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 *†

January 15, 2007

Abstract

We study sleep/wake scheduling for low-duty cycle sensor networks. Our work is different from most previous works in that we explicitly consider the effect of synchronization error in the design of the sleep/wake scheduling algorithm. Prior work on sleep/wake scheduling has either assumed that an underlying synchronization protocol can provide perfect synchronization, or assumed an upper bound on the *clock disagreement*, and used it as a *guard time*. However, for a widely used synchronization scheme we show that its error is non-negligible, and and using a conservative guard time is energy wasteful. We thus conclude that the design of any sleep/wake scheduling algorithm must take into account the impact of the 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 that the problem is non-convex, thus cannot be directly solved by conventional convex optimization techniques. By investigating the unique

*– Yan Wu and Sonia Fahmy are also with the Department of Computer Science, Purdue University, 250 N. University St., West Lafayette, IN 47907–2066, USA. E-mail: {wu26, fahmy}@cs.purdue.edu. Ness B. Shroff is also with the School of Electrical and Computer Engineering, Purdue University, 465 Northwestern Ave., West Lafayette, IN 47907–2035, USA. E-mail: shroff@ecn.purdue.edu

†– This research has been sponsored in part by NSF grants ANI-0238294 (CAREER) and ANI-0207728, an Indiana 21st century grant, and a Tellabs foundation fellowship.

structure of the problem, we transform the non-convex problem into a convex equivalent, and solve it using an efficient search method. Simulation results show that our scheme significantly outperforms schemes that do not intelligently consider the synchronization error.

Next 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, 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. Simulations show that our approximate solution significantly outperforms a scheme without differentiated treatment of the nodes.

1 Introduction

Recent advances in micro-electronic fabrication have allowed the integration of sensing, processing, and wireless communication capabilities into low-cost and low-energy wireless sensors [1, 2]. Sensor network applications can be classified into four classes in terms of the data delivery pattern [3]: (i) continuous, (ii) event driven, (iii) observer initiated, and (iv) hybrid. Our work focuses on the first class, namely, applications employing continuous sensing and data delivery, where each sensor periodically produces a small amount of data. These sensor networks have the following characteristics: (i) a large number of nodes, (ii) upstream traffic, i.e., nodes report their readings to a single (or a few) base station(s), (iii) low duty cycles, and (iv) periodic or highly regular traffic patterns. This large application class includes many typical sensor network applications such as habitat monitoring [4,5], civil structure monitoring [6], and factory maintenance [7]. Several practical systems have been developed for these applications.

A scalable method to manage large sensor networks 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. Time Division Multiple Access (TDMA)-based channel access or frequency allocation can be employed to avoid interference. Moreover, clustering can be extremely effective for data fusion, local decision making, and energy savings [8, 9]. One problem with clustering is that the cluster head is heavily utilized for both intra-cluster coordination and inter-cluster communications. Therefore, the cluster head will quickly deplete its energy. To address this concern, periodic re-clustering is performed to distribute the energy consumption among sensor nodes.

Measurements show that for short-range radio communications in sensor networks, significant energy is wasted due to overhearing, collision, and idle listening. Time Division Multiple Access (TDMA)-based Medium Access Control (MAC) protocols can avoid overhearing and collisions; yet the energy consumption during idle times still causes significant waste. An effective approach to conserve energy 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 with TDMA-based MAC protocols 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, 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 are assumed to continuously monitor their environment and *periodically* report to the cluster head. Because the traffic is highly regular and the load is very low, the cluster head can go to sleep when no activity is going on and only wake up intermittently to send and receive messages. The following question hence becomes critical: *When should the cluster head wake up and how long should it stay active?*

With perfect synchronization, the cluster head and the cluster member simply agree upon a time and wake up simultaneously. But in practice, synchronization always has error. To illustrate the impact of the synchronization error, we investigate a widely used synchronization scheme, proposed in the well-known Reference

Broadcast Synchronization protocol [10]. 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 most synchronization schemes, i.e., due to non-deterministic factors, the synchronization error will grow with time until the next exchange of synchronization messages. We conclude that the design of an effective sleep/wake scheduling algorithm must take into account the impact of synchronization error. We show that there is a trade-off between energy consumption and message delivery performance (defined as the message capture probability in this work). In order to reduce energy consumption, yet still guarantee high message delivery performance, we formulate an optimization problem to minimize the expected energy consumption, with constraints on message delivery performance, i.e., the message capture probability should be no less than a threshold. In the first part of this work, we assume the threshold is *already given*. We show that the problem is non-covex, thus cannot be directly solved by conventional convex optimization techniques. However, by investigating the unique structure of the problem, we are able to transform it into a convex equivalent, and solve it using an efficient search method.

Next 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 the related nodes. This means that as long as the collective performance is guaranteed, the requirement on each individual node can be chosen with *flexibility*. 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 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 2 reviews related work. Section 3 describes the system model. Section 4 discusses the optimal sleep/wake scheduling problem and presents the solution. Section 5 studies how to assign the threshold for messages coming from each individual node. Section 6 concludes the paper.

2 Background and Related Work

We first discuss time synchronization and sleep/wake scheduling, then review previous work on QoS support in sensor networks.

2.1 Time Synchronization for Sensor Networks

Time synchronization has been studied in several contexts, including that of wireless sensor networks [10–18]. Elson and Romer [11] identified differences between time synchronization in sensor networks and traditional computer systems. Clock disagreement among sensor nodes is due to two effects: *phase offset* and *clock skew*. Phase offset corresponds to clock disagreement between nodes at a given instant. Clock skew is because the crystal oscillators used on sensor nodes are imperfect, i.e., there is a difference between the expected frequency and the actual frequency. Further, the frequency may be time varying due to environmental factors, including variations in temperature and pressure [19]. Due to clock skew, clocks diverge over time. Ganeriwal et al. [20] proposed an algorithm that considers environmental changes in making synchronization decisions.

Several synchronization protocols have been proposed to estimate the phase offset and clock skew. Elson et al. [10] have proposed a receiver-receiver synchronization scheme called Reference-Broadcast Synchronization (RBS). In RBS, a node sends beacons to its neighbors using physical-layer broadcast. The recipients use the arrival time of the broadcast as a reference point to compare their times. RBS removes the non-determinism in the transmission time, channel access time, and propagation delay. The only non-determinism is in the packet reception time. To estimate the clock skew and the phase offset, least square linear regression is used. Ganeriwal et al. [12] propose a sender-receiver synchronization approach called Timing-sync Protocol for Sensor Networks (TPSN). The TPSN approach time-stamps synchronization messages at the Medium Access Control (MAC) layer. It eliminates errors caused by access time and propagation delay via two way message exchange. A shortcoming of TPSN is that it does not estimate the clock skew of the nodes. Maroti et al. [16] combine the MAC layer time-stamping of TPSN with clock skew estimation using linear regression, and demonstrated improved performance. In both RBS and TPSN, measurements show that the synchronization error follows a well-behaved *normal distribution with zero mean*. We will use this observation to model the error distribution in our work.

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

2.2 Sleep/Wake Scheduling for Sensor Networks

Current MAC designs for wireless sensor networks can be broadly classified into contention-based or TDMA-based. The IEEE 802.11 Distributed Coordination Function (DCF) [21] is an example of contention-based MAC protocols and is widely used due to its simplicity. Overhearing, collisions, and idle listening waste energy with contention-based MACs. PAMAS [22] tries to avoid overhearing by putting a node to sleep when the message being transmitted is not intended for it. SMAC [23] incorporates virtual clustering to coordinate sleep/wake schedules of neighboring nodes. It further reduces energy consumption via message passing. T-MAC [24] builds upon SMAC by making the duty cycle adaptive. TRAMA [25] tries to guarantee collision-free channel access. Polastre et al. [26] proposed B-MAC for low power listening, where senders use a preamble to alert receivers of the coming data.

Several researchers have argued that TDMA protocols are better suited to sensor network applications, since TDMA protocols avoid energy waste due to contention. PACT [27] adapts node duty cycle to traffic load, and uses passive clustering to further prolong the network lifetime. Coleri et al. [28] propose an energy efficient architecture, in which a centralized Access Point (AP) with unlimited power directly synchronizes with other nodes and explicitly schedules their transmissions. Kannan et al. [29] have introduced the concept of energy-criticality of a sensor node. More critical nodes sleep longer, thereby balancing the energy consumption. Arisha et al. [30] have proposed an energy-aware management protocol for sensor networks. A gateway node sets routes for sensor data and arbitrates medium access among sensors. Sichitiu [31] have proposed a deterministic, schedule-based energy conservation scheme. Time-synchronized sensors form on-off schedules that enable the sensors to be awake only when necessary. The above protocols, however, either assume perfect synchronization in the network, or assume an upper bound on the clock disagreement and use it as a guard time to compensate for the synchronization error.

As pointed out in Section 2.1, existing synchronization protocols like RBS or TPSN achieve micro-second level synchronization at the time instant *immediately following* the exchange of synchronization messages. Due to estimation errors in the clock skew, the clocks will gradually drift as time progresses, until the next exchange of synchronization messages. To see how significant the clock disagreement can be, consider two nodes that have agreed to rendezvous on the radio channel once every 100 seconds to exchange a 20-byte message. Using a 19.2 kbps radio such as RF Monolithics [32], 20 bytes can be transmitted in about 8 ms. The radio must be awakened early to account for clock disagreement. Let the estimation error of the clock skew be 10 ppm[1], i.e., the clocks of the two nodes drift away from each other 10 μs each second. After 100 seconds, the clocks will drift by $10\ \mu s \times 100 = 1\ ms$, which is non-negligible compared to the actual message transmission time. Ye and Heidemann [34] considered this effect in the design of a polling-based MAC protocol called Scheduled Channel Polling(SCP). In SCP, the receivers periodically poll the channel for network activity, and the sender uses a preamble to wake up the receiver before sending the actual message. To accomodate the clock disagreement they extend the preamble by a guard time, which is equal to the product of the maximum clock skew and the time elapsed since last synchronization. Using this worst case clock disagreement as the guard time can compensate for the synchronization error, however, energy efficiency can be further improved by exploiting the non-deterministic nature of the clock disagreement.

2.3 QoS Support for Sensor Networks

Several studies have investigated QoS support issues in sensor networks. Chen et al. [35] discussed the difference between QoS requirement of sensor networks and traditional Internet applications. 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 not decided by the performance of any individual node, but by the *collective* performance of all the related nodes, e.g., collective packet loss and information throughput. Iyer et al. [36] use the number of active nodes as the measure of service quality and adopt Gur Game to adjust the number of nodes staying awake. Kay et al. [37] solve the similar problem using an ACK scheme and demonstrate improved performance. Perillo et al. [38] address the problem of maximizing lifetime for a wireless sensor network while meeting a minimum level of reliability. The maximization is achieved by jointly scheduling active sensor sets and finding paths for data routing. Bhatnagar et al. [39] propose a priority based message dissemination mechanism. Messages carrying important information are assigned high priority level and delivered to the sink with larger probability.

Chen et al. [40] propose a data centric approach in sensor networks. The authors observe that data containing information of different qualities represents different values to the destination. Therefore, the overall system objective is no longer to maximize the *raw data throughput*, but to maximize the amount of *useful information* collected at the sink(s). To quantify the value of messages, they associate each message with a *utility value*, which represents the amount of useful

[1]From the datasheet of Mica Motes [33], the clock skew with respect to the standard clock is up to 50 ppm, thus the relative clock skew between two sensor nodes can be 100 ppm in the worst case.

information contained in it. They then formulate an optimization problem whose objective is to maximize the amount of information (utility) collected at sinks, and derive an energy aware solution. The work has quantified the "value" of messages and proposed a general optimization framework for data transport in sensor networks. However, one assumption made in this work is that there is no redundancy in the network, hence the data collected from different sensors contributes additive utilities. In reality, surplus sensors may be deployed in the sensing area and the information collected by nearby sensors may be redundant and correlated.

3 System Model

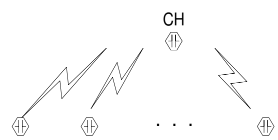


Figure 1: A cluster with a single head and multiple member nodes

We consider the example of a cluster which has been constructed using an existing clustering protocol (e.g. [9, 41–43]). The cluster consists of a single cluster head (CH) and M cluster member nodes $n_1, n_2, \ldots n_M$ (Fig. 1). Time is divided into recurring *epochs* with constant duration T_e. Like many MAC protocols for sensor networks [23–25], 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 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 may occur at a lower frequency than synchronization, i.e., the time between re-clustering the network consists of one or more epochs.

3.1 Assumptions

We make the following assumptions about our system:

(1) Communication pattern: In this work, we focus on the intra-cluster communications, namely, the communications between the cluster members and the CH. Another important question is: after a CH receives messages from its mem-

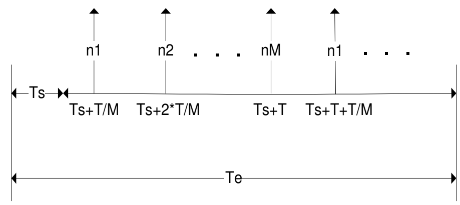


Figure 2: Equispaced upstream transmissions

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

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

(2) Clock skew: Vig [19] 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. The clock skew, however, can change due to environmental factors, such as 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 [20].

The crystal oscillator used by Mica Motes [33] is one type of off-the-shelf crystal oscillator, with similar characteristics to those discussed above. Specifically, its maximum clock skew can be up to 50 ppm.

(3) Radio hardware: For the transmitter circuit, we assume that the sender can precisely control when the message is sent out onto the channel using its *own* clock. This is consistent with the measurements in [12]. Therein, it is observed that non-determinism at the sender is negligible compared to non-determinism at the receiver, i.e., there are minor random effects at the sender.

For the receiver circuit, we assume that if there is an incoming message, the receiver circuit can detect the signal immediately. This is a close approximation of the real situation, since modern transceivers can detect the incoming signal within several microseconds [44]. We further assume that once the receiver circuit detects an incoming message, it can let the processor know, so that the processor will keep the radio active until the reception is completed. This can be easily achieved using a 1-bit status register.

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

(5) Energy expenditure: Measurements show that among all the sensor node components, the radio consumes the most significant amount of energy. In Section 4.2.3, we will show that the computational complexity of our scheduling algorithm is very low. Therefore, in this work, we only account for energy consumption of the radio.

(6) Propagation delay: The communication range for sensor nodes is very short, typically < 100 meters. Therefore, we consider the propagation delay to be negligible.

3.2 Synchronization Algorithm

We adopt a *widely used* synchronization scheme, and study the sleep/wake scheduling problem under this scheme[2]. The scheme was first proposed in RBS [10], and was later adopted by several protocols and system implementations [13–18]. The basic procedure 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.

Either a receiver-receiver approach or a sender-receiver approach can be used in the synchronization protocol. For the purpose of intra-cluster communication, the

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

members only need to synchronize locally with the CH. Thus, at the start of each epoch, each cluster member n_i will exchange several synchronization messages with the CH and obtain N_s pairs of corresponding time instants $(C(j,k), t_i(j,k)), j = 1 \ldots \infty, k = 1 \ldots N_s$, where $C(j,k), t_i(j,k)$ denote the k^{th} time instant of the CH and of node 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.,

$$t_i(C) = a_i(j)C + b_i(j), \tag{1}$$

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.,

$$t_i(j,k) = a_i(j)C(j,k) + b_i(j) + e_i(j,k), \tag{2}$$

where $e_i(j,k)$ is the random error caused by non-determinism in the system. Measurements 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. To be specific, the chi-square test shows a 99.8% confidence [10], which strongly indicates the validity of this model.

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 $\hat{a}_i(j), \hat{b}_i(j)$.

4 Part I: the Optimal Sleep/Wake Scheduling Problem

4.1 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 $(\hat{a}_i(j), \hat{b}_i(j))$,

$$\hat{t}_i(\tau_p) = \hat{a}_i(j)\tau_p + \hat{b}_i(j), \tag{3}$$

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:

$$\tau_p' = \frac{\hat{t}_i(\tau_p) - b_i(j)}{a_i(j)} = \tau_p + \frac{(\hat{a}_i(j) - a_i(j))\tau_p + (\hat{b}_i(j) - b_i(j))}{a_i(j)}. \tag{4}$$

If the estimation is exact, i.e., $(\hat{a}_i(j), \hat{b}_i(j)) = (a_i(j), b_i(j))$, then from Equation (4), $\tau'_p = \tau_p$, i.e., n_i will transmit precisely at τ_p. Under our assumption of negligible propagation delay, τ_p is equal to the time at which p *should* arrive if the synchronization is perfect, i.e., the *scheduled arrival time*; while τ'_p is equal to the time that p *actually* arrives, i.e., the *actual arrival time*. Hence, $\tau'_p = \tau_p$ means that the actual arrival time is exactly the same as the scheduled arrival time. If this is true, the CH simply wakes up at τ_p to receive the message.

However, as given in Equation (2), random errors exist in the measurements. Therefore, $(\hat{a}_i(j), \hat{b}_i(j))$ is also random. As a result, the actual arrival time τ'_p will deviate from the scheduled arrival time τ_p. To compensate for this random deviation and to "capture" (receive) the message, the CH needs to wake up earlier than τ_p and stay active for some time (Fig. 3). This leads to the following question: *When should the cluster head wake up and how long should it stay active?*

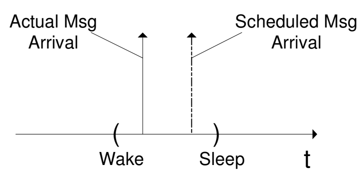


Figure 3: Wake up interval to capture the message

Intuitively, if the CH wakes up much earlier than τ_p and stays active for a long time, it has a high probability 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 the expected energy consumption with constraints on the "capture" probability.

Let p be a message from n_i to arrive during epoch j, i.e., scheduled arrival time $\tau_p \in (jT_e, jT_e + T_e)$. Let τ'_p be the actual arrival time at which p arrives at the CH, as defined in Equation (4). To capture p, the CH wakes up at w_p. If the message does not arrive until s_p, the CH goes back to sleep at s_p; if the message arrives between w_p and s_p, the CH remains active until the message is received, which could be earlier or later than s_p depending upon the actual arrival time and message length. Our goal is to determine w_p and s_p to minimize the expected energy consumption as described by the following optimization problem:

(A) Minimize

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

such that

$$Prob\{\tau'_p \in (w_p, s_p)\} \geq th,$$

where:

- α_I is the power consumption during idle time;
- α_r is the power consumed during reception;
- L_p is the length of the message;
- R is the data rate;
- $f_{\tau'_p}(\cdot)$ is the Probability Density Function (PDF) of τ'_p;
- th is the threshold on the capture probability, $0 < th < 1$. Its value should be decided by the QoS requirement of the application. 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". Later in Section 5 we will study how to set the value of th to meet the QoS requirement of the application.

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

4.2 Solution

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

4.2.1 Computing the PDF $f_{\tau_p'}(x)$

By linear regression analysis [45], Equation (2) can be solved as

$$\hat{a}_i(j) = \frac{\overline{C(j,k)t_i(j,k)} - \overline{C(j,k)} \times \overline{t_i(j,k)}}{\overline{C^2(k,j)} - (\overline{C(j,k)})^2}, \quad \hat{b}_i(j) = \overline{t_i(j,k)} - \hat{a}_i(j)\overline{C(j,k)},$$

where

$$\begin{aligned}
\overline{C(j,k)t_i(j,k)} &= \frac{\sum_{k=1}^{N_s} C(j,k)t_i(j,k)}{N_s}, \ \overline{C(j,k)} = \frac{\sum_{k=1}^{N_s} C(j,k)}{N_s}, \\
\overline{t_i(j,k)} &= \frac{\sum_{k=1}^{N_s} t_i(j,k)}{N_s}, \ \overline{C^2(j,k)} = \frac{\sum_{k=1}^{N_s} C^2(j,k)}{N_s}.
\end{aligned}$$

Substituting this into Equation (4), we have

$$\tau_p' = \frac{1}{a_i(j)} \sum_{k=1}^{N_s} [\frac{1}{N_s} + \frac{1}{N_s} \frac{(\tau_p - \overline{C(j,k)})(C(j,k) - \overline{C(j,k)})}{\overline{C^2(j,k)} - (\overline{C(j,k)})^2}] t_i(j,k) - \frac{b_i(j)}{a_i(j)}. \tag{5}$$

Therefore, τ_p' is a linear combination of $t_i(j,k), k = 1 \dots N_s$ plus a constant offset. Since $t_i(j,k) = a_i(j)C(j,k) + b_i(j) + e_i(j,k)$ is normally distributed, τ_p' is also normally distributed. After some algebraic manipulations, we have

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

Therefore,

$$f_{\tau_p'}(x) = \frac{1}{\sqrt{2\pi}\sigma_p} e^{-\frac{(x-\tau_p)^2}{2\sigma_p^2}}. \tag{7}$$

4.2.2 Transformation

Substituting Equation (7) into problem (A), $\hat{\tau} = \frac{\tau_p' - \tau_p}{\sigma_p}, w = \frac{w_p - \tau_p}{\sigma_p}, s = \frac{s_p - \tau_p}{\sigma_p}$, i.e., $\hat{\tau}, (w, s)$ are the normalized arrival time and normalized wake up interval respectively. After simple algebraic operations, problem (A) is transformed into:

(A1) Minimize

$$\begin{aligned}
F(w,s) &= [1 - Q(w) + Q(s)](s - w)\sigma_p\alpha_I - [Q(w) - Q(s)]w\sigma_p\alpha_I \\
&\quad + [g(w) - g(s)]\sigma_p\alpha_I + [Q(w) - Q(s)]\frac{L_p}{R}\alpha_r
\end{aligned}$$

such that

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

where $g(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ is the probability density function for the standard normal distribution, and $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz$ is the complementary cumulative distribution function.

The main difficulty in solving (A1) is that the problem is not a convex optimization problem. To see this, we compute

$$\begin{aligned}
\frac{\partial^2 F}{\partial w^2} &= \frac{1}{\sqrt{2\pi}}(\frac{L_p}{R}\alpha_r - \sigma_p \alpha_I s) w e^{-\frac{w^2}{2}} - \frac{1}{\sqrt{2\pi}}\sigma_p \alpha_I e^{-\frac{w^2}{2}} \\
&\quad + \frac{1}{\sqrt{2\pi}}\sigma_p \alpha_I w^2 e^{-\frac{w^2}{2}}, \\
\frac{\partial^2 F}{\partial s^2} &= -\frac{1}{\sqrt{2\pi}}\sigma_p \alpha_I e^{-\frac{s^2}{2}} - \frac{L_p}{R}\alpha_r s e^{-\frac{s^2}{2}}.
\end{aligned}$$

If we select $w \to -\infty, s \to \infty$ such that $Q(w) - Q(s) \geq th$, it is easy to see that $\frac{\partial^2 F}{\partial w^2} > 0$, $\frac{\partial^2 F}{\partial s^2} < 0$. The Hessian matrix cannot be positive semidefinite or negative semidefinite, which means that $F(w, s)$ is neither convex nor concave.

Due to the non-convexity of the problem, we cannot use conventional convex programming techniques [46] 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 without explicit knowledge of L_p.

We start by showing the following lemma.

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

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

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

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

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

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

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

Here is an intuitive explanation of $\frac{\partial F}{\partial s} > 0$. As in (A1), we write

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

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

- The expected energy for receiving the message increases because the capture probability is larger;
- The change in the idle energy consumption is illustrated in Fig. 4. In the figure, t_1, t_2, and t_3 are three possible message arrivals, where $t_1 \in [w, s], t_2 \in (s, s+\Delta), t_3 \notin [w, s+\Delta)$. $I_i^{'}, I_i$ $i = 1, 2, 3$ are the idle time for the message arrival at t_i before and after s is increased to $s + \Delta$, respectively.
  - If the message arrival is in $[w, s]$, e.g., t_1, the idle energy consumption does not change;
  - If the message arrival is in $(s, s+\Delta)$, e.g., t_2, the idle energy increases;
  - If the message arrival is at another time, e.g. t_3, the idle energy increases.

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

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

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

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

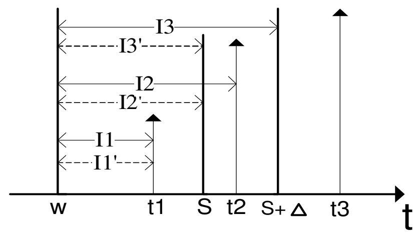


Figure 4: Changes in the idle time when s increases

Meanwhile, we have

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

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

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

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

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

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

(A2) Minimize

$$\begin{aligned} F(w, s) &= (1 - th)(s - w)\sigma_p \alpha_I - th \times w \sigma_p \alpha_I + \\ &\quad \frac{1}{\sqrt{2\pi}}[e^{-\frac{w^2}{2}} - e^{-\frac{s^2}{2}}]\sigma_p \alpha_I + th \frac{L_p}{R} \alpha_r \end{aligned}$$

such that

$$Q(w) - Q(s) = th.$$

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

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

(A3) Minimize

$$G(w) = (1 - th)s(w) - w + g(w) - g(s(w))$$

such that

$s(w) = Q^{-1}(Q(w) - th)$ and $-\infty < w < Q^{-1}(th)$.

We observe that in formulation (A3), the optimal solution (w^*, s^*) only depends on th, specifically, $(w^*, s^*) = (w^*(th), s^*(th))$. Thus, the optimal wake up interval is $(w_p^*, s_p^*) = (\tau_p + \sigma_p w^*(th), \tau_p + \sigma_p s^*(th))$. This has two interesting implications. First, the optimal scheduling policy does not depend on L_p. This is because in the original formulation (A), the goal of optimal scheduling is to minimize the overall expected energy consumption, which is the sum of the expected receiving energy and expected idle energy. However, as shown in Proposition 1, under the optimal scheduling policy, the capture probability has to be equal to th. Thus, the expected receiving energy is fixed to be $th\frac{L_p}{R}\alpha_r$, and will not be affected by different choices of wake up intervals. In this situation, the goal of optimal scheduling degenerates to minimizing the expected idle energy consumption only. Because the value of L_p does not affect idle energy consumption, it has no effect on the scheduling policy.

Second, (w^*, s^*) only depends on th and is the same for messages with different σ_p. This means that we only need to solve for (w^*, s^*) once. As long as the value of th does not change, we can use the solution to compute the wake up intervals for all messages. This greatly facilitates the implementation.

We further notice that from (A2) and (A3), the minimum expected energy to receive the message can be expressed as

$$\sigma_p\alpha_I H(th) + \frac{L_p}{R}\alpha_r th, \tag{8}$$

where

$$\begin{aligned} H(th) \quad = \quad & \min\{G(w) : s(w) = Q^{-1}(Q(w) - th), \\ & w < Q^{-1}(th)\} \end{aligned} \tag{9}$$

is the minimum value of the objective function in (A3). Equations (8) and (9) will be used later in Section 5.

So far, we have transformed the original formulation (A) into an equivalent formulation (A3). Next, we solve (A3).

4.2.3 Solving the Equivalent Formulation

We first show that $G(w)$ is a convex function of w.

Proposition 2 $G^{''}(w) > 0$.

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

Proposition 3 *(1) $G(w)$ has a unique critical point w_0 on $(-\infty, Q^{-1}(th))$;*

(2) w_0 is the global minimum;

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

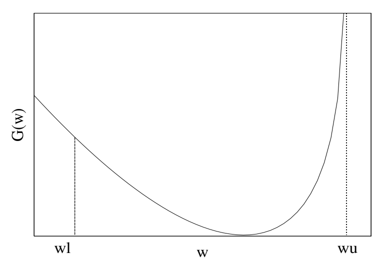


Figure 5: G(w)

Fig. 5 shows the curve of $G(w)$ when $th = 0.85$. Since w_0 is the unique local minimum on (w_l, w_u), we can use an efficient search algorithm such as the Golden Search method to find w_0 [47]. The convergence speed of the Golden Search method is $\rho = \frac{\sqrt{5}-1}{2}$. In order to achieve a precision of δ, the number of iterations n should satisfy $(\frac{\sqrt{5}-1}{2})^n[w_u - w_l] \leq \delta$. We computed that to achieve a precision of 10^{-5}, less than 24 iterations are required for any $th \in (0, 1)$, which means that the search algorithm can be efficiently implemented.

4.3 An Example Implementation

In this section, we describe an example implementation of our approach.

4.3.1 Cluster Initialization

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

4.3.2 Synchronization

As we indicated in Section 3.2, the synchronization scheme can utilize either a receiver-receiver approach (RBS) or a sender-receiver approach (TPSN). Our example implementation uses RBS. We first review how RBS works. In RBS, when two nodes A and B want to synchronize with each other, they need a separate beacon node. The beacon node broadcasts a beacon, which is received at T1 and T2 by A and B respectively. Specifically, let the relationship between node A's clock and node B's clock be $t_B = at_A + b$. Then, $T2 = aT1 + b + e$, where e is the non-deterministic factor, which follows a normal distribution $N(0, \sigma_0^2)$. Hence, one pair of corresponding times $(T1, T2)$ is obtained. Additional pairs can be obtained using multiple broadcast beacons.

To use RBS in the cluster, the CH selects a member as the beacon node. This member sends reference beacons using a sufficiently high power level[3] for the beacons to be received by all other members and the CH. The cluster members then exchange the arrival times of the beacons with the CH and obtain multiple pairs of corresponding time instants. The cluster members will use these pairs to estimate $(a_i(j), b_i(j))$ as described in Sections 3.2 and 4.2.1. At this stage, all the members will have synchronized with the CH except the beacon node. The CH then selects another member to send reference beacons, so that the original beacon node can synchronize with the CH.

To further conserve energy, the cluster members and the CH do not always stay active in the synchronization interval. Instead, they go to sleep if ther are no beacons and wake up right before beacons come. Observe, however, that because of clock disagreement, the CH and the cluster members may not rendezvous with

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

the beacon node. This leads to the loopback problem: the synchronization message itself cannot be successfully exchanged because of synchronization error. To solve this problem, the cluster members and the CH use a guard time to compensate the sync. error. This guard time is chosen to be $3 * message\, transmission\, time$, while the clock disagreement is controlled such that it cannot go far beyond the message transmission time with high probability.Using this mechanism, in all our simulations, beacons and subsequent messages can be successfully communicated.

4.3.3 Determining the Wake up Schedule

To determine the wake up interval, the CH first computes (w^*, s^*) using the Golden Search method. The CH needs to do this computation only once. Next, the CH computes for each message p the value of σ_p using Equation (6). In Equation (6), σ_0 can be obtained from measurements which have already been taken, e.g., in RBS and TPSN. The difficulty is that we do not know $a_i(j)$. However, we can bound $a_i(j)$ in the following manner. According to [19], the maximum clock skew of most off-the-shelf crystal oscillators is no larger than 100 ppm (specifically for Mica Motes, the clock skew is no larger than 50 ppm and the bounds below still hold). Therefore,

$$\frac{1-10^{-4}}{1+10^{-4}} \leq a_i(j) \leq \frac{1+10^{-4}}{1-10^{-4}} \Longrightarrow 0.9998 \leq a_i(j) \leq 1.00021.$$

From Equation (6),

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

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

4.4 Simulation Results

In this section, we conduct several simulations to study our sleep/wake scheduling scheme. Our scheduling policy intelligently compensates for the synchronization error through dynamic computation of the wake up interval. Another scheme that was previously used for compensating the synchronization error assumed an upper bound on the synchronization and use it as a *fixed* guard time. To evaluate the performance gain of dynamic adjustment of wake up intervals, we compare the performance of our scheme with that of the following fixed wake up interval scheme: The CH wakes up $\frac{L}{2}$ earlier than the scheduled message arrival time (recall from Section 4.1 that the scheduled arrival time is the time that the message *should* arrive). If the message does not arrive until $\frac{L}{2}$ after the scheduled arrival time, it goes back to sleep again; otherwise, it stays active until the message is received. To make the comparison fair, we use same message arrivals for both schemes.

In our simulations, we use the synchronization scheme described in Section 4.3.2. We set the synchronization interval to be 60 seconds. During each synchronization interval, the CH transmits to each cluster member in an equispaced manner, and obtains 2 pairs of corresponding times. We adopt the model used in RBS and TPSN to characterize the synchronization error. Specifically, the synchronization error is normally distributed with zero mean, $N(0, \sigma_0^2)$. In our simulations, we choose $\sigma_0 = 36.5\ \mu s$, which is derived from [12][4]. The clock skew of each node is chosen uniformly from $[1 - 50 \times 10^{-6}, 1 + 50 \times 10^{-6}]$ [33].

Table 1 summarizes the simulation parameters and other system constants. Unless specified otherwise, all the simulation results are averaged over 1000 runs.

Table 1: Simulation parameters and system constants

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

[4]In [12], measurements show that the average absolute error is 29.1 μs. Therefore, $\int_{-\infty}^{\infty} |x| \frac{1}{\sqrt{2\pi}\sigma_0} e^{-\frac{x^2}{2\sigma_0{}^2}} dx = 29.1\ \mu s \Longrightarrow \sigma_0 \approx 36.5\ \mu s$.

4.4.1 Comparison with the Fixed Interval Scheme

We first compare the message delivery performance of our scheme with the fixed interval scheme. From Equation (6), we have

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

Within an epoch, the variance of the actual arrival time increases with τ_p, the scheduled arrival time. This is because the clock drifts away more and more as time progresses. As a result, for the fixed interval scheme, the capture rate will decrease as the scheduled arrival time increases. This is illustrated in Fig. 6. In the figure, we show how the capture rate changes as time goes on for both our scheme and the fixed interval scheme. We set $L = 3\ ms$ (recall that in the fixed interval scheme, the CH wakes up $\frac{L}{2}$ earlier than the scheduled arrival and stays active $\frac{L}{2}$ after the scheduled arrival). We observe that for the fixed interval scheme, the capture rate is very high at the beginning, but gradually decreases to below the threshold. If the message is scheduled to arrive near the end of the epoch ($T_e = 1200$), then the capture rate is only 0.55. In practice, this means that the fixed interval scheme cannot provide the threshold capture rate near the end of the epoch, which is undesirable. On the other hand, our scheme dynamically selects the wake up interval, so that the capture rate is always kept at no less than the threshold.

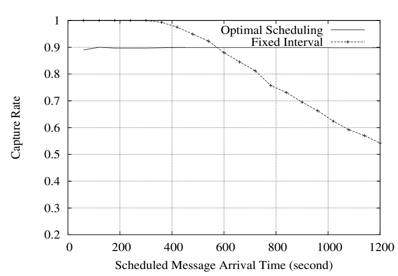
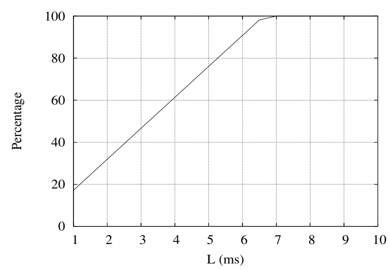


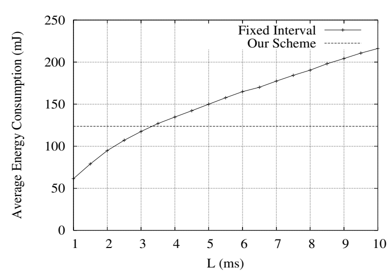
Figure 6: Comparison of the message delivery performance

We next study the energy consumption properties of the two schemes. In Fig. 7(a), we vary L and compute the length of time (given as a percentage of the epoch) that the specified value of L is sufficient to give acceptable capture rates (above the threshold). For example, when $L = 3\ ms$, $percentage \approx 50\%$ means that if L is set to $3\ ms$, then for messages scheduled to arrive during the first half of the epoch, the capture rate is no less than the threshold; but if the message is scheduled to arrive during the second half of the epoch, the capture rate is lower than the threshold.

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



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



(b) Energy consumption of the two schemes

Figure 7: Energy consumption properties of the fixed interval scheme and our scheme

We have also simulated the following combinations of parameters (please refer to Table 1 for the definition of these quantities): $th = 0.9, 0.8, 0.7, 0.6$; $L_p =$

$8, 16, 24$; $T_e = 20, 30, 40, 50, 60$; $T_s = 30, 60, 90, 120, 150$; $N_s = 2, 4, 6, 8, 10$; and $T = 60, 120, 180, 240$. The results were consistent with those discussed above, i.e., the optimal scheduling scheme can guarantee a specified capture rate with lower energy consumption than the fixed interval scheme.

4.4.2 Impact of System Parameters

In this section, we investigate how the choice of system parameters will affect the energy savings of our scheme over the fixed interval scheme. We first investigate how the synchronization scheme parameters, namely, N_s and T_s, affect the performance gain of the optimal scheduling scheme. To make a fair comparison between the energy consumption of our scheme and the fixed interval scheme, for each configuration we choose L to be the minimum interval that can guarantee the threshold capture rate for the entire epoch, i.e.,

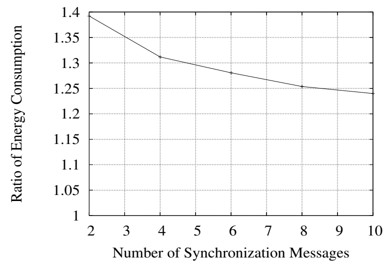
$L = \min\{x$: the fixed interval scheme with $L = x$ can guarantee the threshold capture rate for the entire epoch$\}$.

Fig. 8 depicts how the performance gain of the optimal scheduling scheme changes with N_s and T_s. We observe that as N_s increases, the performance gain of the optimal scheduling scheme gradually decreases. This can be explained as follows. The energy saving of optimal scheduling comes from reducing energy waste. More synchronization messages leads to a better synchronized cluster and reduce the idle listening time. Hence, the overall energy efficiency for both schemes is improved. Under this situation, though the optimal scheduling still consumes less energy than the fixed interval scheme, the performance gain becomes smaller.

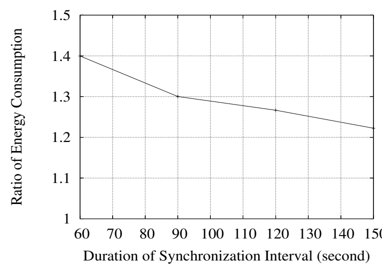
Similarly, when T_s becomes larger, the cluster will become better synchronized. This can be observed from Equation (6). As T_s increases, $C(j,k), k = 1 \dots N_s$ become more spread and $\overline{C^2(j,k)} - \overline{C(j,k)}^2$ increases. Hence, σ_p becomes smaller. This means that the actual message arrival is more likely to be in the vicinity of the scheduled arrival time, i.e., the network is more precisely synchronized.

The above discussion shows that we can save energy by increasing N_s and T_s. However, in practice, N_s and T_s cannot be arbitrarily increased. Increasing N_s means that more synchronization messages need to be exchanged between the CH and the cluster member nodes, which costs more energy; while increasing T_s means that the system will spend more time in synchronization operations, and cannot perform other sensing and communication tasks[5]. Therefore, there exists a trade-off between synchronization and scheduling. We can achieve better perfor-

[5]Fig. 8(b) shows that even if the synchronization interval is as large as 12.5% of the epoch duration ($T_s = 150 seconds = 12.5\% T_e$), the performance gain is still larger than 120%.



(a) Impact of the number of synchronization messages



(b) Impact of the duration of the synchronization interval

Figure 8: Impact of synchronization scheme parameters on the ratio of the energy consumed by the fixed interval scheme to our scheme

mance in scheduling at the cost of more synchronization energy/time. An interesting question arises: what is the optimal scheme if synchronization and scheduling are *jointly* considered? We consider this to be an open issue and plan to investigate it in our future work.

5 Part II: QoS Aware Assignment of the Capture Probability Threshold th

In Section 4, we studied the optimal sleep/wake scheduling problem under the assumption that the capture probability threshold th is *already given* and is uniform for messages coming from different cluster members. In this section, we study how to decide the capture probability threshold to meet the QoS requirement of the application with minimum energy consumption.

5.1 QoS Model and Problem Definition

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" of the environment. The base station uses the collected information to analyze the interested properties, e.g., the chemical contaminant in the area. 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, i.e., the more information collected, the better accuracy. This is consistent with the discusssion in [35], namely, the service quality is not decided by any individual node, but by the collective performance of all the related nodes.

In many sensor networks, hetergenity exists among the sensor nodes. For example, some nodes may be quipped with an expensive sensor which provides high precision measurements, while others only have a low precision sensor for cost reasons. As a result, messages from different nodes may contain information of different qualities and represent different "values". To quantify the value of messages, Chen et al. [40] associate each message with a *utility value*, which represents the amount of useful information contained in it. Using the utility as the quantitative measure of service quality, they proposed a general optimization framework for data transport in sensor networks. However, one assumption made in this work is that there is no redundancy in the network, hence the data collected from different sensors contributes additive utilities. In reality, surplus sensors may be deployed in the sensing area and the information collected by nearby sensors may be redundant and correlated[6].

In this work, we use a method similar to [40]. We associate each message with a *utility value*, which represents the amount of useful information contained in it; messages from the same node i has same utility value U_i, $i = 1 \dots M$. But unlike [40], we consider the messages coming from different nodes are correlated and have redundancy. Therefore, to guarantee the service quality 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 desirable proportion of total utility with minimum energy consumption, we formulate the following optimization problem.

Given an epoch j, as described in Section 3, 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 want to choose $z_i(j)$ to minimize the expected total energy consumption of the CH, and still collect the desirable proportion of the total utility, namely,

[6]Coverage scheduling can help reduce the redundancy, but experimental measurements [48] show that the correlation pattern can be very complex and it is hard to *completely* remove the redundancy.

(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 $z_i(j)$, namely, $E_i(j,h,z_i(j))$ is the minimum value of the objective function in Problem (A) (defined in Section 4.1) 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, namely, $100(1-r)\%$ of the total utility is enough for the CH to make correct judgement about the environment; any additional information is redundant[7];

- p_i is the minimum capture probability threshold for all messages from i. It is used to guarantee the reliability of the system. Without these constraints, it could happen that the thresholds assigned to certain nodes are too low, which means messages from these nodes will almost be ignored. These contraints guarantee that all the cluster members have a minimum opportunity to pass their information on to the CH.

Here is a numerical example of Problem (B). Consider a cluster with only two members besides the CH, n_1 and n_2. n_1 is euqipped with a high precision sensor, while n_2 only has a cheap low precision sensor. Hence, messages from n_1 has higher utility value than that of n_2, $U_1 = 2U_2$. The redundancy level of this cluster is known to be 0.1. To achieve the constraint

$$\sum_{i=1}^{M} z_i(j)U_i \geq (1-r)\sum_{i=1}^{M} U_i,$$

[7]The value of r is application specific and how to decide it is beyond the scope of the work we present in this paper, but we will mention that it is affected by factors including node density, sensing coverage and accuracy requirement. In practice, r can obtained either through theoretical computation or from online training.

i.e.,

$$2z_1(j) + z_2(j) \geq 2.7,$$

we have infinitely many choices of $(z_1(j), z_2(j))$, e.g., $(z_1(j), z_2(j)) = (0.9, 0.9)$ or $(z_1(j), z_2(j)) = (0.95, 0.8)$. The goal is to find the one with minimum energy consumption. Further, for each fixed $(z_1(j), z_2(j))$, the CH will use the sleep/wake schedule developed in Section 4.2, as it is the *optimal sleep/wake schedule*. This is why $E_i(j, h, z_i(j))$ is the minimum value of the objective function in Problem (A) with $\tau_p = jT_e + T_s + i\frac{T}{M} + hT$ and $th = z_i(j)$.

5.2 Solution

We first demonstrate that Problem (B) is not convex, hence it is difficult to solve in general. Then we obtain a suboptimal solution that approximates the optimum.

5.2.1 Non-Convexity of Problem (B)

Since the objective function in Problem (B) is the sum of many $E_i(j, h, z_i(j))$s with different i, h (recall that j is fixed for each epoch), we first analyze the properties of $E_i(j, h, z_i(j))$. From our earlier discussions, $E_i(j, h, z_i(j))$ is exactly the minimum value of the objective function in Problem (A) with $\tau_p = jT_e + T_s + i\frac{T}{M} + hT$ and $th = z_i(j)$, which is (by Equation (8))

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

Here L_p is the message size, σ_p is computed from Equation (6), th is the required threshold and $H(th)$ is given in Equation (9).

To get $E_i(j, h, z_i(j))$ from $E(th)$, we compute σ_p using equation (6) with $\tau_p = jT_e + T_s + i\frac{T}{M} + hT$ and let $th = z_i(j)$, namely,

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

Therefore, $\sum_{i=1}^{M} \sum_{h=1}^{N} E_i(j, h, z_i(j))$ can be written as $\sum_{i=1}^{M} A_i(j) H(z_i(j)) + B_i(j) z_i(j)$, where

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

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

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

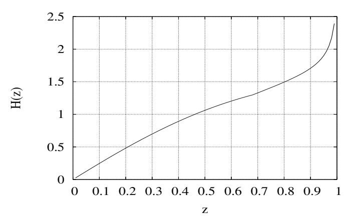


Figure 9: H(z)

We numerically compute $H(z)$ and show the curve in Fig. 9. Obviously it is not convex, hence Problem (B1) is not convex. Further, we don't have an explicit analytical form for $H(z)$. This makes Problem (B1) hard to solve. Next we investigate the structure of the problem and obtain an approximate solution.

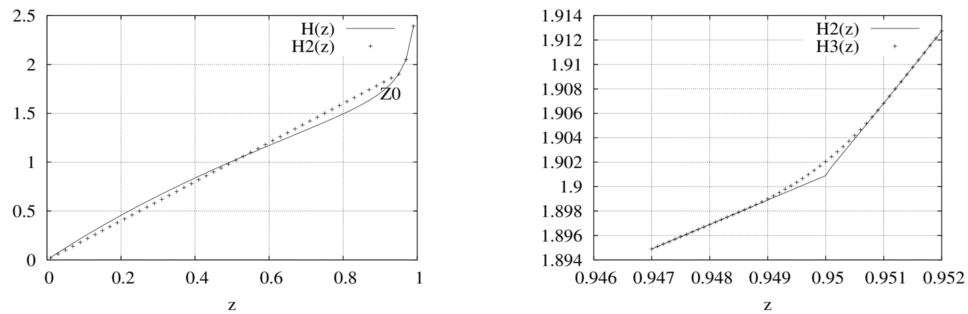
The following proposition characterizes $H(z)$.

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

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

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

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



(a) (b)

Figure 10: Appximating H(z)

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

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

We proved this proposition using Proposition 4 [49]. Fig. 10 (b) illustrates that $H_2(z)$ is a good approximation to $H(z)$. However, one issue is that $H_2(z)$ is not differentiable at Z_0, because

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

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

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

$$\begin{aligned}
q_0 &= H_2(z_1), q_1 = H_2'(z_1), \\
q_2 &= \frac{3[H_2(z_2) - H_2(z_1)] - H_2'(z_2)(z_2 - z_1) - 2H_2'(z_1)(z_2 - z_1)}{(z_2 - z_1)^2}, \\
q_3 &= \frac{[H_2'(z_2) - H_2'(z_1)](z_2 - z_1) - 2[H_2(z_2) - H_2(z_1) - H_2'(z_1)(z_2 - z_1)]}{(z_2 - z_1)^3}, \\
H_3(z) &= \begin{cases} H_2(z) & 0 \le z \ge z_1 \\ q_3(z - z_1)^3 + q_2(z - z_1)^2 + q_1(z - z_2) + q_0 & z_1 \le z_0 \le z_2 \\ H_2(z) & z_2 \le z < 1 \end{cases} \qquad (10)
\end{aligned}$$

We notice that:

- It can be easily verified that $H_3(z)$ is differentiable for all $z \in [0, 1)$.
- If z_1, z_2 are chosen to be very close to z_0, then we have $H_2(z_2) - H_2(z_1) \approx H_1'(z_0)(z_0 - z_1) + H'(z_0)(z_2 - z_0)$. Let $\frac{z_0 - z_1}{z_2 - z_1} = \theta$. Put into equation (10), we have

  $$q_2 \approx \frac{(3\theta - 1)[H'(z_0) - H_1'(z_0)]}{z_2 - z_1}, q_3 \approx \frac{(1 - 2\theta)[H_2'(z_0) - H_2'(z_0)]}{(z_2 - z_1)^2}.$$

  From this intuition[8] we conjecture that if we choose z_1, z_2 such that $\theta \in (\frac{1}{3}, \frac{1}{2})$, then q_2, q_3 are both positive, hence $H_3(z)$ becomes strcitly convex for $z \in [z_1, z_2]$. Using similar techniques in Proposition 5(2), we can show that $H_3(z)$ is strictly convex for $z \in [0, 1)$.
- Further, from the continuity of $H_2(z)$, z_1, z_2 can be chosen to be aribitrarily close to z_0 (with θ unchanged), so that $H_3(z)$ arbitrarily approximates $H_2(z)$.

We choose $z_1 = z_0 - 0.0015, z_2 = z_0 + 0.0010$ and obtain $H_3(z)$ using equation (10). The obtained $H_3(z)$ is differentiable, as illustrated in Fig. 10(b). Further, it can be shown via computations that $H_3(z)$ is strictly convex and $1 \leq \frac{H_3(z)}{H_2(z)} \leq 1.005$. From Proposition 5, $0.929 \leq \frac{H(z)}{H_2(z)} \leq 1.26$, hence,

$$0.925 \leq \frac{H(z)}{H_3(z)} = \frac{H(z)}{H_2(z)} \frac{H_2(z)}{H_3(z)} \leq 1.26. \tag{11}$$

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

(B2) Minimize

$$I_2(\overrightarrow{z}) = \sum_{i=1}^{M} A_i H_3(z_i) + B_i z_i$$

such that

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

[8] Here we do not need a rigorous proof. We just use this guideline to find *one particular* (z_1, z_2), which makes $H_3(z)$ convex.

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

Proposition 6 *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 equation (11), $0.925 \leq \frac{H(z)}{H_3(z)} \leq 1.26$. Hence,

$$0.925 \leq \frac{I_1(\overrightarrow{z})}{I_2(\overrightarrow{z})} \leq 1.26. \tag{12}$$

Therefore,

$$I_1(\overrightarrow{\tilde{z}^*}) \leq 1.26 \times I_2(\overrightarrow{\tilde{z}^*}) \leq 1.26 \times I_2(\overrightarrow{z^*}) \leq 1.26 \times I_1(\overrightarrow{z^*})/0.925 \leq 1.37 I_1(\overrightarrow{z^*}).$$

The first and third "$\leq$" come from equation (12), and the second "$\leq$" holds because $\overrightarrow{\tilde{z}^*}$ is the optimal solution of (B2). ■

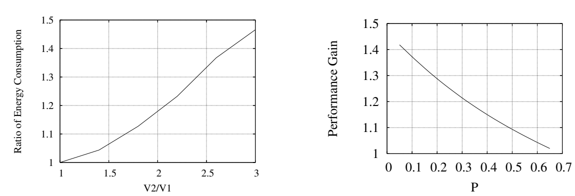
Proposition 6 is important as it shows that $\overrightarrow{\tilde{z}^*}$ is an approximate solution to (B1) with *approximation ratio* 1.37. As described earlier, (B1) is a non-convex optimization problem, hence it is difficult to obtain the optimal solution $\overrightarrow{z^*}$. However, (B2) is 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 [46]. 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 6, the energy consumption using $\overrightarrow{\tilde{z}^*}$ is no more than 37% larger than the minimum.

5.3 Simulation Results

For simulations we consider a cluster of $M = 10$ nodes. We assume the redundancy level of the cluster, r, is known to be 0.7. Half of the nodes have utility value of $V1$, while the other half are more powerful and have utility value of $V2 > V1$. We further set $p_i = P = 0.1, i = 1 \ldots M$. Other simulation parameters are as specified before in Section 4.4 Table 1.

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. 11(a), 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 comes from differentiated treatment of the nodes, namely, 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 to treat the nodes differently; as $\frac{V2}{V1}$ increases, the difference between nodes becomes larger, which makes it profitable to provide differentiated services to the nodes and favor the important ones.

In Fig. 11(b) we keep the value of $\frac{V2}{V1}$ fixed at 3, and vary the value of P(we still set $p_i = P, \forall i = 1 \dots M$). We observe that the performance gain decreases as P increases. This is as expected. When P is small, some nodes might be assigned very small capture probability threshold, which makes the system less reliable. As P increases, the system reliability is improved. But at the same time, the constrain region $z_i \geq p_i$ shrinks, which means we have less flexibility in choosing z_i. Consequently, the performance gain becomes less significant. Hence, the choice of P controls a trade off between system reliability and energy saving.



(a) (b)

Figure 11: Performance gain over the uniform assignment scheme

6 Conclusions and Future Work

In this paper, we have studied sleep/wake scheduling for low duty cycle sensor networks. Our work is different from most previous work in that we explicitly consider the effect of synchronization error in the design of sleep/wake scheduling algorithm. Most previous work on sleep/wake scheduling either assumes perfect synchronization, or assumes an upper bound on the clock disagreement and uses a fixed guard time to compensate for the synchronization error. We utilized a widely used synchronization scheme, which was proposed in the well known RBS protocol. We demonstrated that this scheme, though it achieves microsecond level synchronization *immediately after* the exchange of synchronization messages, *turns out to have non-negligible clock disagreement as time progresses.* Therefore, We conclude that the design of any sleep/wake scheduling algorithm must take into account the impact of this synchronization error, and study the optimal sleep/wake scheduling scheme with consideration of the synchronization error. 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. The problem is non-convex, hence cannot be directly solved by conventional convex optimization techniques. By exploiting the structure of the problem, we were able to transform the original non-convex problem into a convex equivalent, and solve it using an efficient searching method.

Next 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 the fact that in many sensor network applications, the QoS is not decided by the performance of any individual node, but by the *collective* performance of all the related nodes. We thus formulated an optimization problem, which aims to set the 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. However, by investigating its unique structure, we have obtained a suboptimal solution with approximation ratio 1.37. Simulations show that our approximate solution significantly outperforms a scheme without differentiated treatment of the nodes.

We want to point out that the work conducted in this paper has largely been motivated by sensor networks running continuous monitoring applications. While this encompasses a large class of interesting applications, our approach is not limited to such applications. The major requirement of our approach is that the sender and the receiver agree upon message arrival times. This requirement can also be satisfied in many sensor applications with a *hybrid data delivery pattern* [50, 51]. For example, a sensor network which monitors the concentration of a chemical can have two modes: the silent mode and the vigilant mode. When the average concentration collected from the whole network is lower than a certain threshold, the base station will set the network in silent mode, where nodes monitor the chemical concentration and periodically report to the base station; however, if the base station finds the average concentration to exceed the threshold, it can trigger the network into vigilant mode, where all the nodes stay active and transmissions can occur at any time. For such applications with a hybrid data delivery pattern, our approach is ideal for use in the silent mode.

We are currently extending this work in the following directions:

- In this work, we have fixed the synchronization scheme and 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. As illustrated in our simulations (Section 4.4.2), there is a trade-off between synchronization and scheduling, i.e., we can achieve better performance in scheduling at the cost of more synchronization energy/time. Therefore, it is necessary to jointly consider synchronization and scheduling to improve the overall system performance.

- As discussed in Section 3, the scenario considered in this paper is many-to-one communication, e.g., single hop intra-cluster communication. After messages are received by the cluster head, they may need to be forwarded to the base station, potentially over multiple hops. The cluster head may also need to relay messages from other cluster heads to the base station. An important question is how to decide the sleep/wake schedules of the cluster heads over such multiple hops. Section 3 gave a simple example mechanism. A more efficient solution would be to develop an adaptive sleep/wake scheduling methodology, as in this paper, for inter-cluster communications. This is an open issue that merits further investigation.

To prove the propositions, we first study the properties of several auxiliary functions.

Lemma 2 *(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* $z \in (0, 1)$*,* $\phi_5(z) \geq \frac{1}{2}$*.*

Proof: (1) Given that $[Q^{-1}(\frac{1-z}{2})]' = \frac{1}{2g(Q^{-1}(\frac{1-z}{2}))}$, we have

$$\begin{aligned}
\phi_1(0) &= -0 + 2Q^{-1}(\frac{1-0}{2})g(Q^{-1}(\frac{1-0}{2})) = 0, \\
\phi_1'(z) &= -1 + 2[Q^{-1}(\frac{1-z}{2})]'g(Q^{-1}(\frac{1-z}{2})) - 2Q^{-1}(\frac{1-z}{2})g(Q^{-1}(\frac{1-z}{2}))[Q^{-1}(\frac{1-z}{2})]' \\
&= -[Q^{-1}(\frac{1-z}{2})]^2 < 0, \forall z \in (0, 1).
\end{aligned}$$

Hence $\phi_1(z) < 0, \forall z \in (0, 1)$.

(2) Since

$$\begin{aligned}
\phi_2(0) &= (1-0)\frac{g(0)}{g(Q^{-1}(\frac{1}{2}-0))} - 1 + Q^{-1}(\frac{1}{2}-0))g(0) = 0, \\
\phi_2'(z) &= (1-z)\frac{g(0)Q^{-1}(\frac{1}{2}-z)}{g^2(Q^{-1}(\frac{1}{2}-z))} > 0, \forall z \in (0, \frac{1}{2}),
\end{aligned}$$

we have $\phi_2(z) > 0, \forall z \in (0, \frac{1}{2})$,.

(3) $Q^{-1}(z)$ decreases with z, thus, when $z > \frac{1}{2}$, $Q^{-1}(z) < Q^{-1}(\frac{1}{2}) = 0$. Let $z_1 > z_2 > \frac{1}{2}$, we have $Q^{-1}(z_1) < Q^{-1}(z_2) < 0$ and $0 < g(Q^{-1}(z_1)) < g(Q^{-1}(z_2))$. Therefore

$$\phi_3(z_1) = \frac{-Q^{-1}(z_1)}{g(Q^{-1}(z_1))} - 1 > \frac{-Q^{-1}(z_2)}{g(Q^{-1}(z_2))} - 1 = \phi_3(z_2).$$

(4) Let $x(z) = Q^{-1}(\frac{1-z}{2})$,then $\phi_4(z) = \frac{2x(z)Q(x(z))}{g(x(z))} - 1$. Since $x(z)$ increases with z, it suffcies to show that $\frac{2xQ(x)}{g(x)}$ increases with x.

From [52] $Q(x)$ has the following properties: $g(x) > xQ(x)$ and $lim_{x\to\infty}\frac{xQ(x)}{g(x)} = 1$.

$(\frac{xQ(x)}{g(x)})' = \frac{Q(x)-xg(x)+x^2Q(x)}{g(x)}$, thus it suffices to show that $Q(x) - xg(x) + x^2Q(x) \geq 0$.

Suppose $Q(x_0) - x_0g(x_0) + x_0^2Q(x_0) < 0$ for some x_0. Then because $(Q(x) - xg(x) + x^2Q(x))' = -2g(x) + 2xQ(x) < 0$, for $x > x_0$ we have

$$Q(x) - xg(x) + x^2Q(x) < Q(x_0) - x_0g(x_0) + x_0^2Q(x_0) < 0.$$

Hence, for $x \geq x_0$, $(\frac{xQ(x)}{g(x)})' < 0$. Therefore, for $x > x_0$, $\frac{xQ(x)}{g(x)} < \frac{x_0Q(x_0)}{g(x_0)} < 1$. But this is contradictory to the fact that $lim_{x\to\infty}\frac{xQ(x)}{g(x)} = 1$.

Therefore $Q(x) - xg(x) + x^2Q(x) \geq 0$. Consequently, $\phi_4(z)$ increases with z.

(5) We first show that $\phi_5(z)$ decreases with z.

$$\phi_5'(z) = -\frac{1}{2}\frac{Q^{-1}(\frac{1-z}{2})g(Q^{-1}(1-z)) - 2Q^{-1}(1-z)g(Q^{-1}(\frac{1-z}{2}))}{g^2(Q^{-1}(1-z))}.$$

Let $y_1 = Q^{-1}(\frac{1-z}{2}), y_2 = Q^{-1}(1-z)$, it suffices to show that

$$\frac{y_1Q(y_1)}{g(y_1)} \geq \frac{y_2Q(y_2)}{g(y_2)},$$

which is true because $y_1 > y_2$ and $\frac{2yQ(y)}{g(y)}$ is an increasing function.

Next we show $lim_{z\to 1}\phi_5(z) = \frac{1}{2}$. Because $\phi_5(z)$ decreases and is always positive, so the above limit must exist. Let it be d. From L'Hopital's rule, we have

$$\begin{aligned} d &= lim_{z\to 1}\frac{g(Q^{-1}(\frac{1-z}{2}))}{g(Q^{-1}(1-z))} = lim_{z\to 1}\frac{-Q^{-1}(\frac{1-z}{2})g(Q^{-1}(\frac{1-z}{2}))\frac{1}{2g(Q^{-1}(\frac{1-z}{2}))}}{-Q^{-1}(1-z)g(Q^{-1}(1-z))\frac{1}{g(Q^{-1}(1-z))}} \\ &= lim_{z\to 1}\frac{Q^{-1}(\frac{1-z}{2})}{2Q^{-1}(1-z)} = lim_{z\to 1}\frac{\frac{1}{2g(Q^{-1}(\frac{1-z}{2}))}}{\frac{2}{g(Q^{-1}(1-z))}} = \frac{1}{4d}, \end{aligned}$$

i.e., $d^2 = \frac{1}{4}$. Combined with $d \geq 0$ we have $d = \frac{1}{2}$.

Because $\phi_5(z)$ decreases with z and $lim_{z\to 1}\phi_5(z) = \frac{1}{2}$, we have $\phi_5(z) \geq \frac{1}{2}, \forall z \in (0,1)$. ■

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

Proof: We will use the following properties of $g(x)$ in the proof: $g(x) = g(-x), g'(x) = -xg(x)$.

From formulation (A3),

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

It is sufficient to show that

$$s^{''}(w) > 0$$

and

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

We first examine $s^{''}(w)$. Taking the derivative on both sides of $Q(w) - Q(s(w)) = th$, we have

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

Thus,

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

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

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

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

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

Next,

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

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

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

There are three cases:

- $w < s \leq 0$

  By the Mean Value Theorem,

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

we have

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

Then

$$\begin{aligned} w\le\zeta\le s\le 0 &\implies 0\le\frac{g(\zeta)}{g(s)}\le 1, 0\le\frac{\zeta}{w}\le 1\\ &\implies 1-\frac{g(\zeta)}{g(s)}\frac{\zeta}{w}\ge 0\\ &\implies -w(s-w)[1-\frac{g(\zeta)}{g(s)}\frac{\zeta}{w}]\ge 0. \end{aligned}$$

- $0\le w<s$

  This case can be proved using Mean Value Theorem in the same way as above. Just notice that $0\le w<s$ implies $\frac{g(\zeta)}{g(s)}\ge 1$ and $\frac{\zeta}{w}\ge 1, \forall\zeta\in[w,s]$.

- $w\le 0\le s$

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

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

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

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

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

$$\begin{aligned} &w\le\zeta\le -s\le 0\Longrightarrow 0\le\frac{\zeta}{w}\le 1, 0\le\frac{g(\zeta)}{g(-s)}\le\\ &\le 1, -1\le\frac{w+s}{s-w}\le 0\Longrightarrow\frac{\zeta}{w}\frac{g(\zeta)}{g(-s)}\frac{w+s}{s-w}\ge -1\\ &\Longrightarrow -w(s-w)[1+\frac{\zeta}{w}\frac{g(\zeta)}{g(-s)}\frac{w+s}{s-w}]\ge 0. \end{aligned}$$

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

Proposition 3: (1) $G(w)$ has a unique critical point w_0 on $(-\infty, Q^{-1}(th))$;

(2) w_0 is the global minimum;

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

Proof: (1) We first show the existence of the critical point. Assume that there is no critical point. Then, because $G^{'}$ is continuous (from the existence of $G^{''}$), $G^{'}(w)$ is either strictly positive for all $w \in (-\infty, Q^{-1}(th))$, or strictly negative for all $w \in (-\infty, Q^{-1}(th))$.

First assume $G^{'}(w) < 0, \forall w \in (-\infty, Q^{-1}(th))$. Choose an arbitrary point w_1 from $(-\infty, Q^{-1}(th))$, then

$$\forall w > w_1, G(w) < G(w_1).$$

However, as $w \to Q^{-1}(th), s(w) \to \infty$, so

$$G(w) = (1-th)s(w) - w + g(w) - g(s(w)) \to \infty > G(w_1),$$

which is contradictory.

A similar contradiction can be obtained if we assume $G^{'}(w) > 0, \forall w \in (-\infty, Q^{-1}(th))$. Therefore, there exists at least one critical point on $(-\infty, Q^{-1}(th))$.

Next, we show uniqueness of the critical point. Suppose there is another critical point $G^{'}(w_2) = 0$. Then, by the Mean Value Theorem, $G^{''}(\zeta) = 0$ for some $\zeta \in [w_0, w_2]$. This is contradictory since $G^{''}(w) > 0$. Thus, the critical point must be unique.

(2) Global minimum immediately follows from convexity.

(3) To show $w_0 > w_l$, we first show that $G^{'}(w_l) < 0$.

$$\begin{aligned} G^{'}(w_l) &= (1-th)s^{'}(w_l) - 1 + (s(w_l) - w_l)g(w_l) \\ &= (1-th)\frac{g(w_l)}{g(s(w_l))} - 1 + (s(w_l) - w_l)g(w_l), \end{aligned}$$

$$\begin{aligned} w_l = Q^{-1}(\frac{1+th}{2}) &\Longrightarrow s(w_l) = Q^{-1}(\frac{1-th}{2})) = -w_l \\ &\Longrightarrow g(s(w_l)) = g(w_l) \\ &\Longrightarrow G^{'}(w_l) = -th + 2Q^{-1}(\frac{1-th}{2}))g(Q^{-1}(\frac{1-th}{2})) \\ &\Longrightarrow \phi_1(th). \end{aligned}$$

From Lemma 2(1), $\phi_1(th) < 0$, hence $G^{'}(w_l) = \phi_1(th) < 0$.

Since $G^{'}(w_l) < 0$ and $G^{''}(w) > 0$, we have $G^{'}(w) < 0, \forall w \leq w_l$. Therefore, the unique critical point w_0 must be greater than w_l.

Next we show $w_0 < w_u$. When $th \geq \frac{1}{2}$, $w_0 < Q^{-1}(th) = w_u$. So we only need to show when $0 < th < \frac{1}{2}, w_0 < 0$. Similarly we first show $G^{'}(0) > 0$.

$$\begin{aligned} G^{'}(0) &= (1-th)s^{'}(0) - 1 + (s(0) - 0)g(0) \\ &= (1-th)\frac{g(0)}{g(Q^{-1}(\frac{1}{2} - th))} - 1 + Q^{-1}(\frac{1}{2} - th))g(0) = \phi_2(th), \end{aligned}$$

From Lemma 2(2), for $z \in (0, \frac{1}{2})$, $\phi_2(z) > 0$. Hence $G^{'}(0) > 0$.

Since for $th \in (0, \frac{1}{2})$, $G^{'}(0) > 0$ and $G^{''}(w) > 0$, we have $G^{'}(w) > 0, \forall w \geq 0$. Thus, for $th \in (0, \frac{1}{2})$, the unique critical point w_0 must be less than 0. Combined with $w_0 < Q^{-1}(th)$, we have $w_0 < min(0, Q^{-1}(th)) = w_u$.

We now show that w_0 is the unique local minimum of (w_l, w_u). It is easy to see that w_0 is a local minimum of (w_l, w_u) because w_0 is the global minimum. For uniqueness, we use contradiction. Assume that there is another local minimum $w_3 \in (w_l, w_u)$. Then it must satisfy $G^{'}(w_3) = 0$, which is contradictory to the fact that w_0 is the unique critical point in $(-\infty, Q^{-1}(th))$. ■

Proposition 4: (1) For $z \geq 0.86$, $H(z)$ is strictly convex;

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

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

$$H(z) = (1-z)s_0(z) - w_0(z) + g(w_0(z)) - g(s_0(z)) \tag{13}$$

From Proposition 3, w_0 is the unique critical point of $G(w)$, therefore, $w_0(z)$ satisfies

$$G'(w_0(z)) = (1-z)\frac{g(w_0(z))}{g(s_0(z))} - 1 + (s_0(z) - w_0(z))g(w_0(z)) = 0. \tag{14}$$

Using equations (13) (14) and implicit differentiation, we get

$$\begin{aligned} w_0'(z) &= \frac{-(1-z)s_0 g(w_0)}{g(w_0)g(s_0)[g(w_0) - g(s_0)] - w_0 g^2(s_0) + (1-z)s_0 g^2(w_0)} \qquad (15) \\ s_0'(z) &= \frac{g(w_0)[g(w_0) - g(s_0)] - w_0 g(s_0)}{g(w_0)g(s_0)[g(w_0) - g(s_0)] - w_0 g^2(s_0) + (1-z)s_0 g^2(w_0)} \\ H'(z) &= \frac{1-z}{g(s_0)} \\ H''(z) &= \frac{1}{g(s_0)}[\frac{(1-z)s_0[g^2(w_0) - g(w_0)g(s_0) - w_0 g(s_0)]}{g^2(w_0)g(s_0) - g^2(s_0)g(w_0) - w_0 g^2(s_0) + (1-z)s_0 g^2(w_0)} - 1] \end{aligned}$$

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

$$\frac{(1-z)s_0[g^2(w_0) - g(w_0)g(s_0) - w_0 g(s_0)]}{g(w_0)g(s_0)[g(w_0) - g(s_0)] - w_0 g^2(s_0) + (1-z)s_0 g^2(w_0)} > 1. \tag{16}$$

From Proposition 3,for $0 < z < 1$, $Q^{-1}(\frac{1+z}{2}) < w_0 < min(0, Q^{-1}(z))$, hence $s_0 > Q^{-1}(\frac{1-z}{2}) > 0$. Therefore, $g(w_0) - g(s_0) > 0$. Thus, the denominator of the left side in equation (16) is positive for any $z \in (0,1)$. Multiply it on both sides of equation (16), after some algebra operations, it suffices to prove for $z \geq 0.86$,

$$[g(s_0) - (1-z)s_0][g(w_0) + w_0] > g^2(w_0). \tag{17}$$

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

$$\begin{aligned} w_0 < Q^{-1}(z) &\implies g(w_0) < g(Q^{-1}(z)) \\ &\implies -w_0 - g(w_0) > -Q^{-1}(z) - g(Q^{-1}(z)) \\ &\qquad = g(Q^{-1}(z))\phi_3(z), \end{aligned} \tag{18}$$

and for $z \geq 0.86$,

$$Q^{-1}(z) \leq Q^{-1}(0.86) \approx -1.0803 \implies -Q^{-1}(z) - g(Q^{-1}(z)) > 0.$$

Similarly,

$$\begin{aligned} w_0 > Q^{-1}(\frac{1+z}{2}) &\implies s_0 > Q^{-1}(\frac{1-z}{2}) \geq 0 \implies g(s_0) < g(Q^{-1}(\frac{1-z}{2})) \\ &\implies -g(s_0) + (1-z)s_0 > -g(Q^{-1}(\frac{1-z}{2})) + (1-z)Q^{-1}(\frac{1-z}{2}) \\ &\qquad = g(Q^{-1}(\frac{1-z}{2}))\phi_4(z), \end{aligned} \tag{19}$$

and as shown in Lemma 2, $\phi_4(z)$ increases with z, hence for $z \geq 0.86$,

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

Combine equations 18 and 19, we have

$$\begin{aligned} [g(s_0) - (1-z)s_0][g(w_0) + w_0] &> g(Q^{-1}(\frac{1-z}{2}))\phi_4(z)g(Q^{-1}(z))\phi_3(z) \\ &= \phi_3(z)\phi_4(z)\phi_5(z)g^2(Q^{-1}(z)). \end{aligned} \tag{20}$$

Also,

$$w_0 < Q^{-1}(z) \leq Q^{-1}(0.86) < 0 \implies g^2(w_0) < g^2(Q^{-1}(z)). \tag{21}$$

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 show

$$\phi_3(z)\phi_4(z)\phi_5(z) > 1.$$

Further, because $\phi_5(z) \geq \frac{1}{2}, \forall z \in (0,1)$, it suffices to prove for $z \geq 0.86$,

$$\frac{1}{2}\phi_3(z)\phi_4(z) > 1. \quad (22)$$

From Lemma 2, $\phi_3(z)$ and $\phi_4(z)$ both increase with z, hence,

$$\begin{aligned}\phi_3(z) &\geq \phi_3(0.86) \approx 3.8537, \phi_4(z) \geq \phi_4(0.86) \approx 0.5388 \Longrightarrow \frac{1}{2}\phi_3(z)\phi_4(z) \\ &\geq \frac{1}{2}\phi_3(0.86)\phi_4(0.86) \approx 1.0382 > 1.\end{aligned}$$

(2)From equation (15), $s_0'(z) = \frac{g(w_0)[g(w_0)-g(s_0)]-w_0g(s_0)}{g(w_0)g(s_0)[g(w_0)-g(s_0)]-w_0g^2(s_0)+(1-z)s_0g^2(w_0)}$. As we show in the proof of Proposition 4(1), $g(w_0) > g(s_0)$. Therefore, both the numerator and the denominator in the equation are positive, hence $s_0'(z) > 0$.

Next we bound $H(z)$ in two steps.

(i) **Bounding** $H'(z)$

From equation (15), $H'(z) = \frac{1-z}{g(s_0)}$. 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 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 we have

$$L_i = \frac{1-\frac{i+1}{n}}{g(s_0(\frac{i}{n}))} < H'(z) = \frac{1-z}{g(s_0(z))} < \frac{1-\frac{i}{n}}{g(s_0(\frac{i+1}{n}))} = U_i, \forall z \in [\frac{i}{n}, \frac{i+1}{n}), \quad (23)$$

where L_i, U_i can be numerically computed.

(ii) **Bounding** $H(z)$

Let $z \in [\frac{i}{n}, \frac{i+1}{n})$, we have $H(z) = \int_0^z H'(z)dz = \sum_{j=0}^{i-1}\int_{\frac{j}{n}}^{\frac{j+1}{n}} H'(z)dz + \int_{\frac{i}{n}}^{z} H'(z)dz$, substitute equation (23), we have $\frac{\sum_{j=0}^{i-1} L_j\frac{1}{n}+L_i(z-\frac{i}{n})}{z} < \frac{H(z)}{z} < \frac{\sum_{j=0}^{i-1} U_j\frac{1}{n}+U_i(z-\frac{i}{n})}{z}$. Hence,

$$L_{1i} = min\{L_j, j = 0 \ldots i\} < \frac{H(z)}{z} < max\{U_i, j = 0 \ldots i\} = U_{1i}.$$

Further, because $H(z)$ is an increasing function, $L_{2i} = \frac{H(\frac{i}{n})}{\frac{i+1}{n}} < \frac{H(z)}{z} < \frac{H(\frac{i+1}{n})}{\frac{i}{n}} = U_{2i}$. In all, we have

$$max(L_{1i}, L_{2i}) < \frac{H(z)}{z} < min(U_{1i}, U_{2i}), \forall z \in [\frac{i}{n}, \frac{i+1}{n}) \tag{24}$$

We set $n = 10000$, then use equation (24) and compute that for $z \in [0, 0.99]$, $1.86z < H(z) < 2.52z$. ■

Proposition 5:(1) $0.929 \leq \frac{H(z)}{H_2(z)} \leq 1.26$;

(2) $H_2(z)$ is strictly convex.

Proof: (1) When $z \geq z_0$, $\frac{H(z)}{H_2(z)} = 1$. When $0 \leq z \geq z_0$, from Proposition 4(2), $1.86z < H(z) < 2.52z$, so $\frac{1.86z}{H_2(z)} \leq \frac{H(z)}{H_2(z)} \leq \frac{2.52z}{H_2(z)}$. Therefore, for $0 \leq z < 1$, $0.929 \leq \frac{H(z)}{H_2(z)} \leq 1.26$.

(2) We need to show that for $z_1 \neq z_2$, $H_2(\theta z_1 + (1-\theta)z_2) < \theta H_2(z_1) + (1-\theta)H_2(z_2)$. Without loss of generality, assume $z_1 < z_2$, it suffices to show that

$$\frac{H_2(\theta z_1 + (1-\theta)z_2) - H_2(z_1)}{(1-\theta)(z_2 - z_1)} < \frac{H_2(z_2) - H_2(\theta z_1 + (1-\theta)z_2)}{\theta(z_2 - z_1)}. \tag{25}$$

There are three cases:

- $z_1 < z_2 \leq z_0$ In this case $H_2(z) = H_1(z)$. Because $H_1(z)$ is strictly convex, hence (25) holds.
- $z_0 \leq z_1 < z_2$ In this case $H_2(z) = H(z)$. As shown in Proposition 4, $H(z)$ is strictly convex for $z \geq 0.86$. Hence (25) holds.
- $z_1 < z_0 < z_2$ Without loss of generality, suppose $\theta z_1 + (1-\theta)z_2 \leq z_0$. By Mean Value Theorem, in (25)

$$\begin{aligned} LHS &= \frac{H_2(\theta z_1 + (1-\theta)z_2) - H_2(z_1)}{(1-\theta)(z_2 - z_1)} \\ &= \frac{H_1(\theta z_1 + (1-\theta)z_2) - H_1(z_1)}{(1-\theta)(z_2 - z_1)} \\ &= \frac{H_1'(\zeta_1)(1-\theta)(z_2 - z_1)}{(1-\theta)(z_2 - z_1)} \\ &= H_1'(\zeta_1), \end{aligned}$$

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

$$\begin{aligned} RHS &= \frac{H_2(z_2) - H_2(\theta z_1 + (1-\theta)z_2)}{\theta(z_2 - z_1)} \\ &= \frac{H(z_2) - H(z_0) + H_1(z_0) - H_1(\theta z_1 + (1-\theta)z_2)}{z_2 - z_0 + z_0 - [\theta z_1 + (1-\theta)z_2]} \\ &= \frac{H'(\zeta_2)(z_2 - z_0) + H_1'(\zeta_3)[z_0 - (\theta z_1 + (1-\theta)z_2)]}{z_2 - z_0 + z_0 - [\theta z_1 + (1-\theta)z_2]}, \end{aligned}$$

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

We compute that $H_1'(z_0) \approx 2.0019 < H'(z_0) \approx 5.7241$. Since $H_1(z)$ is strictly convex, and $H(z)$ is strictly convex for $z \geq 0.86$. Therefore, we have

$$\begin{aligned} H'(\zeta_2) \geq H'(z_0) > H_1'(z_0) \geq H_1'(\zeta_1) \\ H_1'(\zeta_3) \geq H_1'(\theta z_1 + (1-\theta)z_2) \geq H_1'(\zeta_1) \end{aligned}$$

Therefore,

$$\begin{aligned} RHS &= \frac{H'(\zeta_2)(z_2 - z_0) + H_1'(\zeta_3)[z_0 - (\theta z_1 + (1-\theta)z_2)]}{z_2 - z_0 + z_0 - [\theta z_1 + (1-\theta)z_2]} \\ &> \frac{H_1'(\zeta_1)(z_2 - z_0) + H_1'(\zeta_1)[z_0 - (\theta z_1 + (1-\theta)z_2)]}{z_2 - z_0 + z_0 - [\theta z_1 + (1-\theta)z_2]} \\ &= H_1'(\zeta_1) = LHS. \end{aligned}$$

Similarly, we can prove (25) holds if $\theta z_1 + (1-\theta)z_2 \geq z_0$.

Hence (25) holds for all possible z, which shows $H_2(z)$ is strictly convex. ∎

References

[1] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," in *Proc. of ACM MOBICOM*, August 1999.

[2] J. Kahn, R. Katz, and K. Pister, "Next Century Challenges: Mobile Networking for "Smart Dust"," in *Proc. of ACM MOBICOM*, August 1999, also JCN vol 2 no 3, September 2000.

[3] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, "A taxonomy of wireless micro-sensor network models," *ACM Mobile Computing and Communication Review*, vol. 6, pp. 28–36, April 2002.

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

[5] G. Tolle, J. Polastre, R. Szewczyk, N. Turner, K. Tu, S. Burgess, D. Gay, P. Buonadonna, W. Hong, T. Dawson, and D. Culler, “A Macroscope in the Redwoods ,” in *Proc. of ACM SenSys*, 2005.

[6] 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.

[7] K. Srinivasan, M. Ndoh, H. Nie, H. Xia, K. Kaluri, and D. Ingraham, “Wireless Technologies for Condition-Based Maintenance (CBM) in Petroleum Plants,” *Proc. of the 1st International Conference on Distributed Computing in Sensor Systems (Poster Session)*, 2005.

[8] P. K. Varshney, “Distributed Detection and Data Fusion,” Springer, New York, 1997.

[9] 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.

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

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

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

[13] K. Romer, P. Blum, and L. Meier, “Time Synchronization and Calibration in Wireless Sensor Networks.” in *Ivan Stojmenovic (Ed.): Handbook of Sensor Networks: Algorithms and Architectures, John Wiley and Sons, ISBN 0-471-68472-4*, To appear September 2005.

[14] J. Elson, L. Girod, and D. Estrin, “Short Paper: A Wireless Time-Synchronized COTS Sensor Platform, Part I: System Architecture,” in *Proc. of the IEEE CAS Workshop on Wireless Communications and Networking*, September 2002.

[15] Q. Gao, K. J. Blow, and D. J. Holding, “Simple algorithm for improving time synchronization in wireless sensor networks,” *Electronics Letters*, vol. 40, pp. 889–891, July 2004.

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

[17] D. Cox, A. Milenkovic, and E. Jovanov, “Time Synchronization for ZigBee Networks,” in *Proc. of the 37th Southeastern Symposium on System Theory*, Tuskegee, AL, March 2005.

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

[19] 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.

[20] 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.

[21] “IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical layer (PHY) Specifications,” 1997.

[22] 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.

[23] 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.

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

[25] 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.

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

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

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

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

[30] K. Arisha, M. Youssef, and M. Younis, "Energy-Aware TDMA-Based MAC for Sensor Networks," in *Proc. of the IEEE Integrated Management of Power Aware Communications, Computing and Networking*, May 2002.

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

[32] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System Architecture Directions for Networked Sensors," in *Architectural Support for Programming Languages and Operating Systems*, 2000, pp. 93–104.

[33] http://www.tinyos.net/scoop/special/hardware.

[34] W. Ye and J. Heidemann, "Ultra-low duty cycle mac with scheduled channel polling," USC/Information Sciences Institute, Tech. Rep. ISI-TR-2005-604, July 2005. [Online]. Available: http://www.isi.edu/ johnh/PAPERS/Ye05a.html

[35] D. Chen and P.K.Varshey, "Qos support in wireless sensor netwoks: A survey," in *Proc. of the International Conference on Wireless Networks*, 2004.

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

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

[38] M. Perillo and W. Heinzelman, "Providing Application QoS through Intelligent Sensor Management," in *Proc. of IEEE SNPA*, 2003.

[39] S. Bhatnagar, B. Deb, and B. Nath, "Service differentiation in sensor networks," in *Proc. of the 4th International Symposium on Wireless Personal Multimedia Communications*, 2001.

[40] W.-P. Chen and L. Sha, "An energy-aware data-centric generic utility based approach in wireless sensor networks," in *Proc. of IEEE/ACM IPSN*, 2004.

[41] S. Basagni, "Distributed Clustering Algorithm for Ad-hoc Networks," in *International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, 1999.

[42] 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.

[43] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks," in *Proc. of IEEE INFOCOM*, April 2001.

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

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

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

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

[48] F. Koushanfar, N. Taft, and M. Potkonjak, "Sleeping coordination for comprehensive sensing using isotonic regression and domatic partitions," in *Proc. of IEEE INFOCOM*, to appear April 2006.

[49] 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.

[50] Y. Yao and J. E. Gehrke, "Query processing in sensor networks," in *Proc. of the 1st Biennial Conference on Innovative Data Systems Research*, 2003.

[51] A. Boukerche, R. W. N. Pazzi, and R. B. Araujo, "A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications," in *Proc. of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, 2004.

[52] N. Kingsbury, "Approximation formulae for the gaussian error integral, q(x)," http://cnx.rice.edu/content/m11067/2.4/, 2005.