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

Yan Wu, Sonia Fahmy, Ness B. Shroff

Center for Wireless Systems and Applications (CWSA), Purdue University

Abstract— We study the *sleep/wake scheduling problem* in the context of *clustered sensor networks*. Unlike most prior work on sleep/wake scheduling that assumed perfect time synchronization in the network, we argue that the synchronization error is non-negligible and demonstrate its effect with a widely used synchronization scheme. We conclude that the design of any sleep/wake scheduling algorithm must take into account the impact 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). 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, thus cannot be directly solved by conventional convex optimization techniques. However, 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.

In the second part, we remove the assumption that the capture probability threshold is *already given*, and study how to decide it to meet the Quality of Services (QoS) requirement of the application. We observe that in many sensor network applications, a group of sensors collaborate to perform common task(s). Therefore, the QoS is usually not decided by the performance of any individual node, but by the *collective* performance of all the related nodes. To achieve the collective performance with minimum energy consumption, intuitively we should provide *differentiated* services for the nodes and favor more important ones. We thus formulate an optimization problem, which aims to set the capture probability threshold for messages from each individual node such that the expected energy consumption is minimized, while the collective performance 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. Simulations show that our approximate solution significantly outperforms a scheme without differentiated treatment of the nodes.

I. INTRODUCTION

There is growing interest in the area of wireless sensor networks. An important class of sensor network applications is the class of continuous monitoring applications: applications employing a large number of sensor nodes for continuous sensing and data delivery, where each sensor *periodically* produces

a small amount of data and reports to a single (or a few) base station(s). This application class includes many typical sensor network applications such as habitat monitoring [1] and civil structure monitoring [2].

A scalable method to manage the large number of sensor nodes in such applications 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. One problem with clustering is that the CH is heavily utilized for both intra-cluster coordination and inter-cluster communications. Therefore, energy efficiency of the CH is an important concern.

Sleep/wake scheduling has been proposed to reduce energy consumption in sensor networks [3]–[11]. 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 with each other. 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, synchronization is imperfect [12]–[14], and as time progresses, the clock disagreement becomes more and more significant. Periodic re-synchronization can prevent the clocks from drifting away, 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 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 continuously monitor the environment and *periodically* report to the CH. Because the traffic is highly regular and the load is low, the CH can go to sleep when no activity is going on and only wake up intermittently to send and receive messages. The following question hence becomes critical: *When should the CH wake up and how long should it stay active?*

With perfect synchronization, the CH and the cluster mem-

ber simply agree upon a time and wake up simultaneously. However, in practice, synchronization always has error. To illustrate the impact of the synchronization error, we investigate a widely used synchronization scheme, RBS [12]. We show 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 almost all synchronization schemes: due to non-deterministic factors, the synchronization error will grow with time until the next exchange of synchronization messages. We thus 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 trade-off between energy consumption and message delivery performance (defined as the message capture probability in this work). 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.

In the second part, we remove the assumption that the capture probability threshold is *already given*, and study how to decide it to meet the Quality of Services (QoS) requirement of the application. Unlike most Internet applications where different users compete with each other for network resources, in many sensor network applications, a group of sensors collaborate to perform common task(s). Therefore, the QoS for sensor networks is usually not decided by the performance of any individual node, but by the *collective* performance of all related nodes. To achieve the collective performance with minimum energy consumption, we should provide *differentiated* services for the nodes and favor more important ones. We thus formulate an optimization problem, which aims to set the capture probability threshold for messages from each individual node such that the expected energy consumption is minimized, and still the collective performance 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 describes the system model. Section III discusses the optimal sleep/wake scheduling problem. Section IV studies how to assign the threshold for messages originating from each individual node. Section V concludes the paper.

II. SYSTEM MODEL

Consider a cluster which has been constructed using an existing clustering protocol (e.g., [15], [16]). The cluster consists of a single cluster head (CH) and M cluster member nodes

$n_1, n_2, \ldots n_M$. Time is divided into recurring *epochs* with constant duration T_e. Like many MAC protocols for sensor networks [4], [6], each epoch begins with a synchronization interval T_s followed by a transmission interval. 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 the 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 occurs at a lower frequency than synchronization, i.e., the time between re-clustering the network consists of one or more epochs.

In this work, we focus on intra-cluster communications, i.e., communications between the cluster members and the CH. We 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 ($10-40$ kbps). Further, measurements show that among all the sensor node components, the radio consumes the most significant amount of energy, hence we only account for energy consumption of the radio. Finally, the communication range for sensor nodes is very short, so we consider the propagation delay to be negligible.

A. Synchronization Algorithm

Time synchronization has been extensively studied in the context of wireless sensor networks [12]–[14], [17]–[19]. 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, pressure, voltage, and radiation [20]. 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 [18].

In this work, we adopt the well known RBS synchronization scheme, and study the sleep/wake scheduling problem under this scheme[1]. The scheme consists of two steps: (1) Exchange synchronization messages to obtain multiple pairs of corresponding time instants; and (2) Use linear regression to estimate the clock skew and phase offset.

---

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

For the purpose of intra-cluster communication, cluster 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 $(C(j,k), t_i(j,k)), k = 1 \ldots N_s$, where $C(j,k), t_i(j,k)$ denote the kth 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 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. Measurements [12] show that $e_i(j,k)$ follows a well-behaved *normal* distribution with zero mean $N(0, \sigma_0^2)$, and σ_0 is on the order of several tens of microseconds.

In each epoch j, pairs $(C(j,k), t_i(j,k)), k = 1 \ldots N_s$ 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 $\hat{a}_i(j), \hat{b}_i(j)$.

III. PART I: THE OPTIMAL SLEEP/WAKE SCHEDULING PROBLEM

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., $\hat{t}_i(\tau_p) = \hat{a}_i(j)\tau_p + \hat{b}_i(j)$, and 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:

$$\hat{\tau}_p = \frac{\hat{t}_i(\tau_p) - b_i(j)}{a_i(j)} = \tau_p + \frac{(\hat{a}_i(j) - a_i(j))\tau_p + \hat{b}_i(j) - b_i(j)}{a_i(j)}. \quad (2)$$

As given in Equation (1), random errors exist in the measurements. Therefore, $(\hat{a}_i(j), \hat{b}_i(j))$ is also random and may not equal $(a_i(j), b_i(j))$. As a result, the actual arrival time $\hat{\tau}_p$ 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. 1). This leads to the following question: *When should the CH wake up and how long should it stay active?*

Intuitively, if the CH wakes up much earlier than τ_p and stays active for a long time, it has a high probability of "capturing" the message; however, waking up early and staying active for a long time wastes energy. In order to reduce energy consumption, yet still guarantee high message delivery performance, we formulate the following optimal sleep/wake scheduling problem which attempts to minimize
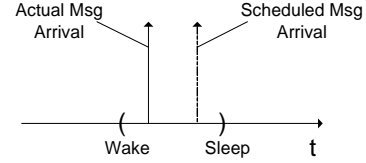

Fig. 1. Wake up interval to capture the message

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 $\hat{\tau}_p$ 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, and waits for the message until s_p. Our goal is to determine w_p and s_p to minimize the expected energy consumption as described by the following optimization problem:

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

such that $Prob\{\hat{\tau}_p \in (w_p, s_p)\} \geq th,$

where:

- α_I / α_r is the power consumption during idle/reception time;
- L_p is the length of the message;
- R is the data rate;
- $f_{\hat{\tau}_p}()$ is the Probability Density Function (PDF) of $\hat{\tau}_p$;
- th is the capture probability threshold, $0 < th < 1$. Its value is application specific. In this section, we assume that the value of th is *already given* and is the same for messages coming from different cluster members, i.e., all members are treated "uniformly." In Section IV, 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., $\hat{\tau}_p \notin (w_p, s_p)$. 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 $f_{\hat{\tau}_p}(x)$. By linear regression analysis [21], we find that $\hat{\tau}_p$ is *normally* distributed and

$$E(\hat{\tau}_p) = \tau_p, \quad (3)$$

$$VAR(\hat{\tau}_p) \equiv \sigma_p^2 = \frac{\sigma_0^2}{a_i^2(j)} \frac{1}{N_s} [1 + \frac{(\tau_p - \overline{C(j,k)})^2}{\overline{C^2(j,k)} - (\overline{C(j,k)})^2}],$$

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

Substituting Equation (3) into problem (A), and letting $\hat{\tau} = \frac{\hat{\tau}_p - \tau_p}{\sigma_p}, w = \frac{w_p - \tau_p}{\sigma_p}, s = \frac{s_p - \tau_p}{\sigma_p}$, 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) Minimize

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

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

where $g(x)$ is the PDF 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, which can be shown by computing the Hessian matrix. Due to the non-convexity, we cannot use conventional convex optimization techniques [22] 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.

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 (A1), then $Q(w^*) - Q(s^*) = th$.

Proof: Due to space limitations, we give all the detailed proofs in our technical report [23].

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 is easily understood as we can imagine that to guarantee a larger capture probability, more energy will be consumed.

Substituting $Q(w^*) - Q(s^*) = th$ into $F(w, s)$, we have

$$F(w, s) = [(1 - th)s - w + g(w) - g(s)]\sigma_p\alpha_I + th\frac{L_p}{R}\alpha_r. \quad (4)$$

Substituting Equation (4) into (A1), removing the constant term $th\frac{L_p}{R}\alpha_r$, then extracting the common factor $\sigma_p\alpha_I$, formulation (A1) becomes:

(A2) Minimize $G(w) = (1 - th)s(w) - w + g(w) - g(s(w))$, such that $s(w) = Q^{-1}(Q(w) - th)$ and $w < Q^{-1}(th)$.

So far, we have transformed the original formulation (A) into an equivalent formulation (A2). Next, we solve (A2).

The following proposition shows that $G(w)$ is a convex function, and gives the position of the global minimum.

Proposition 2: (1) $G''(w) > 0$;

(2) 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.

Since w_0 is the unique minimum on (w_l, w_u), we can use the Golden Search method to find w_0 [24]. 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.

Once we obtain w^*, s^*, we compute the optimal sleep/wake schedule as $(w_p^* = \tau_p + w^*\sigma_p, s_p^* = \tau_p + s^*\sigma_p)$.

We conduct simulations to study our optimal sleep/wake scheduling scheme. Simulation results show that our scheme significantly outperforms a previous scheme that does not

intelligently consider the synchronization error. For brevity, we omit the details, but include them in our technical report [23].

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

In Section III, we studied the optimal sleep/wake scheduling problem under the assumption that the capture probability threshold th is *already given*. In this section, we study how to decide the capture probability threshold(s) to meet the QoS requirement of the application with minimum energy consumption.

A. QoS Model and Problem Definition

Several studies have investigated QoS support issues in sensor networks [25]–[27]. In contrast to most Internet applications where different users compete with each other for network resources, in many sensor networks a group of sensors collaborate to perform common task(s). Therefore, the QoS for sensor networks is usually not decided by the performance of any individual node, but by the *collective* performance of all related nodes. For example, consider an environment monitoring system where sensor nodes periodically report to a base station. Each message contains some sensing data and represents certain amount of "information" about the environment. The base station uses the collected information to analyze interesting properties, e.g., the chemical contaminant in the area, moving speed of a target, etc. The service quality is defined as the accuracy of the analysis, which is decided by the total amount of information collected from all the nodes, namely, the *collective* information.

Another property of many sensor networks is that heterogeneity may exist among sensor nodes. For example, some nodes may be equipped with an expensive sensor which provides high precision measurements, while others may only have a low precision sensor for cost reasons. As a result, messages from different nodes may contain information of different qualities and represent different "values." To quantify the value of messages, we use a method similar to [27]. 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, $i = 1 \ldots M$. Further, we note that in sensor networks, redundant sensors are often deployed to achieve robustness, and hence the information collected by nearby sensors may be correlated. Therefore, to guarantee QoS, the CH only needs to collect a certain *proportion* of the total utility. As long as this proportion is collected, the requirement on each *individual* node can be chosen with flexibility. To collect this desired proportion of total utility with minimum energy consumption, we formulate the following optimization problem.

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

(B) Minimize $\sum_{i=1}^{M}\sum_{h=1}^{N} E_i(j,h,z_i(j))$
such that $\sum_{i=1}^{M} z_i(j)U_i \geq (1-r)\sum_{i=1}^{M} U_i,$
$p_i \leq z_i(j) \leq 1, i = 1\ldots M.$

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 III-B. Hence, $E_i(j,h,z_i(j))$ is the minimum value of the objective function in Problem (A) (defined in Section III-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 adequate for the CH to correctly analyze the environment; any additional information is redundant[2];
- 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, the thresholds assigned to certain nodes may be so low that their messages 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. 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 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 $th = z_i(j)$. Following the derivations in Section III-B, the minimum value of the objective function in Problem (A) can be expressed as a function of th in the following format:
$$E(th) = \sigma_p \alpha_I H(th) + \frac{L_p}{R}\alpha_r th,$$
where $H(th) = min\{G(w) = (1-th)s(w) - w + g(w) - g(s(w)) : s(w) = Q^{-1}(Q(w) - th), -\infty < w < Q^{-1}(th)\}.$

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., $E_i(j,h,z_i(j)) = \alpha_I H(z_i(j))\sqrt{\frac{\sigma_0^2}{a_i^2(j)}\frac{1}{N_s}[1 + \frac{(jT_e+T_s+i\frac{T}{M}+hT-\overline{C(j,k)})^2}{\overline{C^2(j,k)}-(\overline{C(j,k)})^2}]} + \alpha_r z_i(j)\frac{L_p}{R}$. Therefore, $\sum_{i=1}^{M}\sum_{h=1}^{N} E_i(j,h,z_i(j))$ can be written as:

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

where

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

[2]The value of r is application specific and how to decide it is beyond the scope of this work, but we should mention that it is affected by factors including node density, sensing coverage, and accuracy requirements.
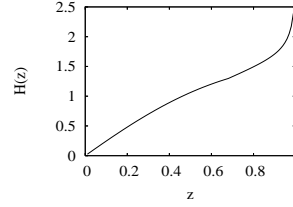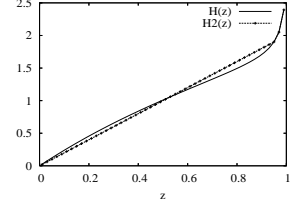

Fig. 2. H(z) is non-convex

Fig. 3. $H_2(z)$ approximates $H(z)$

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) Minimize $I_1(\overrightarrow{z}) = \sum_{i=1}^{M} A_i H(z_i) + B_i z_i$
such that $\sum_{i=1}^{M} z_i U_i \geq (1-r)\sum_{i=1}^{M} U_i,$
$p_i \leq z_i \leq 1, i = 1\ldots M.$

We plot $H(z)$ in Fig. 2. 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 exactly. We now investigate the structure of the problem to obtain an approximate solution.

The following lemma characterizes $H(z)$.
Lemma 1: (1) For $z \geq 0.86$, $H(z)$ is strictly convex;
(2) For $z \in [0, 0.99], 1.86z < H(z) < 2.52z$.
Hence, $H(z)$ 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 $H_1(z) = 2z + 0.001z^2$. It intersects $H(z)$ at $Z_0 \approx 0.95$. Select $Z_1 < Z_0 < Z_2$. Let

$$q_0 = H_1(Z_1), q_1 = H_1'(Z_1),$$
$$q_2 = \frac{3[H(Z_2) - H_1(Z_1)]}{(Z_2 - Z_1)^2} - \frac{H'(Z_2) + 2H_1'(Z_1)}{Z_2 - Z_1},$$
$$q_3 = \frac{H(Z_2) - q_2(Z_2 - Z_1)^2 - q_1(Z_2 - Z_1) - q_0}{(Z_2 - Z_1)^3},$$
$$H_2(z) = \begin{cases} H_1(z), 0 \leq z \leq Z_1 \\ H(z), Z_2 \leq z < 1 \\ q_0 + q_1(z - Z_1) + q_2(z - Z_1)^2 + \\ q_3(z - Z_1)^3, Z_1 \leq z \leq Z_2 \end{cases} \quad (5)$$

In [23], we proved that the following lemma holds.
Lemma 2: (1) $H_2(z)$ is strictly convex and differentiable;
(2) If we choose $Z_1 = Z_0 - 0.0015, Z_2 = Z_0 + 0.0010$, then $0.925 \leq \frac{H(z)}{H_2(z)} \leq 1.26$;

Therefore, we can use $H_2(z)$ as a *convex and differentiable* approximation to $H(z)$ (see Fig. 3). Now we can obtain an approximate solution to (B1). Consider the following problem (B2):

(B2) Minimize $I_2(\overrightarrow{z}) = \sum_{i=1}^{M} A_i H_2(z_i) + B_i z_i$
such that $\sum_{i=1}^{M} z_i U_i \geq (1-r)\sum_{i=1}^{M} U_i,$
$p_i \leq z_i \leq 1, i = 1\ldots M.$

Note that the only difference between problem (B1) and (B2) is that $H(z)$ is replaced by its approximation $H_2(z)$. The following proposition shows that the solution of (B2) is an approximate solution of (B1).

Proposition 3: Let $\overrightarrow{z^*}$ be the solution to (B1), $\overrightarrow{\tilde{z}^*}$ be the solution to (B2), then $I_1(\overrightarrow{\tilde{z}^*}) \leq 1.37 I_1(\overrightarrow{z^*})$

Proof: From Lemma 2(2), $0.925 \leq \frac{H(z)}{H_2(z)} \leq 1.26$. Hence,

$$0.925 \leq \frac{I_1(\overrightarrow{z})}{I_2(\overrightarrow{z})} \leq 1.26. \qquad (6)$$

Therefore,

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

where the first and third "$\leq$" come from Equation (6), and the second "$\leq$" holds since $\overrightarrow{\tilde{z}^*}$ is the optimal solution of (B2). ∎

Proposition 3 is important as it shows that $\overrightarrow{\tilde{z}^*}$ 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 $\overrightarrow{z^*}$. However, (B2) is a convex optimization problem and its optimal solution, $\overrightarrow{\tilde{z}^*}$, can be easily obtained using conventional optimization techniques such as the Logarithmic Barrier method [22]. Thus, in our approximation scheme, we first solve (B2) and obtain $\overrightarrow{\tilde{z}^*}$, then use them as the capture probability threshold(s). This may not result in minimum energy consumption, but from Proposition 3, the energy consumption using $\overrightarrow{\tilde{z}^*}$ is no more than 37% higher than the minimum.

V. Conclusions and Future Work

In this paper, we have studied sleep/wake scheduling for low duty cycle sensor networks. The distinguishing factor of our work from most previous work is that we explicitly consider the effect of synchronization error in the design of the sleep/wake scheduling algorithm. Our work includes two parts. In the first part, we study how to decide the sleep/wake schedule to achieve a *given* constraint on the message capture probability with minimum energy consumption. This optimization problem is non-convex and cannot be directly solved by conventional convex optimization techniques. By investigating its unique structure, we transform the non-convex problem into a convex equivalent, and solve it using an efficient search method.

In the second part, we remove the assumption that the capture probability threshold is *already given*, and study how to decide 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 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, and the collective performance is guaranteed. The problem turns out to be non-convex and hard to solve exactly. By investigating its unique structure, we have obtained a suboptimal solution with approximation ratio 1.37. We have validated the effectiveness of our solution via simulations (refer to the appendix). In our future work, we plan to investigate the sleep/wake scheduling of cluster heads that participate in relaying messages from other cluster heads.

References

[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *Proc. of ACM WSNA*, September 2002.

[2] 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.

[3] 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.

[4] W. Ye, J. Heidenmann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proc. of IEEE INFOCOM*, New York, NY, June 2002.

[5] T. Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proc. of ACM SenSys*, 2003.

[6] 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.

[7] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *Proc. of ACM SenSys*, 2004.

[8] G. Pei and C. Chien, "Low power TDMA in Large Wireless Sensor Networks," in *Proc. of IEEE MILCOM*, October 2001.

[9] S. Coleri, A. Puri, and P. Varaiya, "Power Efficient system for Sensor Networks," in *Proc. of IEEE ISCC*, July 2003.

[10] R. Kannan, R. Kalidindi, S. Iyengar, and V. Kumar, "Energy and Rate based MAC Protocol for Wireless Sensor Networks," in *SIGMOD Record*, December 2003.

[11] M. L. Sichitiu, "Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks," in *Proc. of IEEE INFOCOM*, March 2004.

[12] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time synchronization Using Reference Broadcasts," in *Proc. of USENIX/ACM OSDI*, 2002.

[13] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timing-sync Protocol for Sensor Networks," in *Proc. of ACM SenSys*, 2003.

[14] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," in *Proc. of ACM SenSys*, 2004.

[15] 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.

[16] 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.

[17] J. Elson and K. Romer, "Wireless Sensor Networks: A new Regime for Time Synchronization," in *Proc. of HotNets-I*, October 2002.

[18] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. B. Srivastava, "Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks," in *Proc. of ACM SenSys*, 2005.

[19] S. PalChaudhuri, A. K. Saha, and D. B. Johnson, "Adaptive clock synchronization in sensor networks," in *Proc. of IEEE/ACM IPSN*, 2004.

[20] 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.

[21] J. L. Devore, *Probability and Statistics for Engineering and the Sciences*. International Thomson Publishing Inc., 1995.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[23] 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.

[24] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*. John Wiley & Sons, Inc., 2001.

[25] R. Iyer and L. Kleinrock, "QoS Control for Sensor Networks," in *Proc. of IEEE ICC*, 2003.

[26] J.Kay and J.Frolik, "QoS Analysis and Control in Wireless Sensor Networks," in *Proc. of IEEE MASS*, 2004.

[27] 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.

APPENDIX

In this appendix, we present our numerical results. 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 a utility value of $V1$, while the other half have more capabilities and thus have a utility value of $V2 > V1$. We set $p_i = P = 0.1, i = 1 \ldots M$. Other simulation parameters are specified in Table I.

TABLE I

SIMULATION PARAMETERS AND SYSTEM CONSTANTS

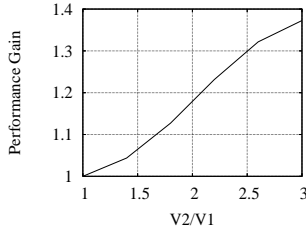|	
Receiving power α_r/Idle power α_I (mW)	13
Data rate R (kbps)	19.2
Message length L_p (byte)	8
Number of cluster member nodes M	10
Epoch duration T_e (minute)	20
Synchronization interval T_s (second)	60
Number of synchronization messages N_s	2
σ_0 (μs)	36.5
Transmission period T (second)	60

Fig. 4. Performance gain over the uniform assignment scheme

We compare our approximation scheme with the previously used uniform assignment scheme, i.e., the scheme with $z_i = 1 - r, i = 1 \ldots M$. In Fig. 4, we vary the value of $\frac{V2}{V1}$ 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 $\frac{V2}{V1}$. This is because the performance gain of our scheme over the uniform assignment scheme stems from 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 ($\frac{V2}{V1} = 1$), there is no benefit in treating the nodes differently; as $\frac{V2}{V1}$ increases, the difference between nodes becomes larger, which makes it advantageous to provide differentiated services to nodes and favor the important ones.

In Fig. 5, we keep the value of $\frac{V2}{V1}$ fixed at 3, and vary the value of P (we still use $p_i = P, \forall i = 1 \ldots M$). 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 improves. At the same time, the region $z_i \geq p_i$ shrinks, which means that we have less flexibility in choosing z_i. Consequently, the performance gain becomes less significant.
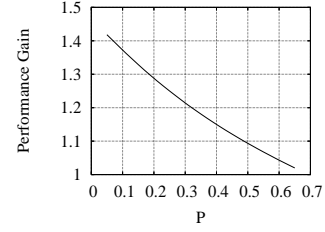

Fig. 5. Tradeoff between reliability and energy savings

Hence, the choice of P controls a tradeoff between system reliability and energy savings.