Sleep/Wake Scheduling for Multi-hop Sensor Networks: Non-convexity and Approximation Algorithm☆

Yan Wu, Sonia Fahmy, Ness B. Shroff☆☆
E-mail: yanwu@microsoft.com, fahmy@cs.purdue.edu, shroff@ece.osu.edu

Abstract

We investigate the problem of sleep/wake scheduling for low duty cycle sensor networks. Our work differs from prior work in that we explicitly consider the effect of synchronization error in the design of the sleep/wake scheduling algorithm. In our previous work, we studied sleep/wake scheduling for *single hop* communication, e.g., intra-cluster communication between a cluster head and cluster members. We showed that there is an inherent trade-off between energy consumption and message delivery performance (defined as the message capture probability). We proposed an optimal sleep/wake scheduling algorithm, which satisfies a given message capture probability threshold with minimum energy consumption.

In this work, we consider multi-hop communication. We remove the previous assumption that the capture probability threshold is *already given*, and study how to decide the per-hop capture probability thresholds to meet the Quality of Services (QoS) requirements of the application. In many sensor network applications, the QoS is decided by the amount of data delivered to the base station(s), i.e., the multi-hop delivery performance. We formulate an optimization problem to set the capture probability threshold at each hop such that the network lifetime is maximized, while the multi-hop delivery performance is guaranteed. The problem turns out to be non-convex and hence cannot be efficiently solved using

☆A short version of this paper appeared in Proc. of INFOCOM 2007. This submission includes a more detailed discussion of the system model (Section 3), reclustering (Section 4.4), and more complete proofs.

☆☆Yan Wu is with Microsoft Corporation. Sonia Fahmy is with the Department of Computer Science, Purdue University. Ness B. Shroff is with the departments of ECE and CSE, The Ohio State University. Please send all correspondence to: Sonia Fahmy, 305 N. University St., West Lafayette, IN 47907–2107, USA. Tel: +1 (765) 494-6183; Fax: +1 765-494-0739. This paper has been partially supported by NSF projects 0238294, 0721434, and 0721236, and ARO MURI W911NF-07-10376 (SA08-03).

standard methods. By investigating the unique structure of the problem and using approximation techniques, we obtain a solution that provably achieves at least 0.73 of the optimal performance. Our solution is extremely simple to implement.

Key words: Sensor networks, Time synchronization, Sleep/wake scheduling, Energy-Efficiency

---

1. Introduction

An important class of wireless sensor network applications is the class of continuous monitoring applications. These applications employ a large number of sensor nodes for continuous sensing and data gathering. Each sensor *periodically* produces a small amount of data and reports to one (or several) base station(s). This application class includes many typical sensor network applications such as habitat monitoring [1] and civil structure monitoring [2].

Measurements show that idle listening consumes a significant amount of energy for sensor devices. An effective approach to conserve energy is to put the radio to sleep during idle times and wake it right before message transmission/reception. This requires precise synchronization between the sender and the receiver, so that they can wake up simultaneously to communicate. The state-of-the-art in sleep/wake scheduling assumes that the underlying synchronization protocol can provide nearly perfect (e.g., μs level) synchronization, so that clock disagreement can be ignored. However, in our previous work [3], we determined that the impact of synchronization error is non-negligible. We found that although existing synchronization schemes achieve precise synchronization *immediately after* the exchange of synchronization messages, there is still random synchronization error because of non-deterministic factors in the system. Thus, clock disagreement grows with time and can be comparable to the actual message transmission time. This means that the design of an effective sleep/wake scheduling algorithm must consider the impact of synchronization error. We demonstrated the inherent trade-off between energy consumption and message delivery performance (defined as the message capture probability). We then proposed an optimal sleep/wake scheduling algorithm, which achieves a message capture probability threshold (assumed to be *given*) with minimum energy consumption.

Our previous work focused on single-hop communications. In this paper, we consider *multi-hop* communication. For illustration, we consider a network that has been *hierarchically clustered*. We remove the assumption that the capture probability threshold is given, and study how to decide the per-hop capture probability thresholds to meet the Quality of Service (QoS) requirement of the application. In many applications, sensor nodes gather data and report to a base station(s) (BS). Therefore, the QoS is decided by the amount of data delivered from the nodes to the BS. We formulate an optimization problem which aims to set the capture probability threshold at each hop such that the network lifetime is maximized, while a minimum fraction of data is guaranteed to be delivered to the BS. The problem turns out to be non-convex and hard to solve exactly, but we design an 0.73-approximation algorithm that can be easily implemented in sensor networks.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 gives the system model and briefly describes our sleep/wake scheduling algorithm for single hop communications. Section 4 studies how to assign the thresholds along multi-hop paths in the cluster hierarchy. Section 5 concludes the paper.

2. Background and Related Work

Sleep/wake scheduling has been extensively studied, e.g., [4, 5, 6]. The basic idea is to let the radio sleep during idle times, and wake it up right before message transmission/reception. Measurements show that this can effectively prevent energy waste caused by overhearing, collisions, and idle listening.

Clustering is generally considered to be a scalable method to manage large sensor networks. Sensors within a geographical region are grouped into a cluster. The sensors are then locally managed 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 BS or other cluster heads. This grouping process can be recursively applied to build a cluster hierarchy. Sensor nodes first elect level-1 CHs, then level-1 CHs elect a subset of themselves as level-2 CHs. Cluster heads at levels $3, 4, \ldots$ are elected in a similar fashion to generate a hierarchy of CHs, in which any level-i CH is also a CH of level $(i-1), (i-2), \ldots, 1$. Fig. 1 depicts nodes organized in a three-level cluster hierarchy with each number representing the level of the corresponding node.

Hierarchical clustering provides a convenient framework for resource management and local decision making. It can also be extremely effective for data fusion, i.e., sensing data can be aggregated before being passed onto the next higher level in the hierarchy. Hence, hierarchical clustering is used in many practical systems [2, 7, 8]. Due to this widespread use, in this work we choose the cluster hierarchy model as an illustrative example. We assume that the network has been hierarchically clustered using one of the popular clustering techniques [9, 10].

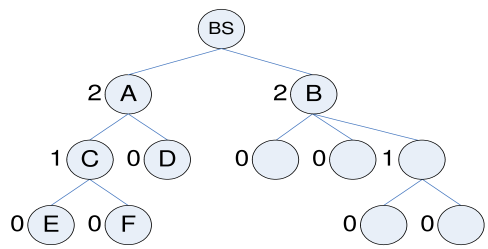

Figure 1: A three-level cluster hierarchy

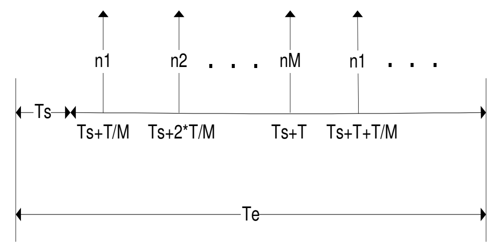

Figure 2: Equispaced upstream transmissions

3. System Model

We consider a cluster hierarchy, where each cluster consists of a single cluster head (CH) and multiple cluster members. Note that a node can be both the CH in one cluster, and a member in another cluster at a higher level, e.g., in Fig. 1, C is the CH of E, but is also a member of A. Time is divided into recurring *epochs* with constant duration T_e. As in many MAC protocols for sensor networks [5, 6], each epoch begins with a synchronization interval T_s followed by a transmission interval (Fig. 2). During the synchronization interval, the cluster members synchronize with their CH and no transmissions are allowed. During the transmission interval, each member node transmits in a TDMA manner and sends one message to the CH every T seconds. The message consists of the aggregate of its own sensing data, and the data collected from its members if the node itself is a CH. 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., if M is the number of cluster members, then transmissions are separated by $\frac{T}{M}$.[1]

3.1. Assumptions

We make the following assumptions about our system:

(1) Orthogonal Frequency Channels: We assume that neighboring clusters use orthogonal frequency bands and do not interfere with each other. This is a reasonable assumption since the data rate of sensor networks is usually low, typically

[1]We summarize all the symbols used in Tables 2 and 3 in the appendix.

around 10−40 kbps. If we run the network in ISM-900 bands (902−928 MHz), then there are more than a thousand frequency channels to choose from.

A node that is both a CH and cluster member needs to communicate with its members and with its CH, e.g., in Fig. 1 node C needs to communicate with both A and E. However, A and E are in neighboring clusters; hence they use different frequency channels. Since every node has only one radio interface, C has to schedule carefully to participate in each cluster. This can be achieved in the following manner. The BS first decides the schedule of the synchronization interval and the transmission schedule for its cluster members (A and B in Fig. 1), then broadcasts this information to the members. A and B, upon hearing the broadcast, will reserve the relevant times for synchronizing/communicating with the root. Then, A and B schedule the synchronization and transmissions for their members at different times. Similarly, C will reserve the times to synchronize/communicate with A, and choose different times for its members (E and F) to synchronize and transmit.

(2) Data aggregation: We adopt a data aggregation model similar to [11]. Consider a cluster with node 0 being the CH, and with M members, $i = 1, \ldots, M$. The length of messages from node i is L_i, $i = 0, \ldots, M$. Thus, the length of the aggregated message is a function of $L_i, i = 0, \ldots, M$. We use the following model for $\chi(L_0, \ldots L_M)$, the length of the aggregated message,

$$\chi(L_0, \ldots L_M) = r \sum_{i=0}^{M} L_i + c. \tag{1}$$

In this model, c corresponds to the overhead of aggregation, while $r \leq 1$ is the compression ratio. Note that r can be 0, in which case Eq. (1) corresponds to the case when all messages can be combined into a single message of fixed length. This models those applications where we want updates of type min, max, and sum (e.g., event count).

The model in Eq. (1) assumes the same compression ratio for messages from different nodes. However, it can be extended to account for different compression ratios, e.g.,

$$\chi(L_0, \ldots L_M) = \sum_{i=0}^{M} r_i L_i + c, \tag{2}$$

where r_i corresponds to the compression ratio for messages from node i. For simplicity in writing, we will use the model in Eq. (1) for the remainder of this paper. However, all the results can be directly extended for the model in Eq. (2).

(3) Radio hardware: We assume that the sender can precisely control when the message is sent out onto the channel using its *own* clock. This is reasonable since in [12], system measurements have shown that non-determinism at the sender is negligible compared to non-determinism at the receiver.

For the receiver, we assume that if there is an incoming message, it can immediately detect the radio signal. This is a close approximation of the real situation, since modern transceivers can detect incoming signals within microseconds [13]. Further, we assume that once the receiver detects an incoming message, it will stay active until the reception is completed.

(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 within a few clock cycles [14, 15]. Thus, we consider the transition time to be negligible.

(5) Collisions: We assume that the separation between transmissions from different members, $\frac{T}{M}$, for a cluster with M members is large enough so that the collision probability for transmissions from different members is negligible. This is a reasonable assumption for low duty cycle sensor networks. Consider a large cluster of $M = 50$ members and each member transmits to the CH every $T = 60$ seconds. The separation is $\frac{T}{M} = 1200\ ms$. For low duty cycle networks, the message size is usually not large; hence the transmission time is much smaller than this separation. Moreover, at the beginning of each epoch, the cluster members re-synchronize with the CH, so that the clock disagreement will not become large enough to cause significant collision probability.

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

(7) Clock skew: Vig [16] 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 [17].

3.2. Synchronization Algorithm

Time synchronization for wireless sensor networks has been extensively investigated [18, 19, 12, 20, 17]. Clock disagreement between sensor nodes can be characterized using two factors: *phase offset* and *clock skew*. Phase offset corresponds to clock disagreement between nodes at a given instant. Clock skew means clocks run at different speeds, i.e., the actual frequency deviates from the expected frequency. This is due to manufacturing imprecision and aging effects. The maximum clock skew is less than than 100 ppm and is usually specified by the manufacturer. Besides manufacturing imprecision and aging, the frequency is also affected by environmental factors including temperature, pressure, and voltage [16]. Among these factors, temperature has the most significant effect. When temperature significantly changes, the variation in the clock frequency 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 seconds 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 empirical observations in [17].

In this work, we adopt the well-known RBS synchronization scheme, and study the sleep/wake scheduling problem[2]. The scheme includes 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.

At the beginning of each epoch j, the cluster members need to synchronize with the CH. Towards this end, each cluster member i exchanges several synchronization messages with the CH and obtains 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 k^{th} time instant of the CH and of node i in epoch j respectively.

Under the assumption that the clock skew of each node does not change over the epoch, during a given epoch j the clock time of member node i, t_i, is a linear

[2]This scheme is chosen for illustration purposes only. Our sleep/wake scheduling solution works with most synchronization schemes.

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 member node i and CH in epoch j.

Because of the non-determinism in the message exchange, the obtained time correspondence is not exactly accurate and contains an error, i.e.,

$$t_i(j,k) = a_i(j)C(j,k) + b_i(j) + e_i(j,k), \tag{3}$$

where $e_i(j,k)$ is the random error caused by non-determinism in the system. Real system measurements [19] show with a high confidence level 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.

At each epoch j, pairs $(C(j,k), t_i(j,k)), k = 1, \ldots, N_s$ are obtained via exchange of synchronization messages. Then, linear regression is performed on these N_s pairs to obtain estimates of $a_i(j), b_i(j)$, denoted by $\hat{a}_i(j), \hat{b}_i(j)$. In this work, we control the exchange of synchronization messages such that $C(j,k) \approx jT_e + k\frac{T_s}{N_s}, k = 1, \ldots, N_s$. This is achieved by letting the CH initiate the message exchange, i.e., the CH selects a member as the beacon node and tells it to broadcast the beacons at $jT_e + k\frac{T_s}{N_s}, k = 1, \ldots, N_s$ according to the CH clock. Due to system uncertainty and clock skew, the beacon node may not broadcast exactly at the desired time instants. But considering the fact that usually the synchronization interval is short compared to the whole epoch duration, the deviation is small. In this manner, the beacons are broadcasted approximately at $jT_e + k\frac{T_s}{N_s}, k = 1, \ldots, N_s$ according to the CH clock.

3.3. The Optimal Sleep/Wake Scheduling Problem

This work leverages our previous work on sleep/wake scheduling for single hop intra-cluster communications [3]. For brevity, here we only give the equations that will be used in the remainder of the paper. Interested readers can refer to [3] for details. In [3], the original problem formulation is given as:

(A) Min $E = (s_p - w_p)\alpha_I Prob\{\tau_p' \notin (w_p, s_p)\} + \int_{w_p}^{s_p} \{(x - w_p)\alpha_I + \frac{L_p}{R}\alpha_r\} f_{\tau_p'}(x)dx$ such that $Prob\{\tau_p' \in (w_p, s_p)\} \geq th$,

After a number of transformations, formulation (A) is turned into:

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

We can see that the minimum expected energy to receive the message is

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

where

$$\gamma(th) \;=\; \min\{G(w) : w < Q^{-1}(th)\} \tag{5}$$

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

When solving (A3), we proved the following proposition:

Proposition 1. *(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.

Finally, we will also use the following Equation (3) from [3] later in this paper:

$$\begin{aligned} E(\tau_p') &= \tau_p, \\ VAR(\tau_p') \equiv \sigma_p^2 &= \frac{\sigma_0^2}{a_i^2(j)} \frac{1}{N_s}[1 + \frac{(\tau_p - \overline{C(j,k)})^2}{\overline{C^2(j,k)} - (\overline{C(j,k)})^2}], \end{aligned} \tag{6}$$

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

4. The Capture Probability Threshold Assignment Problem

We now study how to decide the capture probability threshold to meet the QoS requirement of the application and maximize the network lifetime.

4.1. Problem Definition

Consider a sensor network deployed for environmental monitoring. The network consists of a set of sensor nodes and one or more base stations (BSs), usually personal computers. The network has already been hierarchically clustered using one of these clustering techniques [9, 10]. We assume there is a single BS, denoted by BS. The formulation can be easily extended to the case with multiple BSs. $H(n)$ denotes the cluster head of node n. $M(n)$ denotes the set of nodes that are members of n. $D(n)$ denotes the set of nodes that are the descendants of n. $M(n)$ and $D(n)$ can be empty if node n is at level 0. $d(n)$ is the hop distance from node n to BS, i.e., $H^{(d(n))}(n) \equiv \underbrace{H(H(\ldots H(n)\ldots))}_{d(n)} = BS$.

Each sensor node periodically reports to its CH. The CH aggregates its own sensing data and the data collected from the members over the last transmission period, then forwards the aggregated data to its CH. The process continues until the message finally arrives at BS. Each message contains some sensing data and represents certain amount of "information" about the environment. BS uses the collected information to compute certain properties, e.g., the chemical contaminant in the area. The service quality is defined as the accuracy of the computed properties, which is decided by the amount of information collected by BS, i.e., the more information collected, the better accuracy. Hence, the service quality is not decided by the delivery performance at any particular hop, but by the *multi-hop delivery performance* from the nodes to BS.

However, collecting more information requires higher energy consumption and may lead to widely varying power dissipation levels across nodes, e.g., nodes at high levels in the cluster hierarchy have an excessive relaying burden. This will result in a shorter lifetime for some nodes, which can lead to loss of coverage when these nodes deplete their energy. This is the inherent trade-off between application performance and network lifetime. To maximize the network lifetime and still guarantee the application performance, we formulate the following optimization problem.

We define the network lifetime T_L as the time until the death of the first sensor node. This definition is widely used in the literature [4, 21, 22, 9, 23]. It mainly applies to application scenarios with strict coverage requirements, where each sensor "covers" a certain area in the environment and provides equally important information to BS. To maintain complete coverage and save redeployment cost, we must ensure that all the nodes remain up for as long as possible[3].

Let $z(n)$ be the capture probability threshold of $H(n)$ for messages coming from n, i.e., node $H(n)$ will capture messages from node n with probability no less than $z(n)$. The goal is to choose $z(n)$ to maximize the network lifetime, and still guarantee that all information be delivered to BS with a predefined probability Λ:

(B) Max T_L
such that $\prod_{i=0}^{d(n)-1} z(H^{(i)}(n)) \geq \Lambda, \forall n \in S,$

where Λ is decided by the QoS requirement of the application.

For the data from node n to be received by BS, it needs to pass through

[3]Here, we assume that we will lose the corresponding coverage if a node dies, i.e., there is no redundant node. If the network has redundancy, we can consider the nodes covering the same area (e.g., nodes near the same bird nest) as a single node whose initial energy equals the sum of energy of all the relevant nodes, and then this defi nition and the following results still apply.

$H(n), H^{(2)}(n), \ldots, H^{(d(n)-1)}(n)$. Hence in (B), the constraint $\prod_{i=0}^{d(n)-1} z(H^{(i)}(n)) \geq \Lambda$ means the data from n will be received by BS with probability no less than Λ. Note that the data will be aggregated with data from other nodes at each hop along the path.

4.2. Solution

We solve Problem (B) in this section. We first obtain an explicit form of Problem (B), then show that it is a non-convex optimization problem. The non-convexity makes it hard to solve Problem (B) exactly. Hence, we investigate the structure of the problem and obtain an approximate solution.

In the cluster hierarchy, if the multi-hop delivery performance of a leaf node (a level-0 node) is guaranteed, then the delivery performance for its ancestors is guaranteed as well, i.e., if the information from a leaf node n is delivered to BS with probability no less than Λ, then the information from $H(n), H^{(2)}(n) \ldots H^{(d(n)-1)}(n)$ will also be delivered with probability no less than Λ. Hence, in (B), the constraints on the delivery performance of non-leaf nodes are redundant and can be removed. Let LF denote the set of leaf nodes. We obtain the following formulation:

(B) Max T_L
such that $\prod_{i=0}^{d(n)-1} z(H^{(i)}(n)) \geq \Lambda, \forall n \in LF,$

To obtain an explicit form of Problem (B), we characterize the average power dissipation for each sensor node when $z(m), m \in S$ are given. During an epoch, a node n consumes energy for sensing, synchronization, and transmitting/receiving data messages. Let the sensing energy and synchronization energy be $\varepsilon_s(n)$, and $\varepsilon_{syn}(n)$ respectively. These do not depend on the capture probability thresholds.

Both the transmission energy and the receiving energy depend on the capture probability thresholds. Let l be the amount of sensing data generated by each sensor during each transmission period T, and $L^{avg}(n)$ be the *average* message size from n. Then, from the aggregation model in Eq. (1),

$$L^{avg}(n) = r(l + \sum_{i \in M(n)} z(i) L^{avg}(i)) + c.$$

Recursively applying the above formula, we have

$$L^{avg}(n) = rl + c + \sum_{i \in D(n)} (rl + c) \prod_{k=0}^{d(i)-d(n)-1} [rz(H^{(k)}(i))]. \tag{7}$$

Since N messages are transmitted in each epoch, the average transmission energy in an epoch is

$$\varepsilon_t(n) = N\alpha_t(n)\frac{L^{avg}(n)}{R}, \tag{8}$$

where $\alpha_t(n)$ is the transmission power of node n[4].

We now compute the average receiving energy $\varepsilon_r(n)$. For a node n with $|M(n)|$ members, during a given epoch j, these nodes transmit to n in turn. To decide the transmission sequence, node n orders the $|M(n)|$ members, i.e., each member node $i \in M(n)$ is assigned a sequence number $\theta(i)$ from $\{1, 2, \ldots, |M(n)|\}$, and different member nodes have different sequence numbers. Node i is scheduled to transmit at $jT_e + T_s + \theta(i)\frac{T}{|M(n)|} + hT, h = 1, \ldots, N$. For given capture probability thresholds, node n will use the sleep/wake schedule described in Section 3.3, as it is the *optimal sleep/wake schedule*. Therefore, the average energy used to receive a message scheduled to arrive at τ_p is exactly the minimum value of the objective function in Problem (A), which is (by Eq. (4))

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

Here, L_p is the message size, σ_p is computed from Eq. (6), th is the required threshold, and $\gamma(th)$ is as given in Eq. (5). The average receiving energy $\varepsilon_r(n)$ can be computed by summing up the energy used to receive all messages from its members. As in Section 3.2, the synchronization is controlled such that $C(j,k) \approx jT_e + k\frac{T_s}{N_s}$, so

$$\begin{aligned} \overline{C(j,k)} &\approx jT_e + \frac{1+N_s}{2}\frac{T_s}{N_s}, \\ \overline{C^2(j,k)} - (\overline{C(j,k)})^2 &\approx \frac{\sum_{k=1}^{N_s}(k\frac{T_s}{N_s} - \frac{1+N_s}{2N_s}T_s)^2}{N_s}. \end{aligned} \tag{9}$$

Further, recall that the maximum clock skew is no larger than 100 ppm; hence in

[4]We assume that each node has a fixed number of transmission power levels (as in Mica2 motes), and can choose the appropriate one based upon factors such as distance and channel fading.

Eq. (6), the relative clock skew $a_i(j) \approx 1$. Combining these together, we have

$$\varepsilon_r(n) \approx \sum_{i \in M(n)} \sum_{h=1}^{N} \alpha_r z(i) \frac{L^{avg}(i)}{R} + \alpha_I \gamma(z(i)) \sqrt{\sigma_0^2 \frac{1}{N_s}[1 + \frac{(T_s + \frac{\theta(i)T}{|M(n)|} + hT - \frac{1+N_s}{2}\frac{T_s}{N_s})^2}{\frac{\sum_{k=1}^{N_s}(k\frac{T_s}{N_s} - \frac{1+N_s}{2N_s}T_s)^2}{N_s}}]}. \tag{10}$$

For node n, the average energy consumption in an epoch is the sum of the sensing energy, the synchronization energy, and the transmission/reception energy. Combining Eq. (7), (8) and (10), the average power dissipated in node n is given by

$$\begin{aligned} \eta(n, \overrightarrow{z}) &= \frac{\varepsilon_s(n) + \varepsilon_{syn}(n) + \varepsilon_t(n) + \varepsilon_r(n)}{T_e} \\ &= A(n) + \sum_{i \in M(n)} P(n,i)\gamma(z(i)) + \\ &\quad \sum_{i \in D(n)} Q(n,i) \prod_{k=0}^{d(i)-d(n)-1} z(H^{(k)}(i)), \end{aligned} \tag{11}$$

where

$$A(n) = \frac{1}{T_e}[\varepsilon_s(n) + \varepsilon_{syn}(n) + N\alpha_t(n)\frac{rl+c}{R}],$$
$$P(n,i) = \frac{1}{T_e}\sum_{h=1}^{N} \alpha_I \sqrt{\sigma_0^2 \frac{1}{N_s}[1 + \frac{(T_s + \frac{\theta(i)T}{|M(n)|} + hT - \frac{1+N_s}{2}\frac{T_s}{N_s})^2}{\frac{\sum_{k=1}^{N_s}(k\frac{T_s}{N_s} - \frac{1+N_s}{2N_s}T_s)^2}{N_s}}]},$$
$$Q(n,i) = \frac{1}{T_e}\frac{rN\alpha_t(n) + N\alpha_r}{R}(rl+c)r^{d(i)-d(n)-1}.$$

Let $\xi(n)$ be the initial energy of node n, then Problem (B) can be written as
Max T_L
such that $\prod_{i=0}^{d(n)-1} z(H^{(i)}(n)) \geq \Lambda, \forall n \in LF,$
$\eta(n, \overrightarrow{z}) \leq \xi(n)/T_L, \forall n \in S.$

Next, we introduce a lifetime-penalty function $\Psi(1/T_L)$ to be a strictly convex and increasing function (e.g., $\Psi(x) = x^2$). Then, maximizing the network lifetime is equivalent to minimizing the lifetime-penalty function. We now use a change of variable $u = 1/T_L$ to give the network lifetime maximization problem as the following equivalent problem:

(B) Min $\Psi(u)$
such that $\prod_{i=0}^{d(n)-1} z(H^{(i)}(n)) \geq \Lambda, \forall n \in LF,$
$\eta(n, \overrightarrow{z}) \leq \xi(n)u, \forall n \in S.$

The difficulty in solving (B) is that it is not a convex optimization problem. To see this, we observe that in the second set of constraints, the left side $\eta(n, \overrightarrow{z})$ includes $\gamma(z(i))$ and $\prod z(i)$. $\prod z(i)$ may not be convex, e.g., $z(1)z(2)$; for $\gamma(z(i))$, we numerically show the curve in Fig. 3 which is clearly not convex. Hence, the constraint region is not a convex set, and Problem (B) is not convex. Further, we do not have an explicit analytical form for $\gamma(z)$. This makes Problem (B) hard to solve exactly. Next, we investigate the structure of the problem and obtain an approximate solution.

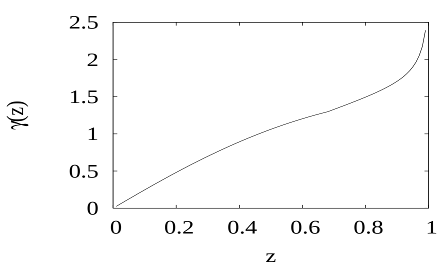

Figure 3: $\gamma(z)$

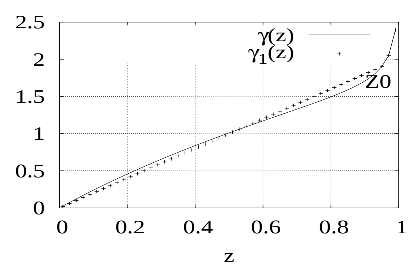

Figure 4: Approximating $\gamma(z)$

The following proposition characterizes $\gamma(z)$.

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

We give the proof in the appendix. The idea is that although we do not have an explicit analytical form of $\gamma(z)$, we have the bounds obtained from Proposition 1(2). Therefore, we compute $\gamma'(z), \gamma''(z)$ using implicit differentiation and bound them. This proposition shows that $\gamma(z)$ is convex in the region $[0.86, 1)$; for the remaining region where $\gamma(z)$ may not be convex, we can bound it fairly tightly.

Next, we approximate $\gamma(z)$ with a convex function. The curve $2z + 0.001z^2$ intersects $\gamma(z)$ at $Z_0 \approx 0.95$. Let

$$\gamma_1(z) = \begin{cases} 2z + 0.001z^2 & 0 \le z \le Z_0 \\ \gamma(z) & Z_0 \le z < 1 \end{cases}$$

The following proposition shows that $\gamma_1(z)$ is a *convex* approximation of $\gamma(z)$.

Proposition 3. *(1)* $0.929 \le \gamma(z)/\gamma_1(z) \le 1.26$*;*
(2) $\gamma_1(z)$ *is strictly convex.*

This proposition can be proven using Proposition 2 (see appendix). Fig. 4 illustrates that $\gamma_1(z)$ is a good approximation of $\gamma(z)$. Now, we can obtain an approximate solution of (B). Consider the following problem (B1):

(B1) Min $\Psi(u)$
such that $\prod_{i=0}^{d(n)-1} z(H^{(i)}(n)) \ge \Lambda, \forall n \in LF$,

$$\eta_1(n, \overrightarrow{z}) = A(n) + \sum_{i \in M(n)} P(n,i)\gamma_1(z(i)) + \sum_{i \in D(n)} Q(n,i) \prod_{k=0}^{d(i)-d(n)-1} z(H^{(k)}(i)) \le \xi(n)u, \forall n \in S.$$

The only difference between (B) and (B1) is that in (B1), $\gamma(\cdot)$ is replaced by $\gamma_1(\cdot)$. The following proposition shows that the solution of (B1) is an approximate solution of (B).

Proposition 4. *Let* $(\overrightarrow{z^*}, u^*)$ *be the optimal solution to (B),* $(\overrightarrow{z_1^*}, u_1^*)$ *be the optimal solution to (B1),* $T_L(\overrightarrow{z^*})$ *be the network lifetime when using* $\overrightarrow{z^*}$ *as the capture probability thresholds,* $T_L(\overrightarrow{z_1^*})$ *be the network lifetime when using* $\overrightarrow{z_1^*}$ *as the capture probability thresholds, then* $T_L(\overrightarrow{z_1^*}) \ge 0.73 T_L(\overrightarrow{z^*})$*.*

Proof: From Proposition 3, $0.929 \le \frac{\gamma(z)}{\gamma_1(z)} \le 1.26$. Therefore,

$$0.929 \le \eta(n, \overrightarrow{z})/\eta_1(n, \overrightarrow{z})) \le 1.26. \tag{12}$$

Because $(\overrightarrow{z_1^*}, u_1^*)$ is the optimal solution of (B1), we have

$$\eta_1(n, \overrightarrow{z_1^*}) \le \xi(n)u_1^*, \forall n \in S.$$

Therefore, $\eta(n, \overrightarrow{z_1^*}) \leq 1.26\eta_1(n, \overrightarrow{z_1^*}) \leq 1.26\xi(n)u_1^*, \forall n \in S$. Hence,

$$T_L(\overrightarrow{z_1^*}) \geq 1/(1.26u_1^*). \tag{13}$$

Also, as $(\overrightarrow{z_1^*}, u_1^*)$ is the *optimal* solution of (B1), there must exist some node i such that $\eta_1(i, \overrightarrow{z^*}) \geq \xi(i)u_1^*$. Otherwise if $\eta_1(n, \overrightarrow{z^*}) < \xi(n)u_1^*, \forall n \in S$, then let $u_1^{'} = \max_{n \in S}\{\eta_1(n, \overrightarrow{z^*})/\xi(n)\}$. It can be easily verified that $(\overrightarrow{z^*}, u_1^{'})$ is a solution to (B1) and $u_1^{'} < u_1^*$, which is contradictory to the fact that $(\overrightarrow{z_1^*}, u_1^*)$ is the *optimal* solution of (B1).

For this node i, we have

$$\eta(i, \overrightarrow{z^*}) \geq 0.929\eta_1(i, \overrightarrow{z^*}) \geq 0.929\xi(i)u_1^*,$$

thus $T_L(\overrightarrow{z^*}) \leq 1/(0.929u_1^*)$. Combined with Eq. (13), we have $T_L(\overrightarrow{z_1^*}) \geq 0.73T_L(\overrightarrow{z^*})$. ■

The intuition behind the proof is that $\gamma_1(\cdot)$ approximating $\gamma(\cdot)$ implies $\eta_1(n, \overrightarrow{z}) \approx \eta(n, \overrightarrow{z}), \forall n \in S$. Hence, $T_L(\overrightarrow{z^*}) = \min_{n \in S}\{\xi(n)/\eta(n, \overrightarrow{z^*})\} \approx \min_{n \in S}\{\xi(n)/\eta_1(n, \overrightarrow{z^*})\}$, and $T_L(\overrightarrow{z_1^*}) = \min_{n \in S}\{\xi(n)/\eta(n, \overrightarrow{z_1^*})\} \approx \min_{n \in S}\{\xi(n)/\eta_1(n, \overrightarrow{z_1^*})\}$. But $\overrightarrow{z_1^*}$ is the optimal solution of (B1), so $\min_{n \in S}\{\xi(n)/\eta_1(n, \overrightarrow{z_1^*})\} \geq \min_{n \in S}\{\xi(n)/\eta_1(n, \overrightarrow{z^*})\}$. Therefore, $T_L(\overrightarrow{z_1^*}) \approx \min_{n \in S}\{\xi(n)/\eta_1(n, \overrightarrow{z_1^*})\}$ cannot be much smaller than $T_L(\overrightarrow{z^*}) \approx \min_{n \in S}\{\xi(n)/\eta_1(n, \overrightarrow{z^*})\}$.

Proposition 4 is important as it shows that $\overrightarrow{z_1^*}$ is an approximate solution of (B) with *approximation ratio* 0.73.

As described earlier, (B) is a non-convex optimization problem; hence it is difficult to obtain the optimal solution $\overrightarrow{z^*}$. However, Proposition 4 shows that if we can solve (B1) and use its solution $\overrightarrow{z_1^*}$ as the capture probability thresholds, then the achieved network lifetime is no less than 73% of the maximum. Next we solve (B1).

Using the variable transformation: $v(i) = \ln(z(i))$, problem (B1) becomes the following equivalent problem (B1'):

(B1') Min $\Psi(u)$

such that $\sum_{i=0}^{d(n)-1} v(H^{(i)}(n)) \geq \ln \Lambda, \forall n \in LF$,

$$\eta_1'(n, \overrightarrow{v}) = A(n) + \sum_{i \in M(n)} P(n,i)\gamma_1(e^{v(i)}) +$$

$$\sum_{i \in D(n)} Q(n,i)e^{\sum_{k=0}^{d(i)-d(n)-1} v(H^{(k)}(i))} \leq \xi(n)u, \forall n \in S.$$

In (B1'), obviously the optimization goal function is convex and the first set of constraints corresponds to a convex set. For the second set of constraints, because both $\exp(\cdot)$ and $\gamma_1(\cdot)$ are strictly convex and increasing, from the composition rule [24], $\gamma_1(\exp(\cdot))$ is also strictly convex. Therefore, the second set of constraints also corresponds to a convex set, and (B1') is a convex equivalent of (B1).

We solve (B1') via dual formulation. The dual problem is

$$\max_{\overrightarrow{\lambda}\geq 0, \overrightarrow{\mu}\geq 0} \Phi(\overrightarrow{\lambda}, \overrightarrow{\mu}),$$

where $\overrightarrow{\lambda}, \overrightarrow{\mu}$ are Lagrange multipliers corresponding to the two sets of constraints in (B1'), and $\Phi(\overrightarrow{\lambda}, \overrightarrow{\mu})$ is the dual function given by

$$\Phi(\overrightarrow{\lambda}, \overrightarrow{\mu}) = \min_{u\geq 0, \overrightarrow{v}<0} \Psi(u) + \sum_{n\in LF} \lambda_n(\ln\Lambda - \sum_{i=0}^{d(n)-1} v(H^{(i)}(n))) + \sum_{n\in S}\mu_n(\eta_1'(n, \overrightarrow{v}) - \xi(n)u). \quad (14)$$

We use the subgradient method [24] to solve the dual problem. Let $u^*, \overrightarrow{v^*}$ be the minimizer in Eq. (14). One subgradient of the negative dual function $-\Phi(\overrightarrow{\lambda}, \overrightarrow{\mu})$ is [24]

$$\begin{aligned}
\vartheta_n &= \sum_{i=0}^{d(n)-1} v^*(H^{(i)}(n)) - \ln\Lambda, \forall n \in LF, \\
\varphi_n &= \xi(n)u^* - \eta_1'(n, \overrightarrow{v^*}), \forall n \in S,
\end{aligned}$$

where $\overrightarrow{\vartheta}$ and $\overrightarrow{\varphi}$ correspond to the dual variables $\overrightarrow{\lambda}$ and $\overrightarrow{\mu}$ respectively.

To obtain the optimal dual variables, the subgradient method uses the following updates at the k^{th} iteration

$$\begin{aligned}
\lambda_n(k+1) &= [\lambda_n(k) - \varpi_k\vartheta_n(k)]^+, \forall n \in LF, \\
\mu_n(k+1) &= [\mu_n(k) - \varpi_k\varphi_n(k)]^+, \forall n \in S,
\end{aligned} \quad (15)$$

where $[.]^+$ denotes projection on the nonnegative orthant [5], and ϖ_k is the step size.

[5]Note that in Problem (B), because $\eta(n, \overrightarrow{z})$ increases with $\overrightarrow{z}$, it can be seen that to guarantee a larger delivery probability, higher power is needed and the lifetime will be reduced. Hence, the optimal solution(s) occurs only when the delivery probabilities equal Λ, i.e., when $\prod_{i=0}^{d(n)-1} z(H^{(i)}(n)) = \Lambda, \forall n \in LF$. Therefore, when updating λ_n, the projection $[.]^+$ is unnecessary.

Convergence to the optimal dual variables is guaranteed if ϖ_k satisfies $\varpi_k \rightarrow 0, \sum_{k=1}^{\infty} \varpi_k = \infty$.

Here is a physical interpretation of the dual variables $\overrightarrow{\lambda}$ and $\overrightarrow{\mu}$. Consider $\overrightarrow{\lambda}$ to be the price of violating the requirement on the delivery performance, and $\overrightarrow{\mu}$ to be the price of exceeding the battery capacity. Then, $\overrightarrow{\vartheta}$ represents the safety margin before breaking the performance requirement, and $\overrightarrow{\varphi}$ represents the excess battery capacity. The updates in Eq. (15) will increase the corresponding prices if the performance requirement is violated or the average power dissipation exceeds the capacity, and reduce the prices otherwise.

4.3. Implementation

In many sensor systems [25, 26], the BS is a Pentium-level PC, which has a high computational capability and sufficient memory compared to the sensor nodes. Further, the BS is often connected to an unlimited power supply. Hence, we should take advantage of the capabilities of the BS and let it perform the computations. This scheme is effective because the BS is more powerful than the sensor nodes, and is assumed to have an unlimited power supply. If the BS has similar capabilities to the sensor nodes, a distributed implementation is clearly desirable.

After the cluster hierarchy has been established, the BS informs the nodes of the systems parameters, including the epoch duration T_e, synchronization interval T_s, and message frequency T. Each node then computes $A(n), P(n,i), Q(n,i)$ and reports to the BS. The transmission is hierarchical: the cluster members compute their $A(n), P(n,i), Q(n,i)$ values, and pass them onto the CH, then the CH combines its own parameter values with those of the members and passes onto its own CH. To guarantee that these values are received by the BS, reliable data delivery mechanisms like hop-by-hop acknowledgments can be used.

The BS solves problem (B1) using the subgradient method and computes the capture probability thresholds, then informs the sensor nodes. The nodes decide the wake up schedule as described in Section 3.3.

We note that the computation of the optimal capture probability thresholds is *infrequently* performed, i.e., the capture probability thresholds are computed only once after the cluster hierarchy is constructed. Hence, the message overhead is insignificant in the long run.

4.4. Reclustering

In our discussions thus far, the network topology is fixed at one particular cluster hierarchy. In many systems [9, 23], periodic reclustering is used to balance the load, and the network topology alternates between multiple cluster hierarchies. Thus, we extend the formulation to account for reclustering. Suppose the network alternates between I topologies (cluster hierarchies) and the fraction of time it stays with topology j is $p_j, 1 \leq j \leq I$. The average power dissipation for a node n can be computed as:

(1) The average power dissipation for node n in cluster hierarchy j, $\eta_j(n, \overrightarrow{z_j})$, is computed as in Eq. (11);

(2) The average power dissipation for node n, $\eta(n)$, equals the weighted sum of $\eta_j(n, \overrightarrow{z_j})$ over all $j, 1 \leq j \leq I$:

$$\eta(n) = \sum_{j=1}^{I} p_j \eta_j(n, \overrightarrow{z_j}). \tag{16}$$

The network lifetime maximization problem becomes:

(C) Min $\Psi(u)$

$$\text{s. t.} \prod_{i=0}^{d_j(n)-1} z_j(H_j^{(i)}(n)) \geq \Lambda, \forall n \in LF_j, j = 1, \ldots, I,$$
$$\eta(n) \leq \xi(n)u, \forall n \in S,$$

where $\eta(n)$ is computed as in Eq. (16). The solution of Problem (C) exactly follows that of Problem (B).

4.5. Simulation Results

For illustration, we consider the cluster hierarchy in Fig. 5. The initial energy for all nodes is 1 Joule. Each node will generate $l = 4bytes$ of sensing data during each transmission period. The data aggregation overhead c is 4 $bytes$; the compression ratio $r \in [0, 1]$. We set $\Lambda = 0.7$, i.e., all information should be delivered to the BS with probability ≥ 0.7. Other simulation parameters are specified in Table 1.

For the given topology, we first note that since the BS has unlimited power supply, it can always stay awake. Thus, for messages coming from node 1, the BS will always "capture" them, and we can directly set $z(1) = 1$. Further, due to symmetry, the algorithm should set $z(2) \approx z(3)$ and $z(4) \approx z(5) \approx \ldots z(11)$. Next we consider two special cases:

- $r = 1$ corresponds to the case without any compression. In this case, node 1 is the bottleneck since it has the highest relaying burden. Hence, $z(2)$ and $z(3)$ should be set small such that node 1 spends less energy for receiving.

Table 1: Simulation parameters

Idle power α_I (mW)	13
Receiving power α_r (mW)	13
Data rate R (kbps)	19.2
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

Our algorithm sets $z(2) \approx z(3) \approx 0.71$ and $z(4) \approx \ldots z(11) \approx 0.99$, correctly identifying the bottleneck.

- $r = 0$ corresponds to the case where we want updates of the type min, max, and sum. Here, transmission energy is the same for all the nodes, and the receiving energy decides the lifetime for each node. Thus, nodes 2 and 3 become the bottleneck since they need to receive from more member nodes. Therefore, $z(4) \ldots z(11)$ should be set small to save energy for nodes 2 and 3. Our algorithm achieves this by setting $z(2) \approx z(3) \approx 0.999$ and $z(4) \approx \ldots z(11) \approx 0.703$.

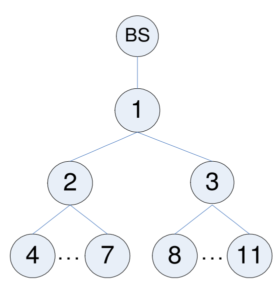

Figure 5: Simulation topology

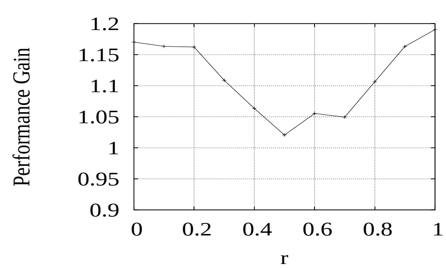

Figure 6: Performance gain

To illustrate the performance gain of our threshold assignment algorithm, we compare with a scheme which sets equal capture probability threshold at each hop along the cluster hierarchy, $z(2) = \ldots = z(11) = \sqrt{\Lambda}$. In Fig. 6, we vary the value of r and show the performance gain, which is defined as the ratio between the network lifetime with the two schemes. We observe that our scheme always outperforms the scheme with equal thresholds. As r increases from 0 to 1, the gain first decreases and then increases. This is because, from the above discussion, when $r = 0$, nodes 2 and 3 are the bottlenecks; hence our scheme sets $z(4), \ldots, z(11)$ to be small and $z(2), z(3)$ to be large. As r increases from 0, node 1 has a higher burden of relaying. To balance the energy consumption, our scheme increases $z(4), \ldots, z(11)$ and decreases $z(2), z(3)$. Consequently, our solution becomes closer to the scheme with equal thresholds. When $r = 0.5$, our solution almost overlaps with the other scheme and the performance gain is relatively small. But as r increases further, our solution diverges from the other scheme and achieves a higher gain, which is as large as 19% when $r = 1$. This confirms that it is necessary to adopt an intelligent scheme to assign the thresholds, and validates the effectiveness of our scheme.

5. Conclusions and Future Work

We have studied sleep/wake scheduling for low duty cycle sensor networks. We explicitly consider the effect of synchronization error in the design of the sleep/wake scheduling algorithm. In our previous work [3], we showed that the impact of synchronization error is non-negligible, even for single hop communications. Our proposed optimal sleep/wake scheduling algorithm achieved a given message capture probability threshold with minimum energy consumption.

In this work, we considered multi-hop communications. We relaxed the assumption that the capture probability threshold is given, and studied how to determine per-hop capture probability thresholds to meet the QoS requirement of the application. QoS in many sensor networks for continuous monitoring applications is decided by the amount of data delivered from the nodes to the base station(s). We formulate an optimization problem that sets the capture probability threshold at each hop such that the network lifetime is maximized, and yet the QoS is guaranteed. The main difficulty we encounter is that the problem turns out to be non-convex. However, by investigating its unique structure, we have obtained a 0.73-approximation algorithm that is simple to implement in practice. We first approximated the minimum value of our objective function, $\gamma(z)$, with $\gamma_1(z)$, and then defined a convex optimization problem (B1) using $\gamma_1(z)$. Next, we proved that the solution of problem (B1) is an approximate solution to problem (B). Finally, we solved problem (B1) using the subgradient method. Simulations show that our solution correctly identifies the bottleneck and significantly extends the network lifetime.

We have fixed the synchronization scheme in this paper, and only focused on energy conservation with sleep/wake scheduling. Synchronization and scheduling are, however, closely tied to each other and will both affect the overall system performance. Therefore, it is necessary to jointly consider synchronization and scheduling to improve the overall system performance. Further, the definition of network lifetime in this work mainly applies to application scenarios with strict coverage requirements. We plan to extend our framework to consider other definitions of network lifetime, e.g., time until network partitioning.

References

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

[2] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, D. Estrin, A Wireless Sensor Network for Structural Monitoring, in: Proc. of ACM SenSys, 2004.

[3] Y. Wu, S. Fahmy, N. B. Shroff, Optimal sleep/wake scheduling for time-synchronized sensor networks with qos guarantees, in: Proc. of IEEE IWQoS, 2006.

[4] S. Singh, C. Raghavendra, PAMAS: Power Aware Multi-Access protocol with Signalling for Ad Hoc Networks, ACM Computer Communication Review 28 (3) (1998) 5–26.

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

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

[7] M. Coates, Evaluating causal relationships in wireless sensor/actuator networks, in: Proc. of IEEE ICASSP, 2005.

[8] B. Hohlt, L. Doherty, E. Brewer, Flexible power scheduling for sensor networks, in: Proc. of IEEE/ACM IPSN, 2004.

[9] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An Application-Specific Protocol Architecture for Wireless Microsensor Networks, IEEE Transactions on Wireless Communications 1 (4) (2002) 660–670.

[10] S. Bandyopadhyay, E. Coyle, An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks, in: Proc. of IEEE INFOCOM, 2003.

[11] V. Mhatre, C. Rosenberg, Design Guidelines for Wireless Sensor Networks Communication: Clustering and Aggregation, Ad-hoc Networks Journal 2 (1) (2004) 45–63.

[12] S. Ganeriwal, R. Kumar, M. Srivastava, Timing-sync Protocol for Sensor Networks, in: Proc. of ACM SenSys, 2003.

[13] CC1000 low power FSK transciever, Chipcon Corporation. http://www.chipcon.com.

[14] C. Lynch, F. O'Reilly, Processor choice for wireless sensor networks, in: Proc. Workshop on Real-World Wireless Sensor Networks, 2005.

[15] A. J. Martin, M. Nystrm, 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, A. Tura, The lutonium: A sub-nanojoule asynchronous 8051 micro-controller, in: Proc. 9th IEEE International Symposium on Asynchronous Systems and Circuits, 2003.

[16] J. R. Vig, Introduction to Quartz Frequency Standards, Technical Report SLCET-TR-92-1, Army Research Laboratory, available at http://www.ieee-uffc.org/freqcontrol/quartz/vig/vigtoc.htm (October 1992).

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

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

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

[20] M. Maroti, B. Kusy, G. Simon, A. Ledeczi, The flooding time synchronization protocol, in: Proc. of ACM SenSys, 2004.

[21] H. Nama, M. Chiang, N. Mandayam, Optimal utility-lifetime trade-off in self-regulating wireless sensor networks: A distributed approach, in: Proc. of 40th Conference on Information Sciences and Systems, 2006.

[22] R. Madan, S. Cui, S. Lall, A. Goldsmith, Cross-layer design for lifetime maximization in interference-limited wireless sensor networks, in: Proc. of IEEE INFOCOM, 2005.

[23] O. Younis, S. Fahmy, Distributed Clustering in Ad-hoc Sensor Networks: A Hybrid, Energy-Efficient Approach, in: Proc. of IEEE INFOCOM, Hong Kong, 2004.

[24] D. P. Bertsekas, Nonlinear programming, Athena Scientific, 1999.

[25] G. Hackmann, C.-L. Fok, G.-C. Roman, C. Lu, Middleware support for seamless integration of sensor and IP networks, in: Proc. of IEEE DCOSS, 2006.

[26] L. Ho, M. Moh, Z. Walker, T. Hamada, C. Su, A prototype on RFID and sensor networks for elder healthcare: progress report, in: Proc. of ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis, 2005.

[27] Y. Wu, S. Fahmy, N. B. Shroff, Optimal QoS-aware sleep/wake scheduling for time-synchronized sensor networks, Technical Report, available at http://www.cs.purdue.edu/homes/fahmy/reports/qos-techrep.pdf (2006).

Appendix: Proofs

To prove Proposition 2, we first prove the following lemma about the properties of several auxiliary functions.

Lemma 1. *(1) Let* $\phi_1(z) = -z + 2Q^{-1}(\frac{1-z}{2})g(Q^{-1}(\frac{1-z}{2}))$*, then* $\phi_1(z) < 0, \forall z \in (0,1)$*;*

(2) Let $\phi_2(z) = (1-z)\frac{g(0)}{g(Q^{-1}(\frac{1}{2}-z))} - 1 + Q^{-1}(\frac{1}{2}-z))g(0)$*, then* $\phi_2(z) > 0, \forall z \in (0, \frac{1}{2})$*;*

(3) Let $\phi_3(z) = \frac{-Q^{-1}(z)}{g(Q^{-1}(z))} - 1$*. For* $z > \frac{1}{2}$, $\phi_3(z)$ *increases with* z*;*

(4) Let $\phi_4(z) = \frac{(1-z)Q^{-1}(\frac{1-z}{2})}{g(Q^{-1}(\frac{1-z}{2}))} - 1$*.* $\phi_4(z)$ *increases with* z*.*

(5) Let $\phi_5(z) = \frac{g(Q^{-1}(\frac{1-z}{2}))}{g(Q^{-1}(1-z))}$*. For* $z \in (0,1)$, $\phi_5(z) \geq \frac{1}{2}$*.*

We include the detailed proof in our technical report [27]. Now we prove Proposition 2.

Proposition 2: (1) For $z \geq 0.86$, $\gamma(z)$ is strictly convex;

(2) for $z \in [0, 0.99], 1.86z < \gamma(z) < 2.52z$.

Proof: (1) We first compute $\gamma''(z)$. Let $w_0(z)$ be the solution to $\min\{G(w) = (1-z)s(w) - w + g(w) - g(s(w)) : s(w) = Q^{-1}(Q(w) - z), -\infty < w < Q^{-1}(z)\}$, and $s_0(z) = Q^{-1}(Q(w_0(z)) - z)$, then

$$\gamma(z) = (1-z)s_0(z) - w_0(z) + g(w_0(z)) - g(s_0(z)). \tag{17}$$

From Proposition 1, w_0 is the unique critical point of $G(w)$, therefore, $w_0(z)$ satisfies

$$\begin{aligned} G'(w_0(z)) = (1-z)\frac{g(w_0(z))}{g(s_0(z))} - 1 \\ +(s_0(z) - w_0(z))g(w_0(z)) = 0. \end{aligned} \tag{18}$$

Using equations (17) (18) and implicit differentiation, we get

$w_0'(z) = \frac{-(1-z)s_0 g(w_0)}{g(w_0)g(s_0)[g(w_0)-g(s_0)] - w_0 g^2(s_0) + (1-z)s_0 g^2(w_0)}$,

$s_0'(z) = \frac{g(w_0)[g(w_0)-g(s_0)] - w_0 g(s_0)}{g(w_0)g(s_0)[g(w_0)-g(s_0)] - w_0 g^2(s_0) + (1-z)s_0 g^2(w_0)}$,

$\gamma'(z) = \frac{1-z}{g(s_0)}$,

$\gamma''(z) = [\frac{(1-z)s_0[g^2(w_0) - g(w_0)g(s_0) - w_0 g(s_0)]}{g^2(w_0)g(s_0) - g^2(s_0)g(w_0) - w_0 g^2(s_0) + (1-z)s_0 g^2(w_0)} - 1]\frac{1}{g(s_0)}$.

Therefore, it suffices to prove for $z \geq 0.86$,

$$\frac{(1-z)s_0[g^2(w_0) - g(w_0)g(s_0) - w_0 g(s_0)]}{g(w_0)g(s_0)[g(w_0)-g(s_0)] - w_0 g^2(s_0) + (1-z)s_0 g^2(w_0)} > 1.$$

From Proposition 1, $Q^{-1}(\frac{1+z}{2}) < w_0 < \min(0, Q^{-1}(z))$, hence $s_0 > Q^{-1}(\frac{1-z}{2}) > 0$. Therefore,

$$g(w_0) - g(s_0) > 0. \tag{19}$$

Thus, the denominator of the left side in the above inequality is positive for any $z \in (0,1)$. We multiply it on both sides, and after some algebraic operations, it suffices to prove for $z \geq 0.86$,

$$[g(s_0) - (1-z)s_0][g(w_0) + w_0] > g^2(w_0). \tag{20}$$

Since $\forall z \geq 0.86, Q^{-1}(\frac{1+z}{2}) < w_0 < Q^{-1}(z)$ and $Q^{-1}(z) < 0$, when $z \geq 0.86$ we have

$$\begin{aligned} w_0 < Q^{-1}(z) \Longrightarrow g(w_0) < g(Q^{-1}(z)) \Longrightarrow -w_0 - g(w_0) \\ > -Q^{-1}(z) - g(Q^{-1}(z)) = g(Q^{-1}(z))\phi_3(z), \end{aligned} \tag{21}$$

and

$$\begin{aligned} Q^{-1}(z) \leq Q^{-1}(0.86) \approx -1.0803 \\ \Longrightarrow -Q^{-1}(z) - g(Q^{-1}(z)) > 0. \end{aligned}$$

Similarly,

$$\begin{aligned} & w_0 > Q^{-1}(\frac{1+z}{2}) \Longrightarrow s_0 > Q^{-1}(\frac{1-z}{2}) \geq 0 \Longrightarrow \\ & g(s_0) < g(Q^{-1}(\frac{1-z}{2})) \Longrightarrow -g(s_0) + (1-z)s_0 > \\ & -g(Q^{-1}(\frac{1-z}{2})) + (1-z)Q^{-1}(\frac{1-z}{2}) \\ & = g(Q^{-1}(\frac{1-z}{2}))\phi_4(z), \end{aligned} \tag{22}$$

and as shown in Lemma 1, $\phi_4(z)$ increases with z, hence for $z \geq 0.86$,

$$\phi_4(z) \geq \phi_4(0.86) \approx 0.5388 \Longrightarrow g(Q^{-1}(\frac{1-z}{2}))\phi_4(z) > 0.$$

Combining equations (21) and (22), we have

$$\begin{aligned} [g(s_0) - (1-z)s_0][g(w_0) + w_0] \\ > \phi_3(z)\phi_4(z)\phi_5(z)g^2(Q^{-1}(z)). \end{aligned} \tag{23}$$

Also,

$$\begin{aligned} w_0 < Q^{-1}(z) \leq Q^{-1}(0.86) < 0 \\ \Longrightarrow g^2(w_0) < g^2(Q^{-1}(z)). \end{aligned} \tag{24}$$

Therefore, it suffices to prove for $z \geq 0.86$,

$$\phi_3(z)\phi_4(z)\phi_5(z)g^2(Q^{-1}(z)) > g^2(Q^{-1}(z)),$$

which is equivalent to showing $\phi_3(z)\phi_4(z)\phi_5(z) > 1$. Further, because $\phi_5(z) \geq \frac{1}{2}, \forall z \in (0,1)$, it suffices to prove for $z \geq 0.86$,

$$\frac{1}{2}\phi_3(z)\phi_4(z) > 1. \tag{25}$$

¿From Lemma 1, $\phi_3(z)$ and $\phi_4(z)$ both increase with z, hence,

$$\begin{aligned}\phi_3(z) \geq \phi_3(0.86), \phi_4(z) \geq \phi_4(0.86) \Longrightarrow \frac{1}{2}\phi_3(z)\phi_4(z) \\ \geq \frac{1}{2}\phi_3(0.86)\phi_4(0.86) \approx 1.0382 > 1.\end{aligned}$$

(2) As computed in the proof of (1),

$s_0'(z) = \frac{g(w_0)[g(w_0)-g(s_0)]-w_0 g(s_0)}{g(w_0)g(s_0)[g(w_0)-g(s_0)]-w_0 g^2(s_0)+(1-z)s_0 g^2(w_0)}$.

Combined with equation (19), both the numerator and the denominator in the above equality are positive, thus $s_0'(z) > 0$.

Next we bound $\gamma(z)$ in two steps.

(i) **Bounding** $\gamma'(z)$

As computed in the proof of (1), $\gamma'(z) = \frac{1-z}{g(s_0)}$. Since $s_0(z)$ increases with z, so $\frac{1}{g(s_0)}$ increases with z; while $1-z$ is a decreasing function. Hence, for an arbitrary interval $[z_1, z_2)$, we have $\frac{1-z_2}{g(s_0(z_1))} < \frac{1-z}{g(s_0(z))} < \frac{1-z_1}{g(s_0(z_2))}, \forall z \in [z_1, z_2)$.

We divide the interval $[0,1)$ into n equal length intervals $[\frac{i}{n}, \frac{i+1}{n}), i = 0 \ldots n-1$, then for $z \in [\frac{i}{n}, \frac{i+1}{n})$, we have

$$L_i = \frac{1-\frac{i+1}{n}}{g(s_0(\frac{i}{n}))} < \gamma'(z) = \frac{1-z}{g(s_0(z))} < \frac{1-\frac{i}{n}}{g(s_0(\frac{i+1}{n}))} = U_i, \tag{26}$$

where L_i, U_i can be numerically computed.

(ii) **Bounding** $\gamma(z)$

Let $z \in [\frac{i}{n}, \frac{i+1}{n})$, we have $\gamma(z) = \int_0^z \gamma'(z)dz = \sum_{j=0}^{i-1} \int_{\frac{j}{n}}^{\frac{j+1}{n}} \gamma'(z)dz + \int_{\frac{i}{n}}^{z} \gamma'(z)dz$, substitute equation (26), we have

$$\frac{\sum_{j=0}^{i-1} L_j \frac{1}{n} + L_i(z-\frac{i}{n})}{z} < \frac{\gamma(z)}{z} < \frac{\sum_{j=0}^{i-1} U_j \frac{1}{n} + U_i(z-\frac{i}{n})}{z}.$$

Hence, $\min_{0\le j\le i} L_j < \dfrac{\gamma(z)}{z} < \max_{0\le j\le i} U_j$. Further, because $\gamma(z)$ is an increasing function, $\frac{\gamma(\frac{i}{n})}{\frac{i+1}{n}} < \frac{\gamma(z)}{z} < \frac{\gamma(\frac{i+1}{n})}{\frac{i}{n}}$. In all, for $z \in [\frac{i}{n}, \frac{i+1}{n})$, we have

$$\max\left(\min_{0\le j\le i} L_j, \frac{\gamma(\frac{i}{n})}{\frac{i+1}{n}}\right) < \frac{\gamma(z)}{z}$$
$$< \min\left(\max_{0\le j\le i} U_j, \frac{\gamma(\frac{i+1}{n})}{\frac{i}{n}}\right). \tag{27}$$

We set $n = 10000$, then use equation (27) and compute that for $z \in [0, 0.99]$, $1.86 < \frac{\gamma(z)}{z} < 2.52$. ■

Proposition 3:(1) $0.929 \le \frac{\gamma(z)}{\gamma_1(z)} \le 1.26$;

(2) $\gamma_1(z)$ is strictly convex.

Proof: (1) When $z \ge z_0$, $\frac{\gamma(z)}{\gamma_1(z)} = 1$. When $0 \le z \ge z_0$, from Proposition 2(2), $1.86z < \gamma(z) < 2.52z$, so $\frac{1.86z}{\gamma_1(z)} \le \frac{\gamma(z)}{\gamma_1(z)} \le \frac{2.52z}{\gamma_1(z)}$. Therefore, for $0 \le z < 1$, $0.929 \le \frac{\gamma(z)}{\gamma_1(z)} \le 1.26$.

(2) We need to show that for $z_1 \ne z_2$, $\gamma_1(\theta z_1 + (1-\theta)z_2) < \theta\gamma_1(z_1) + (1-\theta)\gamma_1(z_2)$. Without loss of generality, assume $z_1 < z_2$, it suffices to show that

$$\frac{\gamma_1(\theta z_1 + (1-\theta)z_2) - \gamma_1(z_1)}{(1-\theta)(z_2 - z_1)} < \frac{\gamma_1(z_2) - \gamma_1(\theta z_1 + (1-\theta)z_2)}{\theta(z_2 - z_1)}. \tag{28}$$

Let $\kappa(z) = 2z + 0.001z^2$. There are three cases:

- $z_1 < z_2 \le z_0$ In this case $\gamma_1(z) = \kappa(z)$. Because $\kappa(z)$ is strictly convex, hence (28) holds.

- $z_0 \le z_1 < z_2$ In this case $\gamma_1(z) = \gamma(z)$. As shown in Proposition 2, $\gamma(z)$ is strictly convex for $z \ge 0.86$. Hence (28) holds.

- $z_1 < z_0 < z_2$ Without loss of generality, suppose $\theta z_1 + (1-\theta)z_2 \le z_0$. By

Mean Value Theorem, in (28)

$$\begin{aligned}
LHS &= \frac{\gamma_1(\theta z_1 + (1-\theta)z_2) - \gamma_1(z_1)}{(1-\theta)(z_2 - z_1)} \\
&= \frac{\kappa(\theta z_1 + (1-\theta)z_2) - \kappa(z_1)}{(1-\theta)(z_2 - z_1)} \\
&= \frac{\kappa'(\zeta_1)(1-\theta)(z_2 - z_1)}{(1-\theta)(z_2 - z_1)} \\
&= \kappa'(\zeta_1),
\end{aligned}$$

where $\zeta_1 \in [z_1, \theta z_1 + (1-\theta)z_2]$.

$$\begin{aligned}
RHS &= \frac{\gamma_1(z_2) - \gamma_1(\theta z_1 + (1-\theta)z_2)}{\theta(z_2 - z_1)} \\
&= \frac{\gamma(z_2) - \gamma(z_0) + \kappa(z_0) - \kappa(\theta z_1 + (1-\theta)z_2)}{z_2 - z_0 + z_0 - [\theta z_1 + (1-\theta)z_2]} \\
&= \frac{\gamma'(\zeta_2)(z_2 - z_0) + \kappa'(\zeta_3)[z_0 - (\theta z_1 + (1-\theta)z_2)]}{z_2 - z_0 + z_0 - [\theta z_1 + (1-\theta)z_2]},
\end{aligned}$$

where $\zeta_2 \in [z_0, z_2], \zeta_3 \in [\theta z_1 + (1-\theta)z_2, z_0]$.

We compute that $\kappa'(z_0) \approx 2.0019 < \gamma'(z_0) \approx 5.7241$. Since $\kappa(z)$ is strictly convex, and $\gamma(z)$ is strictly convex for $z \geq 0.86$. Therefore, we have

$$\begin{gathered}
\gamma'(\zeta_2) \geq \gamma'(z_0) > \kappa'(z_0) \geq \kappa'(\zeta_1) \\
\kappa'(\zeta_3) \geq \kappa'(\theta z_1 + (1-\theta)z_2) \geq \kappa'(\zeta_1)
\end{gathered}$$

Therefore,

$$\begin{aligned}
RHS &= \frac{\gamma'(\zeta_2)(z_2 - z_0) + \kappa'(\zeta_3)[z_0 - (\theta z_1 + (1-\theta)z_2)]}{z_2 - z_0 + z_0 - [\theta z_1 + (1-\theta)z_2]} \\
&> \frac{\kappa'(\zeta_1)(z_2 - z_0) + \kappa'(\zeta_1)[z_0 - (\theta z_1 + (1-\theta)z_2)]}{z_2 - z_0 + z_0 - [\theta z_1 + (1-\theta)z_2]} \\
&= \kappa'(\zeta_1) = LHS.
\end{aligned}$$

Similarly, we can prove (28) holds if $\theta z_1 + (1-\theta)z_2 \geq z_0$.

Hence (28) holds for all possible z, which shows $\gamma_1(z)$ is strictly convex. ■

Summary of Notation: We list all the symbols we use in Tables 2 and 3.

Author Biographies

Yan Wu received the B.S. degree from the University of Science and Technology of China, the M.S. degree from the Chinese Academy of Sciences, and the Ph.D. degree in Computer Science from Purdue University in 2007. He is currently with Microsoft Corporation. His research interests lie in resource allocation, optimization and cross-layer design in wireless and sensor networks.

Sonia Fahmy 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, overlay networks, 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 a member of the ACM and a senior member of the IEEE. For more information, please see: http://www.cs.purdue.edu/∼fahmy/

Ness B. Shroff received his Ph.D. degree from Columbia University, NY in 1994 and joined Purdue university immediately thereafter as an Assistant Professor. At Purdue, he became Professor of the school of Electrical and Computer Engineering in 2003 and director of CWSA (a university-wide center on wireless systems and applications) in 2004. 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. He is especially interested in fundamental problems in the design, performance, control, and security of these networks. For more information, please see: http://www.ece.ohio-state.edu/∼shroff/

Table 2: List of symbols

T_e	Epoch duration
T_s	Synchronization interval
N_s	Number of synchronization messages
T	Transmission period
N	Rounds of transmissions in an epoch
L_i	Message length from node i
r	Compression ratio
$(C(j,k), t_i(j,k))$, $k = 1 \ldots N_s$	Corresponding time instants between CH and member i in epoch j
$a_i(j), b_i(j)$	Clock skew and phase offset (respectively) between node 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
$\overline{C(j,k)}$ and $\overline{C^2(j,k)}$	Refer to Equation (6)
$\hat{\tau}$ 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
$\gamma(\cdot)$	Refer to Equation (5)
$H(n)$	The cluster head of node n
$M(n)$	The set of nodes that are members of n
$D(n)$	The set of nodes that are the descendants of n
$d(n)$	The hop distance from node n to BS
$z(n)$	Capture probability threshold for messages from node n

Table 3: List of symbols (continued)

T_L	Network lifetime
Λ	Predefined threshold on end to end delivery probability
$\varepsilon_s(n)$	Sensing energy in an epoch
$\varepsilon_{syn}(n)$	Synchronization energy in an epoch
l	The amount of sensing data generated by each sensor in a transmission period T
$L^{avg}(n)$	The *average* message size from n
$\varepsilon_t(n)$	average transmission energy of n in an epoch
$\varepsilon_r(n)$	average receiving energy of n in an epoch
$\eta(n, \overrightarrow{z})$	average power dissipated in node n (refer to Equation (11))
$A(n)$	Refer to Equation (11)
$P(n, i)$	Refer to Equation (11)
$Q(n, i)$	Refer to Equation (11)
$\xi(n)$	Initial energy of node n
$\Psi(1/T_L)$	Lifetime-penalty function
u	$= 1/T_L$
$\gamma_1(\cdot)$	Approximation of $\gamma(\cdot)$ (refer to Proposition 2)
$\eta_1(n, \overrightarrow{z})$	Refer to Problem (B1)