Number of synchronization messages N_s	2
σ_0 (μs)	36.5
Transmission period T (second)	60

1) *Comparison with the Fixed Interval Scheme:* We first compare the message delivery performance of our scheme with the fixed interval scheme. From Equation (3), we have

$$VAR(\tau_p') \equiv \sigma_p^2 = \frac{\sigma_0^2}{a_i^2(j)} \left[\frac{1}{N_s} + \frac{1}{N_s} \frac{(\tau_p - \overline{C(j)})^2}{\overline{C^2(j)} - \overline{C(j)}^2} \right].$$

⁵In [10], measurements show that the average absolute error is $29.1 \mu s$. Therefore, $\int_{-\infty}^{\infty} |x| \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{x^2}{2\sigma_0^2}} dx = 29.1 \mu s \implies \sigma_0 \approx 36.5 \mu s$.

Within an epoch, the variance of the actual arrival time increases with τ_p , the scheduled arrival time. This is because the clock drifts away more and more as time progresses. As a result, for the fixed interval scheme, the capture rate will decrease as the scheduled arrival time increases. This is illustrated in Fig. 4. In the figure, we show how the capture rate changes as time goes on for both our scheme and the fixed interval scheme. We set $L = 3 \text{ ms}$ (recall that in the fixed interval scheme, the CH wakes up $\frac{L}{2}$ earlier than the scheduled arrival and stays active $\frac{L}{2}$ after the scheduled arrival). We observe that for the fixed interval scheme, the capture rate is very high at the beginning, but gradually decreases to below the threshold. If the message is scheduled to arrive near the end of the epoch ($T_e = 1200$), then the capture rate is only 0.55. In practice, this means that the fixed interval scheme cannot provide the threshold capture rate near the end of the epoch, which is undesirable. On the other hand, our scheme dynamically selects the wake up interval, so that the capture rate is always kept at no less than the threshold.

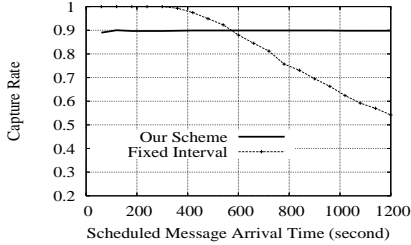
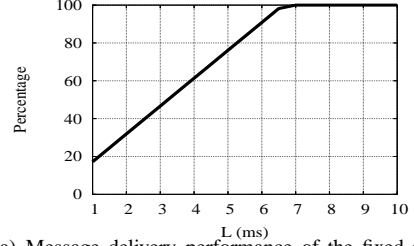


Fig. 4. Comparison of the message delivery performance

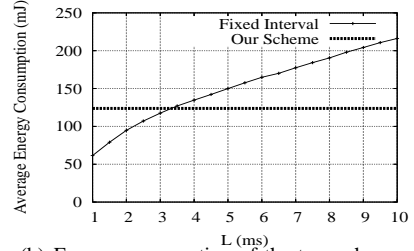
We next study the energy consumption properties of the two schemes. In Fig. 5(a), we vary L and compute the length of time (given as a percentage of the epoch) that the specified value of L is sufficient to give acceptable capture rates (above the threshold). For example, when $L = 3 \text{ ms}$, *percentage* $\approx 50\%$ means that if L is set to 3 ms , then for messages scheduled to arrive during the first half of the epoch, the capture rate is no less than the threshold; but if the message is scheduled to arrive during the second half of the epoch, the capture rate is lower than the threshold.

Fig. 5(b) illustrates the average cluster head energy consumption per epoch with different values of L . For comparison, we also include in the figure the average energy consumption per epoch of our scheme (the straight line). Since both our scheme and the fixed interval scheme use the same synchronization protocol, they consume the same amount of energy for synchronization. Therefore, we do not account for the energy consumed for synchronization here. From Fig. 5(a) and 5(b), we see that $L = 4 \text{ ms}$ can guarantee the threshold capture rate for only 60% of the epoch, but the energy consumption is already higher than our optimal scheduling scheme. In order to guarantee the threshold capture rate for the entire epoch, L must be set to at least 7 ms , with energy consumption 40% higher than the optimal scheduling scheme.

We have also simulated the combinations of parameters specified in Table II. The results were consistent with those



(a) Message delivery performance of the fixed interval scheme under different values of L



(b) Energy consumption of the two schemes

Fig. 5. Energy consumption properties of the fixed interval scheme and our scheme

discussed above, i.e., our scheme can guarantee a specified capture rate with lower energy consumption than the fixed interval scheme.

TABLE II

COMBINATION OF SIMULATION PARAMETERS

Threshold th	0.9	0.8	0.7	0.6
Epoch duration T_e (minute)	20	30	40	50
Synchronization interval T_s (second)	30	60	90	120
Number of synchronization messages N_s	2	4	6	8
Transmission period T (second)	60	120	180	240

2) *Impact of Synchronization Parameters*: In this section, we investigate how the choice of synchronization parameters, namely N_s and T_s , affects the energy savings of our scheme over the fixed interval scheme. To make a fair comparison between the energy consumption of our scheme and the fixed interval scheme, for each configuration we choose L to be the minimum interval that can guarantee the threshold capture rate for the entire epoch, i.e., $L = \min\{x: \text{the fixed interval scheme with } L = x \text{ can guarantee the threshold capture rate for the entire epoch}\}$.

Then we compute the performance gain of the optimal scheduling scheme (defined as the energy consumption ratio between the fixed interval scheme and the optimal scheduling scheme).

Fig. 6 depicts how the performance gain of the optimal scheduling scheme changes with N_s and T_s . We observe that as N_s increases, the performance gain of the optimal scheduling scheme gradually decreases. This can be explained as follows. The energy savings of optimal scheduling stem from reducing energy waste. Increased synchronization messages lead to a better synchronized cluster and reduce the idle listening time. Hence, the overall energy efficiency for both schemes is improved. Under this situation, though the optimal scheduling still consumes less energy than the fixed interval

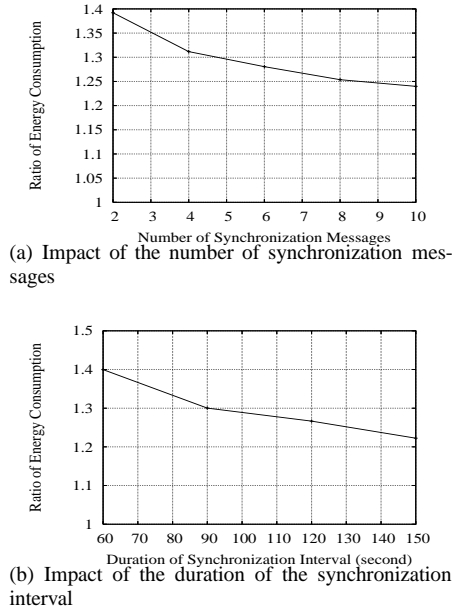


Fig. 6. Impact of synchronization scheme parameters on the ratio of the energy consumed by the fixed interval scheme to our scheme

scheme, the performance gain becomes smaller.

Similarly, when T_s increases, the cluster will become better synchronized. This can be observed from Equation (3). As T_s increases, $C(j, k)$, $k = 1 \dots N_s$ become more spread and $\overline{C^2(j)} - \overline{C(j)}^2$ increases. Hence, σ_p becomes smaller. This means that the actual message arrival is more likely to be in the vicinity of the scheduled arrival time, i.e., the network is more precisely synchronized.

The above discussion shows that we can save energy by increasing N_s and T_s . However, in practice, N_s and T_s cannot be arbitrarily increased. Increasing N_s means that more synchronization messages need to be exchanged between the CH and the cluster member nodes, which costs more energy; while increasing T_s means that the system will spend more time in synchronization operations, and cannot effectively perform other sensing and communication tasks⁶. Therefore, there exists a trade-off among synchronization and scheduling. We can achieve better performance in scheduling at the cost of more synchronization energy/time. An interesting question arises: what is the optimal scheme if synchronization and scheduling are *jointly* considered? We consider this to be an open issue and plan to investigate it in our future work.

V. PART II: QoS-AWARE ASSIGNMENT OF THE CAPTURE PROBABILITY THRESHOLD th

In Section IV, we studied the optimal sleep/wake scheduling problem under the assumption that the capture probability threshold th is *already given* and is identical for messages from different cluster members. In this section, we study how to obtain the capture probability threshold(s) to meet

⁶Fig. 6(b) shows that even if the synchronization interval is as large as 12.5% of the epoch duration ($T_s = 150seconds = 12.5\%T_e$), the performance gain is still larger than 120%.

the QoS requirement of the application with minimum energy consumption.

A. Model and Problem Definition

Consider a sensor network deployed for environmental monitoring. The network has already been clustered using one of the popular clustering techniques, e.g., [5]–[7]. Each sensor node periodically reports to its CH. Each message represents a certain amount of “information” about the environment. The CH uses the collected information to analyze interesting properties, e.g., the chemical contaminant in the area covered by this cluster. The accuracy of the analysis is determined by the *total* amount of information collected from all the member nodes, i.e., the *collective* information, as discussed in [38]–[40].

In many sensor networks, heterogeneity may exist among the sensor nodes. For example, some nodes may be equipped with an expensive sensor which provides high precision measurements, while others only have a low precision sensor for cost reasons. As a result, messages from different nodes may contain information of different quality and represent different “values.” To quantify the value of messages, Chen et al. [40] associate each message with a *utility value*, which represents the amount of useful information contained in it. Using the utility as the quantitative measure of service quality, they propose a general optimization framework for data transport in sensor networks. However, one assumption made in their work is that there is no redundancy in the network, hence the data collected from different sensors contributes additive utilities. In reality, redundant sensors may be deployed in the sensing area, and the information collected by nearby sensors may be correlated⁷.

In this work, we use a method similar to [40]. We associate each message with a *utility value*, which represents the amount of useful information contained in it; messages from the same node i have the same utility value U_i , $i = 1 \dots M$. Unlike [40], we consider that messages from different nodes may be correlated and exhibit redundancy. Therefore, to guarantee the analysis accuracy, the CH only needs to collect a *certain proportion* of the total utility. As long as this proportion is obtained, the requirement on each individual node can be chosen flexibly. We thus define Quality of Service (QoS) to be that the CH collects a desired proportion of the total utility. To achieve this QoS guarantee with minimum energy consumption, we formulate the following optimization problem.

Given an epoch j , as described in Section III, node i is scheduled to transmit at $\tau_i(j, h) = jT_e + T_s + i\frac{T_s}{M} + hT$, $0 \leq h < N$, $1 \leq i \leq M$. Let the capture probability threshold for all messages from node i during epoch j be $z_i(j)$. We aim to choose $z_i(j)$ to minimize the expected total energy consumption of the CH, and still collect the desirable proportion of the total utility:

$$(B) \text{ Min } \sum_{i=1}^M \sum_{h=1}^N E_i(j, h, z_i(j))$$

⁷Coverage scheduling can help reduce the redundancy, but experimental measurements [41] show that the correlation pattern can be very complex and it is difficult to *completely* remove the redundancy.

such that $\sum_{i=1}^M z_i(j)U_i \geq (1-r)\sum_{i=1}^M U_i$,
 $p_i \leq z_i(j) \leq 1, i = 1 \dots M$.

where:

- $E_i(j, h, z_i(j))$ is the expected energy consumption to “capture” the message that is scheduled to arrive at $jT_e + T_s + i\frac{T}{M} + hT$ with probability no less than $z_i(j)$. Note that once $z_i(j)$ is set, the CH will use the *optimal sleep/wake schedule* developed in Section IV-B. Hence, $E_i(j, h, z_i(j))$ is the minimum value of the objective function in Problem (A) (defined in Section IV-A) with $\tau_p = jT_e + T_s + i\frac{T}{M} + hT$ and $th = z_i(j)$;
- r is the redundancy level of the cluster, specifically, $100(1-r)\%$ of the sum utility is sufficient for the CH to make correct estimations of the environmental conditions; any additional information is redundant⁸;
- p_i is the minimum capture probability threshold for all messages from i . It is used to guarantee the reliability of the system. Without these constraints, it may happen that the thresholds assigned to certain nodes are so low that messages from these nodes are almost ignored. These constraints guarantee that all the cluster members have a minimum opportunity to pass their information on to the CH.

B. Solution

We first demonstrate that Problem (B) is not convex and is difficult to solve in general. Then we obtain a suboptimal solution that approximates the optimum with a ratio of 1.37.

Since the objective function in Problem (B) is the sum of many $E_i(j, h, z_i(j))$ s with different i, h (recall that j is fixed for each epoch), we first analyze the properties of $E_i(j, h, z_i(j))$. From our earlier discussions, $E_i(j, h, z_i(j))$ is exactly the minimum value of the objective function in Problem (A) with $\tau_p = jT_e + T_s + i\frac{T}{M} + hT$ and $th = z_i(j)$, which is (from Equation (4))

$$\sigma_p \alpha_I H(th) + \frac{L_p}{R} \alpha_r th.$$

Here, L_p is the message size, σ_p is computed from Equation (3), th is the required threshold, and $H(th)$ is given in Equation (5).

To obtain $E_i(j, h, z_i(j))$ from $E(th)$, we compute σ_p using Equation (3) with $\tau_p = jT_e + T_s + i\frac{T}{M} + hT$ and let $th = z_i(j)$, i.e.,

$$E_i(j, h, z_i(j)) = \alpha_r z_i(j) \frac{L_p}{R} + \alpha_I H(z_i(j)) \sqrt{\frac{\sigma_0^2}{a_i^2(j)} \frac{1}{N_s} \left[1 + \frac{(jT_e + T_s + i\frac{T}{M} + hT - \overline{C(j)})^2}{C^2(j) - (\overline{C(j)})^2} \right]}.$$

Therefore,

$$\sum_{i=1}^M \sum_{h=1}^N E_i(j, h, z_i(j)) = \sum_{i=1}^M A_i(j) H(z_i(j)) + B_i(j) z_i(j),$$

where

⁸The value of r is application specific and how to determine it is beyond the scope of the work we present in this paper. We should mention, however, that it is affected by factors such as node density, sensing coverage, and accuracy requirements. In practice, r can be obtained either through theoretical computation or from online training.

$$A_i(j) = \sum_{h=1}^N \alpha_I \sqrt{\frac{\sigma_0^2}{a_i^2(j)} \frac{1}{N_s} \left[1 + \frac{(jT_e + T_s + i\frac{T}{M} + hT - \overline{C(j)})^2}{C^2(j) - (\overline{C(j)})^2} \right]},$$

$$B_i(j) = N \alpha_r \frac{L_p}{R}$$

are non-negative parameters that do not change with $z_i(j)$. Further, because j is fixed for a given epoch, we omit j for brevity. Then we can write Problem (B) in this form:

$$(B1) \text{ Min } I_1(\vec{z}) = \sum_{i=1}^M A_i H(z_i) + B_i z_i$$

such that $\sum_{i=1}^M z_i U_i \geq (1-r)\sum_{i=1}^M U_i$,
 $p_i \leq z_i \leq 1, i = 1 \dots M$.

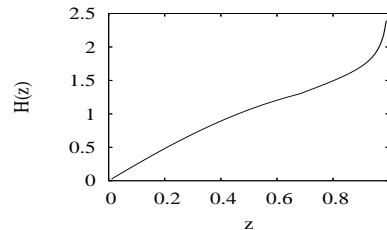


Fig. 7. $H(z)$

We numerically compute $H(z)$ and show the curve in Fig. 7. Obviously it is not convex; hence Problem (B1) is not convex. Further, we do not have an explicit analytical form for $H(z)$. This makes Problem (B1) hard to solve. Next we investigate the structure of the problem and obtain an approximate solution.

The following proposition characterizes $H(z)$.

- Proposition 4:* (1) For $z \geq 0.86$, $H(z)$ is strictly convex;
(2) for $z \in [0, 0.99]$, $1.86z < H(z) < 2.52z$.

We give the proof in our technical report [35]. The main idea is that though we do not have an explicit analytical form of $H(z)$, we have the bounds obtained from Proposition 3(2). Hence, we compute H' , H'' using implicit differentiation and bound them. We show that $H(z)$ is convex for the region $[0.86, 1)$; for the remaining region where $H(z)$ may not be convex, we can bound it fairly tightly.

Next, we approximate $H(z)$ with a convex function. Let $H_1(z) = 2z + 0.001z^2$, then it intersects $H(z)$ at $Z_0 \approx 0.95$.

Let

$$H_2(z) = \begin{cases} H_1(z) & 0 \leq z \leq Z_0 \\ H(z) & Z_0 \leq z < 1 \end{cases}$$

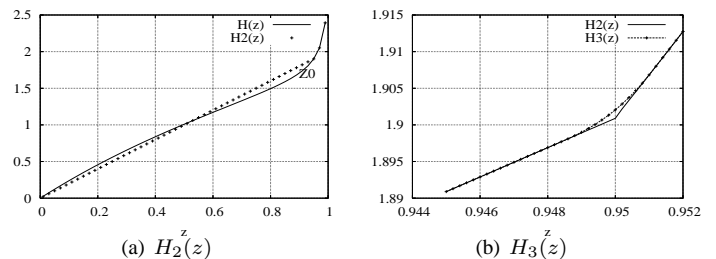


Fig. 8. Approximating $H(z)$

The following proposition shows that $H_2(z)$ is a convex approximation to $H(z)$.

- Proposition 5:* (1) $0.929 \leq \frac{H(z)}{H_2(z)} \leq 1.26$;
(2) $H_2(z)$ is strictly convex.

We proved this proposition using Proposition 4 in our technical report [35]. Fig. 8(a) illustrates that $H_2(z)$ is a good approximation to $H(z)$.

However, one issue is that $H_2(z)$ is not differentiable at Z_0 , because

$$H_2'(Z_0^-) = H_1'(Z_0) \approx 2.0019 < H_2'(Z_0^+) = H'(Z_0) \approx 5.7241.$$

Hence, we adjust $H_2(z)$ to make it differentiable.

Choose $Z_1 < Z_0 < Z_2$. Our idea is to replace $H_2(z)$ with a polynomial function in the interval $[Z_1, Z_2]$, so that the resulting function is continuous and differentiable everywhere. Let

$$\begin{aligned} q_0 &= H_2(Z_1), q_1 = H_2'(Z_1), \\ q_2 &= \frac{3[H_2(Z_2) - H_2(Z_1)]}{(Z_2 - Z_1)^2} - \frac{H_2'(Z_2) + 2H_2'(Z_1)}{Z_2 - Z_1}, \\ q_3 &= \frac{H(Z_2) - q_2(Z_2 - Z_1)^2 - q_1(Z_2 - Z_1) - q_0}{(Z_2 - Z_1)^3}, \\ H_3(z) &= \begin{cases} H_2(z) & 0 \leq z \leq Z_1 \\ q_3(z - Z_1)^3 + q_2(z - Z_1)^2 \\ \quad + q_1(z - Z_1) + q_0 & Z_1 \leq z \leq Z_2 \\ H_2(z) & Z_2 \leq z < 1 \end{cases} \end{aligned}$$

The proof of our next lemma is found in our technical report [35].

Lemma 2: (1) $H_3(z)$ is strictly convex and differentiable (see Fig. 8(b));

(2) If we choose $Z_1 = Z_0 - 0.0015$, $Z_2 = Z_0 + 0.0010$, then $0.925 \leq \frac{H(z)}{H_3(z)} \leq 1.26$;

Therefore, we can use $H_3(z)$ as a *convex and differentiable* approximation to $H(z)$. Now, we can obtain an approximate solution to (B1). Consider the following problem (B2):

$$\begin{aligned} \text{(B2) Minimize } & I_2(\vec{z}) = \sum_{i=1}^M A_i H_3(z_i) + B_i z_i \\ \text{such that } & \sum_{i=1}^M z_i U_i \geq (1-r) \sum_{i=1}^M U_i, \\ & p_i \leq z_i \leq 1, i = 1 \dots M. \end{aligned}$$

Because $H_3(z)$ is strictly convex and differentiable, Problem (B2) is a convex optimization problem and can be solved using conventional techniques. Note that the only difference between Problem (B1) and (B2) is that $H(z)$ is replaced by $H_3(z)$. The following proposition shows that the solution of (B2) is an approximate solution of (B1) with provable performance.

Proposition 6: Let \vec{z}^* be the optimal solution to (B1), \vec{z}^* be the optimal solution to (B2), then $I_1(\vec{z}^*) \leq 1.37I_1(\vec{z}^*)$.

Proof: From Lemma 2(2), $0.925 \leq \frac{H(z)}{H_3(z)} \leq 1.26$. Hence,

$$0.925 \leq \frac{I_1(\vec{z})}{I_2(\vec{z})} \leq 1.26. \quad (6)$$

Therefore,

$$\begin{aligned} I_1(\vec{z}^*) &\leq 1.26 \times I_2(\vec{z}^*) \leq 1.26 \times I_2(\vec{z}^*) \\ &\leq 1.26 \times I_1(\vec{z}^*)/0.925 \leq 1.37I_1(\vec{z}^*), \end{aligned}$$

where the first and third “ \leq ” come from Equation (6), and the second “ \leq ” holds since \vec{z}^* is the optimal solution of (B2). ■

The intuition behind the proof is that $H_3(\cdot)$ approximating $H(\cdot)$ means $I_1(\vec{z}^*) \approx I_2(\vec{z}^*)$ and $I_1(\vec{z}^*) \approx I_2(\vec{z}^*)$. But

because \vec{z}^* is the optimal solution to (B2), $I_2(\vec{z}^*) \leq I_2(\vec{z}^*)$. Therefore $I_1(\vec{z}^*) \approx I_2(\vec{z}^*)$ cannot be much larger than $I_1(\vec{z}^*) \approx I_2(\vec{z}^*)$.

Proposition 6 is important as it shows that \vec{z}^* is an approximate solution to (B1) with *approximation ratio* 1.37.

As described earlier, (B1) is a non-convex optimization problem, hence it is difficult to obtain the optimal solution \vec{z}^* . However, (B2) is a convex optimization problem and its optimal solution, \vec{z}^* , can be easily obtained using conventional optimization techniques such as the Logarithmic Barrier method [34]. Thus, in our approximation scheme, we first solve (B2) and obtain \vec{z}^* , then use them as the capture probability threshold(s). This may not result in minimum energy consumption, but from Proposition 6, the energy consumption using \vec{z}^* is no more than 37% higher than the optimum.

C. Simulations

In our simulations, we consider a cluster of $M = 10$ nodes. We assume the redundancy level of the cluster, r , is known to be 0.7. Half of the nodes have utility value of V_1 , while the other half have more capabilities and thus have a utility value of $V_2 > V_1$. We further set $p_i = P = 0.1, i = 1 \dots M$. Other simulation parameters are as specified before in Section IV-D, Table I.

We compare our approximation scheme with the previously used uniform assignment scheme, i.e., the scheme with $z_i = 1 - r, i = 1 \dots M$. In Fig. 9(a), we vary the value of $\frac{V_2}{V_1}$ and show the performance gain, which is defined as the ratio between the energy consumption of the two schemes. We observe that our scheme always outperforms the uniform assignment scheme, which demonstrates the effectiveness of our scheme. Further, the performance gain increases with $\frac{V_2}{V_1}$. This is because the performance gain of our scheme over the uniform assignment scheme stems from the differentiated treatment of the nodes. To guarantee the collective performance with limited energy, we “favor” the nodes with higher utility values. If all the nodes have the same utility value ($\frac{V_2}{V_1} = 1$), there is no benefit in treating the nodes differently; as $\frac{V_2}{V_1}$ increases, the difference between nodes becomes larger, which makes it advantageous to provide differentiated services to the nodes and favor more important ones.

In Fig. 9(b) we keep the value of $\frac{V_2}{V_1}$ fixed at 3, and vary the value of P (we still set $p_i = P, \forall i = 1 \dots M$). We observe that the performance gain decreases as P increases. This is expected. When P is small, some nodes may be assigned a very small capture probability threshold, which makes the system less reliable. As P increases, the system reliability increases. At the same time, the region $z_i \geq p_i$ shrinks, which means we have less flexibility in selecting z_i . Consequently, the performance gain becomes less significant. Hence, the choice of P exhibits a tradeoff between system reliability and energy savings.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have studied sleep/wake scheduling for low duty cycle sensor networks. Our work explicitly considers the

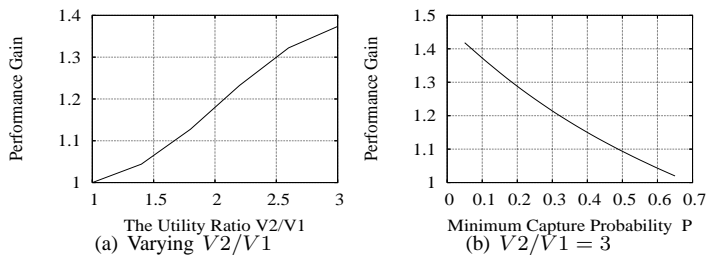


Fig. 9. Performance gain over the uniform assignment scheme

effect of synchronization error in the design of sleep/wake scheduling algorithm. In contrast, most previous works on sleep/wake scheduling either assume perfect synchronization, or assume an upper bound on the clock disagreement and uses a fixed guard time to compensate for the synchronization error. We utilize a widely used synchronization scheme, which was proposed in the well known RBS protocol. We demonstrated that this scheme, though it achieves microsecond level synchronization *immediately after* the exchange of synchronization messages, *turns out to have non-negligible clock disagreement as time progresses*. Therefore, we conclude that the design of any sleep/wake scheduling algorithm must take into account the impact of this synchronization error, and we study the optimal sleep/wake scheduling scheme with consideration of the synchronization error.

We first study how to decide the sleep/wake schedule to achieve a *given* constraint on the message capture probability with minimum energy consumption. The problem is non-convex, and hence cannot be directly solved by conventional convex optimization techniques. By exploiting the structure of the problem, we were able to transform the original non-convex problem into a convex equivalent, and solve it using an efficient search method.

We then removed the assumption that the capture probability threshold is *already given*, and studied how to obtain it to meet the QoS requirement of the application. We observe that in many sensor network applications, the QoS is not decided by the performance of any individual node, but by the *collective* performance of all the related nodes. We thus formulate an optimization problem, which aims to set the threshold for messages from each individual node such that the expected energy consumption is minimized, but the collective performance is guaranteed. The problem turns out to be non-convex and hard to solve exactly. However, by investigating its unique structure, we have obtained a suboptimal solution with approximation ratio 1.37. Simulations show that our approximate solution significantly outperforms a scheme without differentiated treatment of the nodes.

We should point out that the work conducted in this paper has largely been motivated by sensor networks running continuous monitoring applications. While this encompasses a large class of interesting applications, our approach is not limited to such applications. The major requirement of our approach is that the sender and the receiver agree upon message arrival times. This requirement can also be satisfied in many sensor applications with a *hybrid data delivery pattern* [42], [43]. For

example, a sensor network which monitors the concentration of a chemical can have two modes: the silent mode and the vigilant mode. When the average concentration collected from the whole network is lower than a certain threshold, the base station will set the network in silent mode, where nodes monitor the chemical concentration and periodically report to the base station; however, if the base station finds the average concentration to exceed the threshold, it can trigger the network into vigilant mode, where all the nodes stay active and transmissions can occur at any time. For such applications with a hybrid data delivery pattern, our approach is ideal for use in the silent mode.

We have focused on the single hop intra-cluster communication scenario in this work. After messages are received by the cluster head, they may need to be forwarded to the base station, potentially over multiple hops. The cluster head may also need to relay messages from other cluster heads to the base station. An important question is how to decide the sleep/wake schedules of the cluster heads over such multiple hops. Section III gave a simple example mechanism. A more efficient solution would be to develop an adaptive sleep/wake scheduling methodology, as in this paper, for inter-cluster communications. This is an open issue that merits further investigation.

In this work, we express the redundancy within a cluster with a single redundancy level r . The correlation pattern within the cluster may be more complex, e.g., sensing data collected from nearby sensors are more strongly correlated than data from sensors at a longer distance. How to assign the capture thresholds under complex correlation patterns is a challenging problem for future work.

REFERENCES

- [1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *Proc. of ACM WSN*, September 2002.
- [2] G. Tolle, J. Polastre, R. Szewczyk, N. Turner, K. Tu, S. Burgess, D. Gay, P. Buonadonna, W. Hong, T. Dawson, and D. Culler, "A Macroscopic in the Redwoods," in *Proc. of ACM SenSys*, 2005.
- [3] S. C. Visweswara, A. A. Goel, and R. Dutta, "An Adaptive Ad-hoc Self-Organizing Scheduling for Quasi-Periodic Sensor Traffic," in *Proc. of IEEE SECON*, September 2004.
- [4] K. Srinivasan, M. Ndoj, H. Nie, H. Xia, K. Kaluri, and D. Ingraham, "Wireless Technologies for Condition-Based Maintenance (CBM) in Petroleum Plants," *Proc. of the 1st International Conference on Distributed Computing in Sensor Systems (Poster Session)*, 2005.
- [5] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, October 2002.
- [6] O. Younis and S. Fahmy, "Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach," in *Proc. of IEEE INFOCOM*, Hong Kong, March 2004.
- [7] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks," in *Proc. of IEEE INFOCOM*, April 2001.
- [8] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time synchronization Using Reference Broadcasts," in *Proc. of USENIX/ACM OSDI*, 2002.
- [9] J. Elson and K. Romer, "Wireless Sensor Networks: A new Regime for Time Synchronization," in *Proc. of HotNets-I*, October 2002.
- [10] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync Protocol for Sensor Networks," in *Proc. of ACM SenSys*, 2003.

- [11] J. Elson, L. Girod, and D. Estrin, "Short Paper: A Wireless Time-Synchronized COTS Sensor Platform, Part I: System Architecture," in *Proc. of the IEEE CAS Workshop on Wireless Communications and Networking*, September 2002.
- [12] Q. Gao, K. J. Blow, and D. J. Holding, "Simple algorithm for improving time synchronization in wireless sensor networks," *Electronics Letters*, vol. 40, pp. 889–891, July 2004.
- [13] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," in *Proc. of ACM SenSys*, 2004.
- [14] D. Cox, A. Milenkovic, and E. Jovanov, "Time Synchronization for ZigBee Networks," in *Proc. of the 37th Southeastern Symposium on System Theory*, Tuskegee, AL, March 2005.
- [15] S. PalChaudhuri, A. K. Saha, and D. B. Johnson, "Adaptive clock synchronization in sensor networks," in *Proc. of IEEE/ACM IPSN*, 2004.
- [16] S. Ganerwal, D. Ganesan, H. Shim, V. Tsatsis, and M. B. Srivastava, "Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks," in *Proc. of ACM SenSys*, 2005.
- [17] J. R. Vig, "Introduction to Quartz Frequency Standards," Technical Report SLCET-TR-92-1, Army Research Laboratory, October 1992, available at <http://www.ieee-uffc.org/freqcontrol/quartz/vig/vigtoc.htm>.
- [18] S. Singh and C. Raghavendra, "PAMAS: Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks," *ACM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, July 1998.
- [19] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proc. of IEEE INFOCOM*, New York, NY, June 2002.
- [20] T. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proc. of ACM SenSys*, 2003.
- [21] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," in *Proc. of ACM SenSys*, 2003.
- [22] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *Proc. of ACM SenSys*, 2004.
- [23] S. Coleri, A. Puri, and P. Varaiya, "Power Efficient system for Sensor Networks," in *Proc. of IEEE ISCC*, July 2003.
- [24] M. L. Sichitiu, "Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks," in *Proc. of IEEE INFOCOM*, 2004.
- [25] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *Proc. of ACM MOBIHOC*, 2003.
- [26] C. Hsin and M. Liu, "Randomly Duty-cycled Wireless Sensor Networks: Dynamics of Coverage," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3182–3192, November 2006.
- [27] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System Architecture Directions for Networked Sensors," in *Proc. of ACM ASPLOS*, 2000, pp. 93–104.
- [28] <http://www.tinyos.net/scoop/special/hardware>.
- [29] W. Ye and J. Heidemann, "Ultra-low duty cycle MAC with scheduled channel polling," USC/Information Sciences Institute, Tech. Rep. ISI-TR-2005-604, July 2005. [Online]. Available: <http://www.isi.edu/~johnh/PAPERS/Ye05a.html>
- [30] "CC1000 low power FSK transceiver," Chipcon Corporation. <http://www.chipcon.com>.
- [31] C. Lynch and F. O'Reilly, "Processor choice for wireless sensor networks," in *Proc. Workshop on Real-World Wireless Sensor Networks*, 2005.
- [32] A. J. Martin, M. Nyström, K. Papadantonakis, P. I. Penzes, P. Prakash, C. G. Wong, J. Chang, K. S. Ko, B. Lee, E. Ou, J. Pugh, E.-V. Talvala, J. T. Tong, and A. Tura, "The lutonium: A sub-nanojoule asynchronous 8051 microcontroller," in *Proc. 9th IEEE International Symposium on Asynchronous Systems and Circuits*, 2003.
- [33] J. L. Devore, *Probability and Statistics for Engineering and the Sciences*. International Thomson Publishing Inc., 1995.
- [34] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [35] Y. Wu, S. Fahmy, and N. B. Shroff, "Optimal QoS-aware Sleep/Wake Scheduling for Time-Synchronized Sensor Networks," Technical Report, 2006, available at <http://www.cs.purdue.edu/homes/wu26/techrep.pdf>.
- [36] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*. John Wiley & Sons, Inc., 2001.
- [37] "MATLAB," MathWorks Corporation. <http://www.mathworks.com>, 2004.
- [38] R. Iyer and L. Kleinrock, "QoS Control for Sensor Networks," in *Proc. of IEEE ICC*, 2003.
- [39] J. Kay and J. Frolik, "QoS Analysis and Control in Wireless Sensor Networks," in *Proc. of IEEE MASS*, 2004.
- [40] W.-P. Chen and L. Sha, "An energy-aware data-centric generic utility based approach in wireless sensor networks," in *Proc. of IEEE/ACM IPSN*, 2004.
- [41] F. Koushanfar, N. Taft, and M. Potkonjak, "Sleeping coordination for comprehensive sensing using isotonic regression and domatic partitions," in *Proc. of IEEE INFOCOM*, April 2006.
- [42] Y. Yao and J. E. Gehrke, "Query processing in sensor networks," in *Proc. of the 1st Biennial Conference on Innovative Data Systems Research*, 2003.
- [43] A. Boukerche, R. Werner, N. Pazzi, and R. Borges-Araujo, "A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications," in *Proc. of the 7th ACM MSWiM*, 2004.

APPENDIX

Proposition 2: $G''(w) > 0$.

Proof: Since $g(x)$ is the pdf of the standard normal distribution, we will use the fact that: $g(x) = g(-x)$, $g'(x) = -xg(x)$.

From formulation (C),

$$G'' = (1 - th)s''(w) + (s - w)g'(w) + [s'(w) - 1]g(w).$$

It is sufficient to show that

$$s''(w) > 0$$

and

$$(s - w)g'(w) + [s'(w) - 1]g(w) \geq 0.$$

We first examine $s''(w)$. Taking derivatives on both sides of $Q(w) - Q(s(w)) = th$, we have

$$-g(w) + g(s)s'(w) = 0 \implies s'(w) = \frac{g(w)}{g(s)}.$$

Thus,

$$s''(w) = \frac{g'(w)g(s) - g(w)g'(s)s'(w)}{g^2(s)}.$$

Since $g'(x) = -xg(x)$, we have

$$s''(w) = \frac{-wg(w)g^2(s) + sg(s)g^2(w)}{g^3(s)} = \frac{g^2(w)}{g(s)} \left[\frac{s}{g(s)} - \frac{w}{g(w)} \right].$$

Since $\left(\frac{x}{g(x)}\right)' = \frac{g(x) + x^2g(x)}{g^2(x)} > 0$, $\frac{x}{g(x)}$ is a strictly increasing function. Therefore,

$$s > w \implies \frac{s}{g(s)} - \frac{w}{g(w)} > 0 \implies s''(w) > 0.$$

Next,

$$(s - w)g'(w) + [s'(w) - 1]g(w) = -w(s - w)g(w) + \frac{g(w)}{g(s)}[g(w) - g(s)] = g(w)[-w(s - w) + \frac{g(w) - g(s)}{g(s)}].$$

Since $g(w) > 0$, it is sufficient to show that

$$-w(s - w) + \frac{g(w) - g(s)}{g(s)} > 0.$$

There are three cases:

- $w < s \leq 0$

By the Mean Value Theorem,

$$g(w) - g(s) = (w - s)g'(\zeta), \zeta \in [w, s],$$

we have

$$-w(s - w) + \frac{g(w) - g(s)}{g(s)} = -w(s - w) \left[1 - \frac{g(\zeta)}{g(s)} \frac{\zeta}{w} \right].$$

Then

$$\begin{aligned} w \leq \zeta \leq s \leq 0 &\implies 0 \leq \frac{g(\zeta)}{g(s)} \leq 1, 0 \leq \frac{\zeta}{w} \leq 1 \\ &\implies 1 - \frac{g(\zeta)}{g(s)} \frac{\zeta}{w} \geq 0 \\ &\implies -w(s-w) \left[1 - \frac{g(\zeta)}{g(s)} \frac{\zeta}{w}\right] \geq 0. \end{aligned}$$

- $0 \leq w < s$

This case can be proved using the Mean Value Theorem as above: observe that $0 \leq w < s$ implies $\frac{g(\zeta)}{g(s)} \geq 1$ and $\frac{\zeta}{w} \geq 1, \forall \zeta \in [w, s]$.

- $w \leq 0 \leq s$

If $g(w) \geq g(s)$, then

$$\begin{aligned} -w(s-w) \geq 0, \frac{g(w) - g(s)}{g(s)} \geq 0 &\implies -w(s-w) \\ &+ \frac{g(w) - g(s)}{g(s)} \geq 0. \end{aligned}$$

Otherwise, $g(w) < g(s) \implies w < -s$. Hence, by the Mean Value Theorem,

$$\begin{aligned} -w(s-w) + \frac{g(w) - g(s)}{g(s)} &= -w(s-w) + \\ &+ \frac{g(w) - g(-s)}{g(-s)} = -w(s-w) - \frac{\zeta g(\zeta)}{g(-s)}(w+s) \\ &= -w(s-w) \left[1 + \frac{\zeta}{w} \frac{g(\zeta)}{g(-s)} \frac{w+s}{s-w}\right], \end{aligned}$$

where $\zeta \in [w, -s]$.

$$\begin{aligned} w \leq \zeta \leq -s \leq 0 &\implies 0 \leq \frac{\zeta}{w} \leq 1, 0 \leq \frac{g(\zeta)}{g(-s)} \leq \\ &\leq 1, -1 \leq \frac{w+s}{s-w} \leq 0 \implies \frac{\zeta}{w} \frac{g(\zeta)}{g(-s)} \frac{w+s}{s-w} \geq -1 \\ &\implies -w(s-w) \left[1 + \frac{\zeta}{w} \frac{g(\zeta)}{g(-s)} \frac{w+s}{s-w}\right] \geq 0. \end{aligned}$$

In all, $-w(s-w) + \frac{g(w)-g(s)}{g(s)} \geq 0$. Combining this with $s''(w) > 0$, we have $G''(w) > 0$. ■

Summary of Notation: We list the symbols used in Table III.

AUTHOR BIOGRAPHIES

Yan Wu [Student Member] received the B.S. degree from the University of Science and Technology of China and M.S. degree from the Chinese Academy of Sciences. Since 2002, he has been pursuing his Ph.D. degree at the Computer Science Department, Purdue University. His research interests lie in resource allocation, optimization and cross-layer design in wireless and sensor networks.

Sonia Fahmy [Senior Member] is an associate professor at the Computer Science department at Purdue University. She received her PhD degree from the Ohio State University in 1999. She is currently investigating Internet tomography, network security, and wireless sensor networks. She received the National Science Foundation CAREER award in 2003, and the Schlumberger technical merit award in 2000. She is

TABLE III
LIST OF SYMBOLS

M	Number of cluster member nodes (n_1, \dots, n_M)
T_e	Epoch duration
T_s	Synchronization interval
N_s	Number of synchronization messages
T	Transmission period
N	Rounds of transmissions in an epoch
$(C(j, k), t_i(j, k)), k = 1 \dots N_s$	Corresponding time instants between CH and member n_i in epoch j
$a_i(j), b_i(j)$	Clock skew and phase offset (respectively) between n_i and CH in epoch j
$\hat{a}_i(j), \hat{b}_i(j)$	Estimates of $a_i(j), b_i(j)$
σ_0^2	Variance of the random error
τ_p	Scheduled arrival time of packet p
τ_p	Actual arrival time of packet p
w_p	Wake up time to receive packet p
s_p	Sleep time if packet p is not received
th	Capture probability threshold
α_I	Idle power
α_r	Receiving power
R	Data rate
L_p	Message length
$C(j)$ and $C^2(j)$	Refer to Equation (3)
$\hat{\tau}$ and (w, s)	Normalized arrival time and normalized wake up interval respectively
$g(\cdot)$	Probability Density Function of standard normal distribution
$Q(\cdot)$	Complementary cumulative distribution function
$H(\cdot)$	Refer to Equation (5)
$\tau_i(j, h), z_i(j)$ and $E_i(j, h, z_i(j))$	Refer to Problem (B)
r	Redundancy level of the cluster
p_i	Minimum capture probability threshold for messages from i
$A_i(j)$ and $B_i(j)$	Refer to Section V-B
$I_1(\bar{z})$ and $I_2(\bar{z})$	Refer to Problems (B1) and (B2)
$H_1(\cdot), H_2(\cdot), H_3(\cdot)$	Refer to Section V-B

a member of the ACM. For more information, please see: <http://www.cs.purdue.edu/~fahmy/>

Ness B. Shroff [Fellow] received his Ph.D. degree from Columbia University, NY in 1994 and joined Purdue University immediately thereafter. At Purdue, he became Professor of the school of Electrical and Computer Engineering in 2003 and director of CWSA in 2004, a university-wide center on wireless systems and applications. In 2007, he joined The Ohio State University as the Ohio Eminent Scholar of Networking and Communications, and Professor of ECE and CSE. His research interests span the areas of wireless and wireline communication networks. For more information, please see: <http://www.ece.ohio-state.edu/~shroff/>