Dynamic Sleep Time Control in Wireless Sensor Networks

XU NING
Microsoft Corporation
and
CHRISTOS G. CASSANDRAS
Boston University

21

Idle listening is a major source of energy waste in wireless sensor networks. It can be reduced through Low-Power Listening (LPL) techniques in which a node is allowed to sleep for a significant amount of time. In contrast to conventional fixed sleep time policies, we introduce a novel dynamic sleep time control approach that further reduces control packet energy waste by utilizing known data traffic statistics. We propose two distinct approaches to dynamically compute the sleep time, depending on the objectives and constraints of the network. The first approach provides a dynamic sleep time policy that guarantees a specified average delay at the sender node resulting from packets waiting for the end of a sleep interval at the receiver. The second approach determines the optimal policy that minimizes total energy consumed. In the case where data traffic statistics are unknown, we propose an adaptive learning algorithm to estimate them online and develop corresponding sleep time computation algorithms. Simulation results are included to illustrate the use of dynamic sleep time control and to demonstrate how it dominates fixed sleep time methods. An implementation of our approach on a commercial sensor node supports the computational feasibility of the proposed approach.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Distributed networks*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Protocol architecture*

General Terms: Algorithms, Design, Performance

Additional Key Words and Phrases: Energy management, low-power listening, stochastic processes, Learning

The research in this article was conducted while X. Ning was a student at Boston University.
The work is supported in part by NSF under grants DMI-0330171 and EFRI-0735974, by AFOSR under grants FA9550-04-1-0133 and FA9550-04-1-0208, and by DOE under grant DE-FG52-06NA27490.
Authors' addresses: X. Ning, Microsoft Corp., One Microsoft Way, Redmond, WA 98052; email: xuning@microsoft.com; C. G. Cassandras, Division of Systems Engineering, Boston University, 15 St. Mary's St., Brokline, MA 02446; email: cgc@bu.edu.

DOI 10.1145/1754414.1754417 http://doi.acm.org/10.1145/1754414.1754417

ACM Reference Format:
Ning, X. and Cassandras, C. G. 2010. Dynamic sleep time control in wireless sensor networks. ACM Trans. Sensor Netw. 6, 3, Article 21 (June 2010), 37 pages.
DOI = 10.1145/1754414.1754417 http://doi.acm.org/10.1145/1754414.1754417

1. INTRODUCTION

A Wireless Sensor Network (WSN) is a spatially distributed wireless network consisting of low-cost autonomous nodes that are mainly battery powered and have sensing and wireless communication capabilities [Megerian and Potkonjak 2003]. Usually, nodes in such a network share a common objective, such as environmental monitoring or event detection. Due to limited onboard power, nodes rely on short-range communication and form a multi-hop network to deliver information to a base station. Power consumption is a key issue in WSNs, since it directly impacts their lifespan in the likely absence of human intervention for most applications of interest.

Energy in WSN nodes is consumed by the CPU, by sensors/actuators, and by radio, with the last consuming the most [Shnayder et al. 2004]. In order to optimize energy usage, it is important to identify the major sources of waste in communication [Ye et al. 2004]. A *collision* occurs when a packet is corrupted by other simultaneous transmissions and requires a retransmission; *overhearing* arises when a node receives a packet that is not destined to it; *control packet overhead* is the energy cost incurred during sending and receiving control packets instead of actual data payload; finally, *idle listening*, the largest energy waste, occurs when a radio receiver remains listening during an idle period in the network. Energy waste due to idle listening can be reduced by adopting a Medium Access Control (MAC) scheme. Such schemes can be categorized into *scheduled* and *unscheduled*. Scheduled schemes, such as TDMA [Sohrabi et al. 2000] and S-MAC [Ye et al. 2004], maintain a schedule among a small cluster of nodes such that the nodes have coordinated transmission. Therefore, nodes can turn off their radio according to the schedule. Unscheduled schemes, on the other hand, try to emulate an “always-on” receiver by introducing additional ad hoc synchronization. One way to achieve this is to use Low-Power Listening (LPL), which has been adopted previously, for instance, in radio paging systems [Mangione-Smith 1995]. LPL uses a *preamble-channel polling* scheme to synchronize sender and receiver; detail will be given in Section 2. Unscheduled MACs using LPL include B-MAC [Polastre et al. 2004], WiseMAC [El-Hoiydi and Decotignie 2004] and X-MAC [Buettner et al. 2006], and so on. One obvious advantage of unscheduled MAC is its universality, since all transmission controls are transparent to the applications to which it just appears to be a normal, always-on radio. Another advantage is that it does not need advance synchronization (it can, though, benefit from it since a preamble can be shortened when the transmission pair is roughly synchronized).

In LPL, the key design problem is to determine the sleep time in the receiving node. Many MAC schemes that adopt LPL only use periodic sleep time control for its simplicity. In this article we propose a dynamic sleep time control scheme in order to address the control packet overhead problem induced by the preamble and channel polling in LPL. Depending on the tasks they perform, WSNs can be differentiated in terms of continuous monitoring or event-driven operation [Sichitiu 2004]. In continuous monitoring WSNs, data are acquired by sensors on a periodic basis. As an example, Hill and Culler [2002] describe an environmental monitoring application where a WSN collects and sends back temperature, light level, and humidity data over a vineyard. Under common assumptions, such periodic sleeping control can achieve a very high reduction in the duty cycle. In an event-driven WSN, network activities are triggered by external random events, such as the detection of human/wild life activity or other events. Since event times are not deterministic, many wake-ups actually take place during times when an event is not likely to happen, thus energy is wasted.

In this aricle, we propose a dynamic sleep time control scheme where the time between consecutive channel pollings is controlled using available statistical network traffic information as well as prior observations. We will show that if statistical information about event times is known, we can control the sleep time of the receiver so that it samples the channel more frequently when an event is more likely to happen, and less frequently when it is not. Along this line, we propose two dynamic sleep time control approaches (originally introduced in Ning and Cassandras [2006] and Ning and Cassandras [2008]).

(1) *Fixed Expected Preamble Duration*. We assume a desired expected preamble duration is given by design, and we dynamically compute the sleep time interval to ensure this is satisfied, using the interarrival time distribution information. Our numerical results show that this dynamic sleep time control dominates fixed sleep time control in the Pareto optimality sense.

(2) *Total Energy Minimization*. The objective is to minimize the link energy—the overhead energy consumed by control packets to transmit and receive a message. Exploiting the structure of the problem, we are able to formally derive both the necessary optimality conditions and an algorithm which is of low complexity and which provides an optimal sleep time control policy.

Both approaches hinge on the fact that the interarrival time distribution of external events is known. When this information is not present, we propose a probability distribution estimation scheme suitable for WSNs in terms of memory space and computation limits. This scheme is based on recursive quantile estimation [Tierney 1983] and a Stochastic Approximation (SA) algorithm [Robbins and Monro 1951; Kushner and Yin 2003]. Based on the data structure of the distribution, we derive corresponding algorithms for both approaches.

The article is organized as follows: in Section 2, we review pertinent literature regarding recent developments in LPL and their application to WSNs. In Section 3, we introduce the basic modeling framework and the problem of interest. In Sections 4 and 5, we detail the two proposed approaches to dynamically control the sleep time in LPL. In Section 6, we deal with the practical situation where statistical information is not known in advance, by proposing a learning and approximation algorithm. In Section 7 we develop implementations of the sleep time algorithms under the approximation. In Section 8, numerical results are provided and the conclusion is in Section 9.

2. LPL LITERATURE REVIEW

LPL is an approach to saving energy at the link level, which consists of a sender and a receiver. The main steps of LPL are as follows.

(1) The receiver remains at a sleep state most of the time, and occasionally wakes up to sample the channel in order to determine whether it is busy or idle.
(2) When the sender wants to send a message to the receiver, it begins with an attached signal called the preamble. The preamble can be viewed as a wake-up signal. After the preamble signal, the sender sends the message.
(3) When the receiver wakes up and samples the channel, either of two cases may occur: (1) If the channel is idle, the receiver sets the next wake-up time and sleeps again, (2) If the channel is busy (preamble detected), the receiver stays on until the message is received. After transmission, the receiver sets its next wake-up time and sleeps again.

The application of LPL in WSNs has the following three basic implementations.

(1) *Plain-vanilla LPL*. This is the simplest LPL, where both the receiver's sleep period and the sender's preamble length are fixed. Usually the preamble length is a little bit longer than the sleep period to ensure that the preamble is picked up by one of the receiver's channel pollings. B-MAC, proposed in the seminal work Polastre et al. [2004], uses this type of LPL and has implemented it in TinyOS 1.0. Since B-MAC was designed for earlier bit-streaming radio chips such as Chipcon CC1000, which supports long preamble durations, it cannot be directly applied to recent packet radios, such as the IEEE 802.15.4-compliant Chipcon CC2420. Later work such as Virtual Preamble Cross-Checking (VPCC) [Moon et al. 2007] and B-MAC+ [Avvenuti et al. 2006] have implemented LPL for packet radios. While being the simplest one to implement, in plain-vanilla LPL the sender has to transmit the full preamble due to lack of handshaking or other means to shorten the preamble. Therefore, generally the overhead introduced by long preamble and/or frequent channel polling is large compare to other LPL implementations.
(2) *Short preamble LPL*. In this version, the receiver's sleep period remains fixed. However, there exists some degree of clock synchronization between the sender and the receiver so the sender can estimate within some short interval when the receiver will wake up. Hence, the sender can send a shorter preamble as long as it is long enough to cover the estimated interval. This version is used in WiseMAC [El-Hoiydi and Decotignie 2004].
(3) *Variable preamble LPL*. Although the short preamble version also varies the length of the preamble, it requires clock synchronization between the

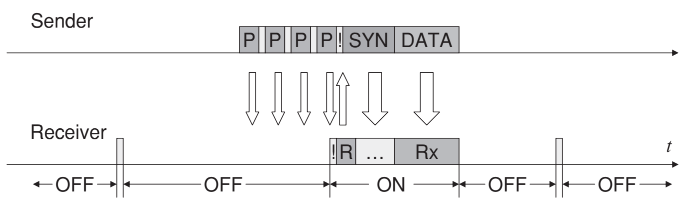

Fig. 1. Low-power listening (LPL) with variable preamble. After each preamble packet P is sent, the sender listens for a brief period for acknowledgement from the receiver, then sends P again if no acknowledgement is received.

two peers. In variable preamble LPL, there is no need for such synchronization. Instead, handshaking is used. As shown in Figure 1, while similar to the packet train implementation in plain-vanilla LPL, variable preamble LPL inserts short listening intervals between consecutive preamble packets. When the receiver wakes up and picks up a preamble signal, it replies to the sender so a handshake is formed. Then, the sender knows for sure the receiver is awake in receiving mode, so the message will be transmitted immediately instead of after the whole preamble. Implementation details can be found in Joe and Ryu [2007], Mahlknecht and Bock [2004] and Buettner et al. [2006] (X-MAC). An important feature of variable preamble LPL is that it does not require a fixed, predetermined sleep time control at all, due to the handshake, hence it is truly asynchronous. This key feature enables us to control the sleep time more flexibly and save more energy.

All the aforementioned implementations of LPL use fixed sleep time. Jurdak et al. [2005] and Jurdak et al. [2007] combine the MAC layer LPL and routing schemes, proposing Energy-Aware Adaptive LPL. In this scheme, the LPL sleeping periodicity is determined by the load imposed at each node, and is adaptive as the routing topology changes. To the best of our knowledge this work is the closest to our problem of interest in that it recognizes the need to select the sleep time so as to conserve energy. However, in Jurdak et al. [2007], nodes assume a fixed sleep time policy where the sleep time is considered a parameter determined by the work load. In our work, we allow the sleep time to dynamically change in each sleep interval according to how likely the sender is to send a message to the receiver. Intuitively, this can reduce energy consumption because, when the incoming transmission is unlikely to occur in the near future, the receiver does not need to sample the channel as frequently and therefore conserves its own energy. On the other hand, when the incoming transmission is highly likely to occur in the near future, the receiver should sample the channel more frequently so that the preamble duration can be shortened so the sender's energy is conserved. Essentially, more information regarding the incoming transmission time is used in the sleep time decision. Therefore, dynamic sleep time control is superior to fixed sleep time control.

3. PROBLEM DESCRIPTION AND MODELING FRAMEWORK

We consider a link in a WSN which consists of a sender and a receiver. The sender is a wireless node equipped with some event detector driven by external, random events such as body movement in a room or fire alarms. When the sender detects an event or reports its status, it sends a message to the receiver, which is a downstream node in the network. In this link, variable preamble LPL is used. To simplify the analysis, we make the following assumptions: (1) Generally, the interarrival time of the events is much larger than preamble durations, the time to transmit a message and the duration of a channel sampling activity. Therefore, we model channel samplings as points in time. (2) In case of multiple events occurring during one preamble duration, they are immediately transmitted after the first message that initiates the preamble. (3) Although in the variable preamble LPL a sequence of discrete preamble packets is sent, we model the preamble as a continuous signal whose duration is a real positive number. In addition, we focus on the preamble and the channel sampling activities and ignore the energy cost incurred during the handshaking and the transmission of the data payload part of the message, since it is not controllable in the scope of our optimization problem.

We define the *ith event* to be the instant when the transmission of the ith message's preamble starts. A_i is the occurrence time of the ith event and $T_i = A_i - A_{i-1}$ ($T_1 = A_1$) is the ith *interevent* time. We assume T_i is a random variable whose distribution, conditioned on all previous history, is specified by a conditional cumulative distribution function, (cdf) $F_i(t) = \Pr(T \leq t|\mathcal{F}_{i-1})$, and a probability density function (pdf), $f_i(t)$, where $\mathcal{F}_i$ is a filtration generated by $\{T_1, T_2, \ldots, T_i\}$ and $\mathcal{F}_0 = \emptyset$. Let t_n be the nth wake-up time of the receiver. D_i is the preamble duration of message i, which is also the delay of message i. Because A_i is random, D_i is a random variable also.

The central problem in LPL consists of determining when the receiver should wake up and sample the channel: a sequence $\{t_n, n \geq 1\}$ of wake up times. Clearly this is a trade-off between the sender and the receiver. On one hand, as the receiver wakes up more often, energy depletes faster because each wake up needs to turn the radio on and off, where some energy cost is incurred. Denote this energy cost by c. On the other hand, if the sleep time between two receiver sampling events is long, the sender will obviously need to send a longer preamble so that it can be picked up by the receiver in its next wake-up. Sending a preamble also incurs an energy cost, which is proportional to the length of time of the preamble. For simplicity, let the energy cost per unit time of the preamble be 1. Moreover, the message experiences some latency. Hence, the problem is to determine how the receiver should control its sleep time so that the total energy spent in each message transmission is minimized, subject to a constraint in delay.

To simplify notation, we introduce the following single-message model, illustrated in Figure 2. The random time until the next message is denoted by T, the preamble duration by D, and the distribution is $F(\cdot)$, dropping the index i throughout. Redefine $\{t_k, k \geq 0\}$ so that t_0 is the time instant at which the last received message has finished transmitting, and t_k is the kth wake-up since

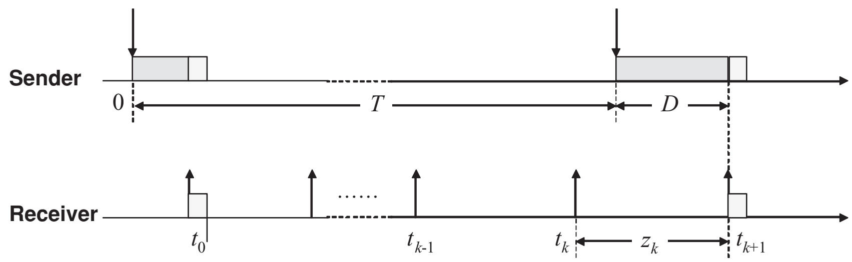

Fig. 2. The single-message model, which only focuses on the very next incoming message. All time epochs are religned with respect to the last event.

the last event. Define sleep time as $z_k = t_{k+1} - t_k$, which will be determined at time t_k. This model allows us to focus only on the very next message. When this message, say A_i, is received, we roll over the single-message model with $F(\cdot)$ replaced by a new $F_{i+1}(\cdot)$.

In fixed sleep time control, $z_k \equiv \bar{Z}$ which is a constant, and $t_k = k\bar{Z} + t_0$. However, in dynamic sleep time control, we adopt a policy $z_k = z(t_k)$, a function of the time t_k elapsed since the last event. In the following two sections, two approaches to determining a policy of the general form $z(t)$ are proposed.

4. DYNAMIC SLEEP TIME CONTROL WITH FIXED EXPECTED PREAMBLE DURATION

4.1 Problem Formulation and Analysis

In WSNs, the senders are usually low powered sensors. It is common to place stringent requirements on the sender's energy consumption, which is proportional to the preamble duration in our case. This section proposes an approach to compute the sleep time such that a desirable average preamble duration $\bar{D}$ is maintained. Because $\bar{D}$ can be viewed as a constraint both on delay and energy placed by the sender, the dynamic sleep time approach proposed in this section allows the receiver to calculate the sleep time so that its own energy is smartly spent while not violating the sender's constraint.

Consider some k in Figure 2 where the kth channel polling has just occurred and the channel is still idle. Knowing that the event (message) has not occurred so far, for any given sleep time z_k, we can evaluate the expectation of D conditioned on the fact that the event occurs before the next wake up at time $t_k + z_k$:

$$\begin{aligned} E\left[D|t_k < T \leq t_k + z_k\right] &= t_k + z_k - E\left[T\,|t_k < T \leq t_k + z_k\right] \\ &= t_k + z_k - \frac{\int_{t_k}^{t_k+z} \tau dF(\tau)}{F\left(t_k + z_k\right) - F\left(t_k\right)}, \end{aligned} \tag{1}$$

which is defined for z_k such that $F(t_k + z_k) - F(t_k) > 0$. Therefore, given some desired average preamble duration $\bar{D} > 0$, we can establish a condition:

$$\bar{D} = E\left[D|t_k < T \leq t_k + z_k\right], \tag{2}$$

and seek a solution z^* in (1) which will be a sleep time satisfying this condition. Because the right-hand side in Equation (1) involves the interarrival distribution $F(\cdot)$, the arrival statistics are utilized. If the next arrival is unlikely to occur over a short time horizon, the sleep time will be longer. To see this, suppose that the probability of an event occurring is very small for time interval $(t_k, t_k + s]$ for some $s > 0$. Because the expectation in (2) must equal to $\bar{D}$, z_k will be larger than s, as the interval contributes little to the expectation. This adaptation helps us reduce unnecessary channel sampling and hence energy cost. Last, it is observed that if $E[D]$ is the unconditional expectation of the delay, then $E[D] = \bar{D}$. To see this, suppose it takes N channel polling attempts to finally receive the message, and observe that the expected delay conditioned on N is:

$$E\left[D|N\right] = E\left[D|t_{N-1} < T \le t_N\right] = \bar{D},$$

and therefore,

$$E\left[D\right] = E\left[E\left(D|N\right)\right] = \bar{D}. \tag{3}$$

As we will see in the numerical examples in Section 8, this sleep time control approach lowers the total energy consumption of the system. However, it does not attempt to minimize it; it only dynamically selects sleep times that meet a specific average delay given by $\bar{D}$. The choice of $\bar{D}$ is by design and usually determined by the sender's energy requirements, as the senders are typically end-point sensors, which have less resources than the receivers, which could be more powerful cluster-head nodes.

4.2 Characterizing the Solution

In order to construct our sleep time control policy, $z(t)$, we have to solve Equation (2) for all possible values of t_k. Therefore, we parameterize Equation (2) by t, replacing t_k by t and z_k by $z(t)$:

$$\bar{D} = t + z(t) - \frac{\int_t^{t+z(t)} \tau dF(\tau)}{F(t + z(t)) - F(t)}. \tag{4}$$

Equation (4) is not trivial to solve for $z(t)$ because it involves an integral over an arbitrary $F(\cdot)$. However, assuming $F(\tau)$ is differentiable with the pdf denoted by $f(\tau)$, and assuming that $F(s) - F(t) > 0$ for all $s > t$ within the support region of $F(\cdot)$, we can see in what follows how the sleep time policy $z(t)$ can be specified through an ordinary differential equation (ODE). For notational simplicity, define:

$$u(t) = t + z(t).$$

We can then rewrite equation (4) as:

$$\int_t^{u(t)} \tau f(\tau) d\tau = (u(t) - \bar{D})(F(u(t)) - F(t)).$$

Differentiating both sides with respect to t and solving for $du(t)/dt$ gives:

$$\frac{du(t)}{dt} = \frac{\left(u(t) - t - \bar{D}\right) f(t)}{F(u(t)) - F(t) - \bar{D} f(u)}. \tag{5}$$

Solving (5), we can construct a sleep time policy $z(t) = u(t) - t$ such that the expected preamble duration is $\bar{D}$. To solve (5), we still need an initial condition, obtained, for example, by solving (4) for $t = 0$. Note that the evaluation of $z(t)$ need only be performed once; from an implementation standpoint it can then be stored in memory.

4.3 Special Cases: Exponential and Uniform Interarrival Time Distributions

4.3.1 *Exponential Distribution.* For an exponential distribution with rate λ, (1) is rewritten as:

$$\bar{D} = t + z(t) - \frac{\int_t^{t+z(t)} \tau \lambda e^{-\lambda \tau} d\tau}{e^{-\lambda t} - e^{-\lambda(t+z(t))}}, \tag{6}$$

which yields:

$$\bar{D} = \frac{e^{-\lambda z(t)} + \lambda z(t) - 1}{\lambda \left[1 - e^{-\lambda z(t)}\right]}. \tag{7}$$

Solving (7) for $z(t)$, we can see that the sleep time does not depend on t, as expected, due to the memoryless property. Therefore, there exists a value $Z(\lambda, \bar{D})$ such that $z(t) \equiv Z(\lambda, \bar{D})$ is the solution to (7) and dynamic control reduces to fixed sleep time control. The exponential distribution can be viewed as the worst case in the sense that we cannot benefit from knowledge of past arrivals. Nonetheless, (7) establishes a relationship between the expected delay, $\bar{D}$, of every message and the sleep time control applied. However, (7) does not have a closed-form solution and can only be solved using numerical methods.

4.3.2 *Uniform Distribution.* Consider a uniform distribution $U[a, b]$, where $0 \leq a < b$. In this case, using fixed sampling will waste energy polling the channel while it is not possible to have an arrival ($0 \leq t \leq a$). An analytical solution to Equation (1) is obtainable by considering two cases:

Case 1: $0 \leq t \leq a$. Equation (1) is rewritten as:

$$\begin{aligned} \bar{D} &= t + z(t) - \frac{\int_a^{t+z(t)} \tau (b-a)^{-1} d\tau}{F(t + z(t))} \\ &= \begin{cases} \frac{1}{2}(t + z(t) - a) & t + z(t) \leq b \\ t + z(t) - \frac{1}{2}(a+b) & t + z(t) > b \end{cases}. \end{aligned} \tag{8}$$

Note that for any positive $\bar{D}$, $t + z(t) > a$, otherwise the right-hand side is not defined. The solution in this case is given by:

$$z(t) = \begin{cases} 2\bar{D} + a - t & \bar{D} \leq \frac{1}{2}(b-a) \\ \bar{D} + \frac{1}{2}(a+b) - t & \bar{D} > \frac{1}{2}(b-a) \end{cases}. \tag{9}$$

Case 2: $a \le t \le b$. In this case, (1) is rewritten as:

$$\bar{D} = t + z(t) - \frac{\int_t^{t+z(t)} \tau \,(b-a)^{-1}\, d\tau}{F\,(t + z\,(t)) - (t-a)\,(b-a)^{-1}}$$
$$= \begin{cases} \frac{1}{2} z\,(t) & t + z\,(t) \le b \\ z\,(t) - \frac{1}{2}\,(b-t) & t + z\,(t) > b \end{cases}. \tag{10}$$

The solution is:

$$z\,(t) = \begin{cases} 2\bar{D} & \bar{D} \le \frac{1}{2}\,(b-t) \\ \bar{D} + \frac{1}{2}\,(b-t) & \bar{D} > \frac{1}{2}\,(b-t) \end{cases}. \tag{11}$$

Letting $v = \max(t, a)$, the two cases can be consolidated as one:

$$z\,(t) = \begin{cases} 2\bar{D} + v - t & \bar{D} \le \frac{1}{2}\,(b-v) \\ \bar{D} + \frac{1}{2}\,(b-t) + \frac{1}{2}\,(v-t) & \bar{D} > \frac{1}{2}\,(b-v) \end{cases}. \tag{12}$$

5. DYNAMIC SLEEP TIME CONTROL WITH TOTAL ENERGY MINIMIZATION

5.1 Problem Formulation and Analysis

From an energy standpoint, the approach in the previous section does not attempt to minimize energy consumption. The energy consumed by LPL overhead consists of the preamble cost and the channel sampling cost. Assuming the sampling time sequence is $\{t_k, k = 1, 2, 3, \ldots\}$, the total cost is

$$Q = cN + D,$$

where $N = \arg\min_k t_k > T$ and $D = t_N - T$, that is, N is the number of samplings needed to receive the message and D is the duration of the preamble. Because the arrival time of the message T is a random variable, D and N are also random. It is difficult to formulate a static optimization problem to minimize Q over all possible $\{t_k, k = 1, 2, 3, \ldots\}$ because the number of variables can potentially be infinite.

Nevertheless, we can consider a different perspective. Let t be the age of the last event: no additional event has occurred for the past t time. Let t be the state and define $J(t)$ as the minimum expected cost-to-go to receive the message:

$$J\,(t) = \min_{z>0} V\,(t, z)\,,$$

where $V(t, z)$ is the expected cost-to-go at event age t and with sleep time z. Hence, $V(t, z)$ should include: (1) a fixed cost c to wake up at time $t + z$; (2) the expected preamble cost if a message arrival event occurs during the sleep time, or equivalently, before $t + z$, which is:

$$E[t + z - T\,|t \le T \le t + z],$$

and (3) the future minimum expected cost to receive the message if it does not occur during the sleep time, which is $J(t + z)$. Let the probability of the event occurring before the receiver waking up at $t+z$ be $P_1(t,z)$; we have:

$$P_1(t,z) = \Pr(T \le t+z | T > t) = \frac{F(t+z) - F(t)}{1 - F(t)} \tag{13}$$

and its complement:

$$P_2(t,z) = \Pr(T > t+z | T > t) = \frac{1 - F(t+z)}{1 - F(t)}. \tag{14}$$

Therefore,

$$V(t,z) = c + E\left[t+z-T | t \le T \le t+z\right] P_1(t,z) + J(t+z) P_2(t,z),$$

so the Bellman equation is:

$$J(t) = \min_{z>0}\{c + E\left[t+z-T | t \le T \le t+z\right] P_1(t,z) + J(t+z) P_2(t,z)\}. \tag{15}$$

For a distribution with a bounded support region $0 < T \le T_{\max}$, the allowable control set is limited to $0 < z \le T_{\max} - t$ and, in this case, the boundary condition for (15) is $J(T_{\max}) = c$. By solving the Bellman equation (15) for all possible $t > 0$, one can obtain the optimal sleep time control policy $z^*(t)$.

5.2 Characterizing the Optimal Solution

In this section we characterize the optimal solution to (15). In the following analysis, we assume that $F(\tau)$ has a finite support $0 \le \tau \le T_{\max}$, and $f(\tau)$ has a continuous derivative.

For notational simplicity, define $u = t + z$, which is the wake-up time epoch of the receiver determined at time t. We can redefine P_1 and P_2 from (13) and (14) using u:

$$P_1(t,u) = \frac{F(u) - F(t)}{1 - F(t)}, \quad P_2(t,u) = \frac{1 - F(u)}{1 - F(t)}$$

and rewrite the Bellman equation (15) as:

$$J(t) = \min_{t<u\le T_{\max}} \{c + E\left[u - T | t \le T \le u\right] P_1(t,u) + J(u) P_2(t,u)\}. \tag{16}$$

Define

$$\begin{aligned} G(t,u) &= c + E\left[u - T | t \le T \le u\right] P_1(t,u) + J(u) P_2(t,u) \\ &= c + \frac{\int_t^u (u-\tau) f(\tau)\, d\tau}{F(u) - F(t)} \cdot \frac{F(u) - F(t)}{1 - F(t)} + J(u) \frac{1 - F(u)}{1 - F(t)} \\ &= c + \frac{\int_t^u (u-\tau) f(\tau)\, d\tau + J(u)(1 - F(u))}{1 - F(t)} \end{aligned} \tag{17}$$

and we can write:

$$J(t) = \min_{t<u\le T_{\max}} G(t,u).$$

Let the optimal policy be $u^*(t)$:

$$u^*(t) = \arg \min_{t<u\le T_{\max}} G(t,u),$$

which is a function of t. Because $u^*(t)$ is the solution to (16), we have:

$$J(t) = G\big(t, u^*(t)\big). \tag{18}$$

Before proceeding, we prove the following two lemmas:

LEMMA 5.1. *Let $0 \le t_1 < t_2 \le T_{\max}$. As long as there exists a measurable subset S of $[t_1, t_2]$ such that $\forall \tau \in S$, $f(\tau) \neq 0$, then $u^*(t_1) \le u^*(t_2)$.*

PROOF. First, $u^*(t) = \arg\min_{t<u\le T_{\max}} G(t,u) > t$. Hence, if $u^*(t_1) \le t_2$, then it is trivially true that $u^*(t_1) < u^*(t_2)$. If $u^*(t_1) > t_2$, we can prove the result by contradiction. To simplify notation, let $u_1 = u^*(t_1)$ and $u_2 = u^*(t_2)$. Suppose $t_1 < t_2 < u_2 < u_1$. Because both u_1 and u_2 are minimizers, we have:

$$G(t_1, u_1) \le G(t_1, u_2) \text{ and } G(t_2, u_2) \le G(t_2, u_1).$$

Using (17) in these inequalities, gives:

$$\int_{t_1}^{u_1} (u_1 - \tau) f(\tau) d\tau + J(u_1)(1 - F(u_1)) \le \int_{t_1}^{u_2} (u_2 - \tau) f(\tau) d\tau + J(u_2)(1 - F(u_2)) \tag{19}$$

$$\int_{t_2}^{u_2} (u_2 - \tau) f(\tau) d\tau + J(u_2)(1 - F(u_2)) \le \int_{t_2}^{u_1} (u_1 - \tau) f(\tau) d\tau + J(u_1)(1 - F(u_1)). \tag{20}$$

Adding the two inequalities, we can cancel the terms involving $J(\cdot)$, obtaining:

$$\int_{t_1}^{u_1} (u_1 - \tau) f(\tau) d\tau + \int_{t_2}^{u_2} (u_2 - \tau) f(\tau) d\tau \le \int_{t_1}^{u_2} (u_2 - \tau) f(\tau) d\tau + \int_{t_2}^{u_1} (u_1 - \tau) f(\tau) d\tau, \tag{21}$$

which implies

$$\int_{t_1}^{t_2} (u_1 - \tau) f(\tau) d\tau \le \int_{t_1}^{t_2} (u_2 - \tau) f(\tau) d\tau,$$

that is

$$\int_{t_1}^{t_2} (u_1 - u_2) f(\tau) d\tau \le 0$$

Because $u_1 > u_2$, this implies that $f(\tau) = 0$ for $t_1 < \tau < t_2$ in order to satisfy the inequality, which contradicts the fact that there exists a measurable subset S of $[t_1, t_2]$ such that for $\tau \in S$, $f(\tau) \neq 0$. □

LEMMA 5.2. *Assuming that $|f'(t)| < L < \infty$ for $0 \le t \le T_{\max}$, there exists $\delta > 0$ such that for all $t \in [T_{\max} - \delta, T_{\max}]$, $u^*(t) = T_{\max}$.*

PROOF. From (16), we know that $J(t) \ge c$ for all $0 \le t \le T_{\max}$. Thus, from (17), we have:

$$J(t) \ge c + \frac{\int_t^{u^*(t)} (u^*(t) - \tau) f(\tau) d\tau + c(1 - F(u^*(t)))}{1 - F(t)}.$$

Define a function $H(t, u)$ as:

$$H(t, u) = c + \frac{\int_t^u (u - \tau) f(\tau) d\tau + c(1 - F(u))}{1 - F(t)}.$$

Recalling the definition of $G(t, u)$ in (17), and since $F(T_{\max}) = 1$, observe that:

$$H(t, u) \leq G(t, u)$$
$$H(t, T_{\max}) = G(t, T_{\max}).$$

Because $J(t) = \min_{t<u\leq T_{\max}} G(t, u)$ and $G(t, u)$ is bounded from below by $H(t, u)$, if we can prove that, for some $\delta > 0$, for all $t \in [T_{\max} - \delta, T_{\max}]$, $H(t, u)$ attains its minimum at $H(t, T_{\max})$, then $H(t, T_{\max})$ will also be the minimum of $G(t, T_{\max})$ and the optimal control is $u^*(t) = T_{\max}$. Therefore, to establish this fact, we take the partial derivative of $H(t, u)$ with respect to u and use the mean value theorem:

$$\frac{\partial H(t, u)}{\partial u} = \frac{\int_t^u f(\tau) d\tau - cf(u)}{1 - F(t)} = \frac{f(\xi)(u - t) - cf(u)}{1 - F(t)}, \tag{22}$$

where $t \leq \xi \leq u \leq T_{\max}$. Using a Taylor expansion of $f(\cdot)$ at u: $f(\xi) = f(u) - f'(u)(u - \xi) + o(u - \xi)$, we can then rewrite (22) as:

$$\begin{aligned}\frac{\partial H(t, u)}{\partial u} &= \frac{\left[f(u) - f'(u)(u - \xi) + o(u - \xi)\right](u - t) - cf(u)}{1 - F(t)} \\ &= \frac{(u - t - c) f(u) - f'(u)(u - \xi)(u - t) + o(u - \xi)(u - t)}{1 - F(t)}\end{aligned}$$

Because $|f'(u)| \leq L$ and $(u-\xi)(u-t) = o(u-t)$, we have the following inequality:

$$\begin{aligned}\frac{\partial H(t, u)}{\partial u} &\leq \frac{(u - t - c) f(u) + L \cdot o(u - t)}{1 - F(t)} \\ &\leq \frac{(T_{\max} - t - c) f(u) + o(T_{\max} - t)}{1 - F(t)}.\end{aligned}$$

Because $T_{\max}$ is the boundary of the support region of $f(\cdot)$, there must exist $\delta_0 > 0$ such that for $T_{\max} - \delta_0 \leq u < T_{\max}$, $f(u) > 0$. Therefore, there must also exist $\delta_1 > 0$ such that for $T_{\max} - \delta_1 \leq t < T_{\max}$, we have $T_{\max} - t \leq \delta_1 < c$ and

$$\frac{\partial H(t, u)}{\partial u} \leq \frac{(T_{\max} - t - c) f(u) + o(T_{\max} - t)}{1 - F(t)} < 0.$$

Letting $\delta = \min\{\delta_0, \delta_1\}$, we know that for $T_{\max} - \delta \leq t < T_{\max}$, $\partial H(t, u)/\partial u$ is strictly negative for all $t < u \leq T_{\max}$. Hence,

$$\min_{t<u\leq T_{\max}} H(t, u) = H(t, T_{\max}).$$

Therefore, $J(t) = G(t, T_{\max}) = H(t, T_{\max})$ and thus $u^*(t) = T_{\max}$. This proves the lemma. □

From Lemmas 5.1 and 5.2, we can conclude that either $u^*(t) = T_{\max}$ for $0 \leq t \leq T_{\max}$, or there exists $0 < t_0 \leq T_{\max}$ such that:

$$\begin{cases} u^*(t) < T_{max} & \text{for } 0 \leq t < t_0 \quad \text{(23a)} \\ u^*(t) = T_{max} & \text{for } t_0 \leq t \leq T_{max}. \quad \text{(23b)} \end{cases}$$

Since the allowable control set is such that $t < u^*(t) \leq T_{\max}$ for any t, and $u^*(t)$ is the minimizer of $G(t, u)$, the following necessary optimality conditions must be satisfied:

$$\begin{cases} \left.\dfrac{\partial G(t,u)}{\partial u}\right|_{u=u^*(t)} = 0 \quad \text{for } t < u^*(t) < T_{\max} & \text{(24a)} \\ \left.\dfrac{\partial G(t,u)}{\partial u}\right|_{u=u^*(t)} \leq 0 \quad \text{for } u^*(t) = T_{\max}, & \text{(24b)} \end{cases}$$

where, using (17),

$$\begin{aligned} \frac{\partial G(t,u)}{\partial u} &= \frac{F(u) + uf(u) - F(t) - uf(u) + J'(u)(1 - F(u)) - J(u) f(u)}{1 - F(t)} \\ &= \frac{F(u) - F(t) + J'(u)(1 - F(u)) - J(u) f(u)}{1 - F(t)}, \end{aligned} \tag{25}$$

where $J'(u)$ is the first derivative of $J(t)$ at u.

We first consider the case where $t < u^*(t) < T_{\max}$. Since (24a) holds in this case, we can rearrange (25) and establish the following equation:

$$F\left(u^*(t)\right) + J'\left(u^*(t)\right)\left(1 - F\left(u^*(t)\right)\right) - J\left(u^*(t)\right) f\left(u^*(t)\right) = F(t). \tag{26}$$

Differentiating (26) with respect to t on both sides, we have:

$$\frac{dF(u^*(t))}{dt} + \frac{d}{dt}\left[J'\left(u^*(t)\right)\left(1 - F\left(u^*(t)\right)\right)\right] - \frac{d}{dt}\left[J\left(u^*(t)\right) f\left(u^*(t)\right)\right] = f(t). \tag{27}$$

Considering each term on the left-hand side, we have:

$$\frac{dF(u^*(t))}{dt} = f\left(u^*(t)\right)\frac{du^*(t)}{dt}$$

$$\begin{aligned} \frac{d}{dt}\left[J'\left(u^*(t)\right)\left(1 - F\left(u^*(t)\right)\right)\right] &= \left[J''\left(u^*(t)\right)\left(1 - F\left(u^*(t)\right)\right)\right. \\ &\quad \left. - J'\left(u^*(t)\right) f\left(u^*(t)\right)\right]\frac{du^*(t)}{dt} \end{aligned}$$

$$\frac{d}{dt}\left[J\left(u^*(t)\right) f\left(u^*(t)\right)\right] = \left[J'\left(u^*(t)\right) f\left(u^*(t)\right) + J\left(u^*(t)\right) f\left(u^*(t)\right)\right]\frac{du^*(t)}{dt}.$$

Therefore, from (27), we derive the following expression for $du^*(t)/dt$:

$$\begin{aligned} \frac{du^*(t)}{dt} = & f(t)\left[f\left(u^*(t)\right) + J''\left(u^*(t)\right)\left(1 - F\left(u^*(t)\right)\right)\right. \\ & \left. - 2J'\left(u^*(t)\right) f\left(u^*(t)\right) - J\left(u^*(t)\right) f'\left(u^*(t)\right)\right]^{-1} \end{aligned} \tag{28}$$

for $t < u^*(t) < T_{\max}$. We now need $J'(\cdot)$ and $J''(\cdot)$ in order to compute (28). First, recalling that $J(t) = G(t, u^*(t))$, we can again differentiate with respect to t on both sides and obtain:

$$\frac{dJ(t)}{dt} = \left.\frac{\partial G(t,u)}{\partial t}\right|_{u=u^*(t)} + \left(\left.\frac{\partial G(t,u)}{\partial u}\right|_{u=u^*(t)}\right)\frac{du^*(t)}{dt}, \tag{29}$$

where (24a) still holds. Therefore,

$$\begin{aligned}\frac{dJ(t)}{dt} &= \frac{\partial}{\partial t}\left[c + \frac{\int_t^u (u-\tau) f(\tau) d\tau}{1-F(t)} + J(u)\frac{1-F(u)}{1-F(t)}\right]\Bigg|_{u=u^*(t)} \\ &= \frac{t-u^*(t)}{1-F(t)} f(t) + \frac{f(t)}{\left[1-F(t)\right]^2}[u^*(t)\left(F\left(u^*(t)\right) - F(t)\right) \\ &\quad - \int_t^{u^*(t)} \tau f(\tau) d\tau + J\left(u^*(t)\right)\left(1-F\left(u^*(t)\right)\right)]. \end{aligned} \tag{30}$$

Using (17) and (18), the second term in the right-hand-side is:

$$\frac{f(t)}{1-F(t)}\left[J(t) - c\right].$$

Therefore,

$$\frac{dJ(t)}{dt} = \frac{f(t)}{1-F(t)}\left[t - u^*(t) + J(t) - c\right]. \tag{31}$$

To compute $J''(t)$, we take the derivative of (31) with respect to t again:

$$\begin{aligned}\frac{d^2J(t)}{dt^2} &= \frac{\left[f'(t)(1-F(t)) - f^2(t)\right](t - u^*(t) + J(t) - c)}{(1-F(t))^2} \\ &\quad + \frac{f(t)\left(1 - u^{*'}(t) + J'(t)\right)}{1-F(t)}.\end{aligned}$$

We can substitute (31) into the first term and obtain:

$$\begin{aligned}\frac{d^2J(t)}{dt^2} &= \frac{\left[f'(t)(1-F(t)) - f^2(t)\right]J'(t)}{(1-F(t))f(t)} + \frac{f(t)\left(1-u^{*'}(t)+J'(t)\right)}{1-F(t)} \\ &= \frac{f'(t)}{f(t)}J'(t) + \frac{f(t)}{1-F(t)}\left(1-u^{*\prime}(t)\right).\end{aligned} \tag{32}$$

Next, let us consider Case (24b) when $u^*(t) = T_{\max}$, that is, $t_0 \le t \le T_{\max}$. We first directly compute $J(t)$:

$$J(t) = c + \frac{\int_t^{T_{\max}} (T_{\max} - \tau) f(\tau) d\tau}{1-F(t)}, \tag{33}$$

so that

$$\frac{dJ(t)}{dt} = \frac{-(T_{\max} - t) f(t)(1-F(t)) - f(t)\int_t^{T_{\max}} (T_{\max}-\tau) f(\tau) d\tau}{(1-F(t))^2}$$

and using (33), this finally reduces to:

$$\frac{dJ(t)}{dt} = \frac{f(t)}{1-F(t)}\left[t - u^*(t) + J(t) - c\right].$$

We can immediately see that this is equivalent to (31).

Now let us consider $du^*(t)/dt$. When $u^*(t) = T_{\max}$, (24b) is satisfied. First, if $\left.\frac{\partial G(t,u)}{\partial u}\right|_{u=T_{\max}} < 0$, we must have:

$$\frac{du^*(t)}{dt} = 0. \tag{34}$$

When both $\left.\frac{\partial G(t,u)}{\partial u}\right|_{u=T_{\max}} = 0$ and $u^*(t) = T_{\max}$, we know that $t = t_0$, in which case we will obtain the same $du^*(t)/dt$ as in (28) by differentiating (26).

To conclude, we have derived the necessary optimality conditions. For $J(t)$, the solution is specified by (31). Provided that $|f'(\cdot)| < L < \infty$, Lemmas 5.1 and 5.2 imply that either $u^*(t) = T_{\max}$ for all $0 \leq t \leq T_{\max}$, or there exists a positive value t_0 such that $u^*(t) = T_{\max}$ for $t \in [t_0, T_{\max}]$ and $u^*(t) < T_{\max}$, $t < t_0$. Thus, $u^*(t)$ is specified either by $du^*(t)/dt = 0$ if $t > t_0$, or by (28) if $0 \leq t \leq t_0$ where t_0 satisfies:

$$u^*(t_0) = T_{\max}, \quad \left.\frac{\partial G(t,u)}{\partial u}\right|_{t=t_0,\ u=T_{\max}} = 0.$$

$du^*(t)/dt$ also involves the second derivative of $J(t)$, which was obtained in (32).

We can use the optimality conditions to solve for $u^*(t)$, by integrating (28), (31), and (32) jointly backwards, with initial conditions provided by Lemma 5.2:

$$u^*(\tau) = T_{\max} \quad \text{for } T_{\max} - \delta \leq \tau \leq T_{\max}.$$

5.3 Special Case: Exponential Interarrival Time Distribution

The fundamental question posed in the Bellman equation (15) is how to optimally control the sleep time conditioned on the fact that $T > t$. Because the exponential distribution is memoryless, knowing $T > t$ does not provide any information. Intuitively, therefore, the optimal control under the exponential distribution will be a constant sleep time, since all wake up events are no different than each other. We can prove this and obtain the optimal fixed sleep time z^* by solving (15).

THEOREM 5.3. *If $F(\tau) = 1 - e^{-\lambda\tau}$, then there exists a constant K such that $J(t) \equiv K$. K satisfies the equation $c + \lambda^{-1}\ln(1+\lambda K) - K = 0$. The optimal sleep time is given by $z(t) \equiv z^* = \lambda^{-1}\ln(1+\lambda K)$.*

PROOF. We just need to verify that $J(t) \equiv K$ and $z^* = \lambda^{-1}\ln(1+\lambda K)$ is the solution to the Bellman equation (15):

$$\begin{aligned} J(t) &= \min_{z>0}\left\{c + \frac{\int_t^{t+z}(t+z-\tau)\lambda e^{-\lambda\tau}d\tau}{e^{-\lambda t}} + J(t+z)\frac{e^{-\lambda(t+z)}}{e^{-\lambda t}}\right\} \\ &= \min_{z>0}\left\{c + \int_0^z (z-\tau)\lambda e^{-\lambda\tau}d\tau + J(t+z)e^{-\lambda z}\right\}. \end{aligned} \tag{35}$$

Letting $J(t) \equiv K$, for the optimal control z^*, we have

$$K = c + \int_0^{z^*}\left(z^* - \tau\right)\lambda e^{-\lambda\tau}d\tau + Ke^{-\lambda z^*}. \tag{36}$$

First, because z^* is the minimizer of the right-hand side, we take the derivative with respect to z^* and set it to zero:

$$\frac{d}{dz^*}\left\{\int_0^{z^*}(z^*-\tau)\lambda e^{-\lambda\tau}d\tau + Ke^{-\lambda z^*}\right\} = 0, \tag{37}$$

that is

$$1 - e^{-\lambda z^*} - K\lambda e^{-\lambda z^*} = 0, \tag{38}$$

whose solution is $z^* = \lambda^{-1}\ln(1+\lambda K)$. Using z^* in (36) we obtain:

$$c + \lambda^{-1}\ln(1+\lambda K) - K = 0, \tag{39}$$

which does have a positive solution. To see this, let $v(x) = c + \lambda^{-1}\ln(1+\lambda x) - x$. Clearly, $v(x)$ is continuous in x, and $v(0) = c$. Because $\ln(1+\lambda x)$ has an order of $o(x)$, when x becomes large enough $v(x)$ is negative. Hence, there must exist a positive value K such that $v(K) = 0$. Therefore, we have verified that $J(t) \equiv K$ and $z^* = \lambda^{-1}\ln(1+\lambda K)$ is the solution to the Bellman equation (16). □

Clearly, the larger c becomes, the larger K has to be, and z^* is larger as well but grows slower than K. The intuitive explanation is that if the cost to poll the channel is higher, on average the energy consumption to receive a message is also higher, and the receiver will wake up at a lower frequency. The exponential distribution represents the worst case in our problem, because knowing $T > t$ produces no value at all in the decision making process. For any other distribution, this information will be helpful and thus the optimal control is dynamic (depending on the state t). Another feature of the exponential distribution is that it has infinite support. For such distributions, we cannot obtain the optimal solution using the necessary optimality conditions because initial conditions given in the last subsection require a finite support. Generally, infinite support cases require solving the Bellman equation (16) analytically. This is in contrast to the fixed expected preamble duration approach in Section 4 where the sleep time is solved forward in time and infinite support cases can be handled.

6. DISTRIBUTION APPROXIMATION AND LEARNING

In Sections 4 and 5 we developed two dynamic sleep time control approaches. In both approaches, to compute the sleep time policy $z(t)$, we need to know the interarrival time distribution $F(\cdot)$. This breaks down to three cases: (1) $F(\cdot)$ is known analytically; (2) $F(\cdot)$ is known but can only be expressed in terms of a histogram or quantiles or just sample data; (3) $F(\cdot)$ is unknown and needs to be learned.

In Case (1), we can substitute $F(\cdot)$ into the equations or ODEs in the previous sections and solve for $z(t)$ using routine ODE solvers. In both Cases (2) and (3), we need to reconstruct $F(\cdot)$ from data. In these cases, $F(\cdot)$ will be approximated, and the dynamic sleep time control algorithms are developed corresponding to the approximation. For Case (3), we assume that the arrival process is independently idntically distributed, that is, the interarrival times $\{T_i, i > 0\}$ form a

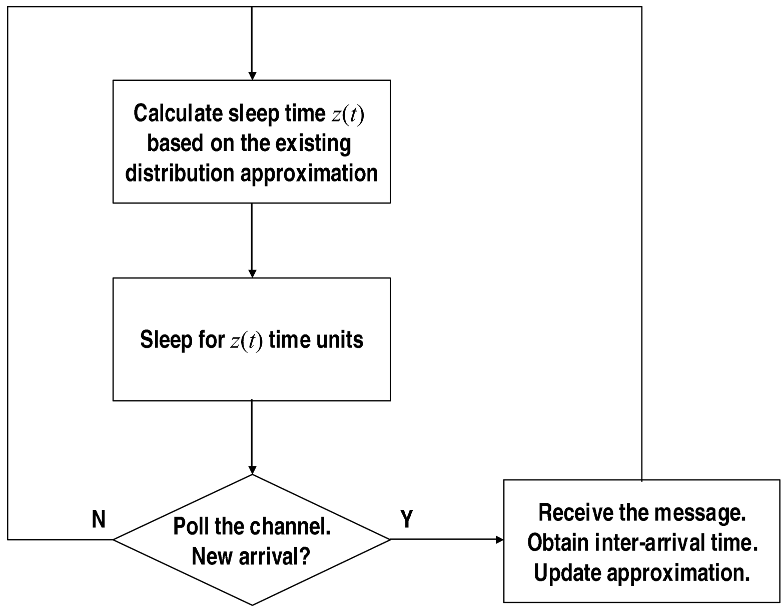

Fig. 3. Flow chart of the adaptive dynamic sleep time control.

sequence of independently idntically distributed random variables with distribution $F(\cdot)$, and provide an online algorithm that learns $F(\cdot)$ as time proceeds. The procedure is as shown in Figure 3. At the beginning of the process, the receiver has a preliminary distribution approximation that is manually input. The receiver computes the sleep time policy using the existing distribution approximation, and follows the policy. When a new message is received, the receiver also obtains a sample of T, which is used to update the current approximation. As the receiver observes more samples, the approximation will approach the underlying interarrival time distribution of the events.

In this section, we first focus on how to approximate and store an arbitrary distribution $F(\cdot)$ in a suitable way for sensor nodes. Then, we illustrate the aforementioned online algorithm, which learns $F(\cdot)$ without storing the sample data.

6.1 Distribution Approximation Using Quantiles

A probability distribution can be approximated in various ways. Generally, for distributions with known finite support, a histogram provides a good approximation. However, in many cases, the upper bound, $T_{\max}$, of the support region of the histogram is not known in advance. Therefore, it is hard to determine the size of each interval in the histogram and the number of intervals. We use an alternative approach for distribution approximation, which occupies a fixed amount of memory while retaining the ability to accommodate an unknown support region as well as the ability to simplify the computation in both dynamic sleep time control approaches.

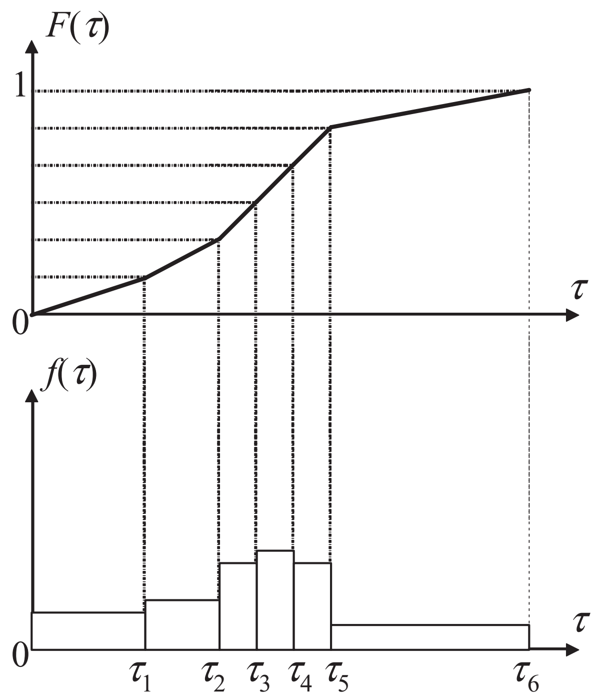

Fig. 4. Piecewise linear approximation of arbitrary distribution.

The aforementioned inherent problem with histograms is due to the fact that it is quantizing the unknown support region of the pdf. On the other hand, consider the cdf $F(\tau)$ of any distribution. $F(\tau)$ can only take values in a fixed interval $[0, 1]$. Thus, we can construct a finite quantization on the y-axis. Figure 4 illustrates the approximation. In this figure, the arbitrary distribution $F(\tau)$ is approximated by $M = 6$ values, from τ_1 to τ_6. The values are selected such that:

$$F(\tau_i) = \frac{i}{M},\ 1 \leq i \leq N. \tag{40}$$

In other words, τ_i is the (i/M)th quantile of $F(\tau)$. If the underlying distribution has an infinite support region, τ_M will have to be infinite. In this case, we choose a large enough value for τ_N, such as the $(1 - 0.1/M)$th quantile.

Points in $[\tau_i, \tau_{i+1})$ are linearly interpolated. Setting $\tau_0 = 0$, f_i is defined as the slope of $F(\tau)$ in the interval $[\tau_{i-1}, \tau_i)$:

$$f_i = \frac{1}{M(\tau_i - \tau_{i-1})},\ i = 1, \ldots, N, \tag{41}$$

and is hence constant. Clearly, the underlying distribution is approximated by a finite mixture of uniform distributions. As we will soon see, this allows us to derive a simple algorithm to solve Equation (1) in the fixed expected preamble duration problem. It also simplifies some computation that arises in the total energy minimization problem.

6.2 Updating the Existing Approximation from New Data

In what follows, we propose a mechanism to update the existing approximation based on new arrival information. The way in which the approximation is updated is essentially a quantile estimation problem. Our main concern, however, is to save memory as opposed to traditional quantile estimation methods that use sample quantiles. For example, Tierney [1983] and Chen et al. [2000] have proposed space-efficient recursive algorithms on quantile estimation based on stochastic approximation (SA) algorithms. Here we propose an approach also based on the stochastic approximation method in Tierney [1983], but modified to take advantage of the setting in our problem.

In sequential quantile estimation we estimate the αth quantile by solving $F(\tau) = \alpha$, where $F(\tau)$ is the unknown cdf. In Robbins and Monro [1951], an SA algorithm is used to solve the aforementioned equation. In the stochastic approximation algorithm, we start with a guessed value of τ. As we observe a new sample, we make small adjustments to τ, and so on. In Tierney [1983], an estimator of $\tau = F^{-1}(\alpha)$ is given by:

$$\hat{\tau}^{k+1} = \hat{\tau}^k - \frac{d^k}{k+1}(\mathbf{1}\{T^{k+1} \leq \hat{\tau}^k\} - \alpha) \tag{42}$$

$$d^k = \min\left\{ \left(\hat{\phi}^k\right)^{-1}, d^0 k^a \right\},\ 0 < a < \frac{1}{2}, \tag{43}$$

where $\hat{\phi}^k$ is an estimate of $F'(\tau)$, given by:

$$\hat{\phi}^{k+1} = \frac{1}{k+1}\left(k\hat{\phi}^k + \frac{\mathbf{1}\left\{\left|T^{k+1} - \hat{\tau}^k\right| \leq h_{k+1}\right\}}{2h_{k+1}}\right), \tag{44}$$

where $\{h_k\}$ is a decreasing sequence satisfying $\sum_{k=1}^{\infty}(k^2 h_k)^{-1} < \infty$. It has been proved in Tierney [1983] that estimator (42) converges to the αth quantile. In (44), $F'(\tau)$ is estimated by averaging the number of samples falling in a shrinking interval $[\hat{\tau}^k - h_{k+1}, \hat{\tau}^k + h_{k+1}]$. The issue with this estimation is that when samples are few, the convergence speed is slow since only local information is used. Also, one needs to carefully adjust $\{h_k\}$ so that it does not decrease too fast so that we can collect enough samples falling in the interval. Since in the problem presented in this article we need to estimate a set of quantiles and use them to approximate the whole function F, we can take advantage of the estimation of other quantiles in estimating $F'(\tau)$ around some particular quantile. For all $1 \leq i \leq M-1$,

$$\hat{\tau}_i^{k+1} = \hat{\tau}_i^k - \frac{d_i^k}{k+1}\left(\mathbf{1}\{T^{k+1} \leq \hat{\tau}_i^k\} - \frac{i}{M}\right) \tag{45}$$

$$d_i^k = \min\left\{ \left(\hat{\phi}_i^k\right)^{-1}, d^0 k^a \right\}, 0 < a < \frac{1}{2} \tag{46}$$

$$\hat{\phi}_i^{k+1} = \frac{2i}{M\left(\hat{\tau}_{i+1}^k - \hat{\tau}_{i-1}^k\right)} \tag{47}$$

and for $i = M$:

$$\hat{\tau}_M^{k+1} = \max\left\{\hat{\tau}_M^k, T^{k+1}\right\}. \tag{48}$$

We need to point out that (47) is not an asymptotically unbiased estimator of $f(\tau_i)$, unlike (44), but rather a finite difference approximation to leverage more information. However, our numerical results show that the modified method presented here has an advantage in terms of convergence speed. After we have obtained new $\hat{\tau}_i^{k+1}$, $i = 1, \ldots, M$, we can recalculate the sleep time control policy with new quantile estimates.

To conclude, for the sensor node, it only needs to keep $\{\hat{\tau}_i\}_{i=1}^M$ in memory. The initial values are given by the a priori knowledge of the application. When a new message arrives, the interarrival time is obtained through a timer. Then, $\{\hat{\tau}_i\}_{i=1}^M$ can be updated using (45)–(47). The novelty of the distribution approximation and learning approach is that: (1) it uses quantiles to approximate a distribution, hence the memory usage is limited; (2) the learning process is sequential and does not require storage of past observations. Both features are extremely suitable for sensor networks, and applicable to any setting that involves distribution learning. Moreover, the structure of the approximation (mixture of uniforms) allows some simplification in the computation of dynamic sleep times, as we will see in the next section.

7. DYNAMIC SLEEP TIME CONTROL COMPUTATION ALGORITHMS

In this section we present two computation algorithms for dynamic sleep time control, based on the distribution approximation scheme described in Section 6. The algorithms correspond to both Case (2), where $F(\cdot)$ is approximated and Case (3), where $F(\cdot)$ is learned on line.

7.1 Fixed Expected Preamble Duration

In the fixed expected preamble duration approach, we need to solve the equation:

$$\bar{D} = t + z - \frac{\int_t^{t+z} \tau dF(\tau)}{F(t+z) - F(t)} \tag{49}$$

for $z(t)$. While we have characterized the solution using ODE (5) in Section 4.2, based on the distribution approximation given in Section 6.1, we are able to solve Equation (49) in a much simpler way. First, we can calculate the integral in (1) by segments. For notational simplicity, let $u = t + z$. To calculate the integral, we first need to determine in which interval $[\tau_{i-1}, \tau_i)$, $i = 1, \ldots, M$, t and u lie. Define integers m, n as in Figure 5 such that:

$$\begin{gathered} 1 \le m, \quad n \le M \\ \tau_{m-1} < t \le \tau_m \\ \tau_{n-1} < u \le \tau_n. \end{gathered}$$

Note that there may not exist m, n, satisfying these inequalities. For example, it is possible that $t > \tau_M$, or $u > \tau_M$. We will deal with these special cases separately.

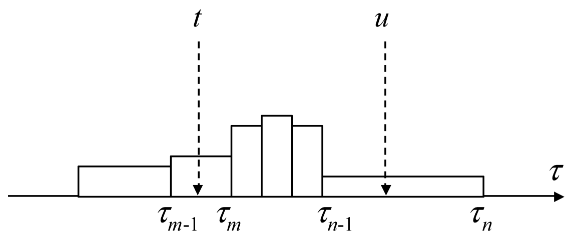

Fig. 5. Definition of m and n.

Define:

$$A_m(t) = \int_t^{\tau_m} \tau dF(\tau) = \frac{f_m}{2}\left(\tau_m^2 - t^2\right)$$
$$B_i = \int_{\tau_{i-1}}^{\tau_i} \tau dF(\tau) = \frac{f_i}{2}\left(\tau_i^2 - \tau_{i-1}^2\right)$$

for $i = 1, \ldots, M-1$, and

$$C_n(u) = \int_{\tau_{n-1}}^{u} \tau dF(\tau) = \frac{f_n}{2}\left(u^2 - \tau_{n-1}^2\right),$$

where f_i are defined in (41). Then, it is clear that the integral:

$$I(t, u) \triangleq \int_t^u \tau dF(\tau)$$

can be rewritten in terms of $A_m(t)$, B_i, and $C_n(t, z)$:

$$I(t, u) = A_m(t) + \sum_{i=m+1}^{n-1} B_i + C_n(u).$$

We replace z and the integral in (49) by u and $I(t, u)$, obtaining:

$$\bar{D} = u - \frac{I(t, u)}{F(u) - F(t)}, \tag{50}$$

where $F(u)$ and $F(t)$ can also be calculated based on the linear interpolation:

$$F(t) = F(\tau_m) - f_m(\tau_m - t) = \frac{m}{M} - f_m(\tau_m - t)$$
$$F(u) = \frac{n}{M} - f_n(\tau_n - u).$$

Noting that $F(u) - F(t)$ is non-zero, (50) becomes a solvable quadratic equation in u. Before solving the equation, we need to identify n, which designates in which interval the solution u lies. Since $E[D|t < T < u]$ is nondecreasing in u, n must satisfy:

$$\tau_{n-1} - \frac{I(t, \tau_{n-1})}{F(\tau_{n-1}) - F(t)} < \bar{D} < \tau_n - \frac{I(t, \tau_n)}{F(\tau_n) - F(t)}. \tag{51}$$

Therefore, n can be obtained by performing a search over all possible $m \leq n < M$. Once n is obtained, we are in position to solve for u. Define:

$$a(t) = I(t, u) - \frac{1}{2} f_n u^2 = A_m(t) + \sum_{i=m+1}^{n-1} B_i - \frac{f_n}{2} \tau_{n-1}^2 \tag{52}$$

$$b(t) = F(u) - F(t) - f_n u = \frac{n}{M} - f_n \tau_n - \frac{m}{M} + f_m(\tau_m - t), \tag{53}$$

where $a(t), b(t)$ are calculable values depending on t. Thus, (50) can be rewritten as:

$$E[D|t < T < u] = u - \frac{\frac{1}{2} f_n u^2 + a(t)}{f_n u + b(t)}.$$

As previously stated, this is a solvable quadratic equation in u. Its solution (omitting the argument t associated with $a(t)$ and $b(t)$) is:

$$u = \frac{-\left(b - f_n \bar{D}\right) + \sqrt{\left(b - f_n \bar{D}\right)^2 + 2 f_n \left(a + b\bar{D}\right)}}{f_n} \tag{54}$$

and thus

$$z(t) = \frac{-\left(b - f_n \bar{D}\right) + \sqrt{\left(b - f_n \bar{D}\right)^2 + 2 f_n \left(a + b\bar{D}\right)}}{f_n} - t.$$

There are three special cases:

Case 1. $n = m$, that is, u and t both lie in the same interval $[\tau_{m-1}, \tau_m]$. Then,

$$\bar{D} = u - \frac{\frac{1}{2} f_m \left(u^2 - t^2\right)}{f_m(u - t)} = \frac{1}{2}(u - t) = \frac{1}{2} z,$$

which gives:

$$z(t) = 2\bar{D} \tag{55}$$

Case 2. $u > \tau_M$, which is equivalent to:

$$\bar{D} > \tau_M - \frac{I(t, \tau_M)}{1 - F(t)}. \tag{56}$$

This case implies that even letting $u = \tau_M$, which is the maximum possible interarrival time in the approximation, cannot satisfy the constraint set by $\bar{D}$. Since $I(t, u) = I(t, \tau_N)$ and $F(u) = 1$ for all $u \geq \tau_M$, the solution to (50) is given by:

$$z(t) = \bar{D} + \frac{I(t, \tau_M)}{1 - F(t)} - t. \tag{57}$$

In practice we will have to make a decision. If τ_M is the real boundary of the support region, there is no benefit to wake up later than τ_M because the incoming event will definitely occur before τ_M. In this case the sleep time should simply be $z = \tau_M - t$ instead. However, if τ_M is an estimated value, we cannot be sure that the incoming event will definitely occur before τ_M. Therefore, we do not have enough information to compute the sleep time, and (57) provides a conservative sleep time such that the expected preamble duration will not exceed $\bar{D}$.

Case 3. $t > \tau_M$. This case implies that the current interarrival time has exceeded the maximal possible interarrival time in the current approximation. This case generally follows Case 2, which means the statistical knowledge about the interarrival time is insufficient. In this case, we take a conservative approach as well:

$$z(t) = \bar{D}. \tag{58}$$

In conclusion, the approximation of an arbitrary distribution by a mixture of uniform distributions has introduced an attractive structure, which has greatly eased the computational difficulty of solving (1).

7.2 Total Energy Minimization

Now we switch to the total energy minimization problem. Assume we have an approximation of $F(\cdot)$ specified by $\{\tau_0, \ldots, \tau_M\}$, where $\tau_0 = 0$ and τ_i is the i/Mth quantile. Define the state space as $\{0, 1, \ldots, M-1\}$, where state i corresponds to $t \in [\tau_i, \tau_{i+1})$. Therefore, we have quantized the finite continuous state space $[0, T_{\max}]$ into M discrete states. With this discretization we can solve the Bellman equation (15) exhaustively without suffering from the curse of dimensionality, and the exhaustive algorithm's complexity only depends on the discretization resolution M. The allowable wake up time epochs are $\{\tau_{i+1}, \ldots, \tau_M\}$. Hence, the decision space at state i will be $\{i+1, \ldots, M\}$, corresponding to wake-up time at τ_j. Define the value function:

$$V_{i,u} = c + \frac{\tau_u(u-i) - M\int_{\tau_i}^{\tau_u} xf(x)\,dx}{M-i} + J_u\frac{M-u}{M-i},$$

where J_i is the minimum expected cost-to-go at state i, and u_i^* is the optimal wake up time. Therefore, the discrete Bellman equation (16) is:

$$\begin{aligned} J_i &= \min_{u\in\{(i+1),\ldots,M\}} V_{i,u} \\ &= \min_{u\in\{(i+1),\ldots,M\}} \left\{ c + \frac{\tau_u(u-i) - M\int_{\tau_i}^{\tau_u} xf(x)\,dx}{M-i} + J_u\frac{M-u}{M-i} \right\} \\ u_i^* &= \arg\min_{u\in\{(i+1),\ldots,M\}} V_{i,u}. \end{aligned} \tag{59}$$

Using the fact that under a quantile-based approximation, a distribution is approximated by a mixture of uniforms, we can easily evaluate the integral with (41):

$$\begin{aligned} \int_{\tau_i}^{\tau_u} xf(x)\,dx &= \sum_{j=i}^{u-1}\int_{\tau_j}^{\tau_{j+1}} x\frac{1}{M(\tau_{j+1}-\tau_j)}dx \\ &= \sum_{j=i}^{u-1}\frac{1}{2M}\left(\tau_j + \tau_{j+1}\right). \end{aligned}$$

Table I. The DP Algorithm

For	$i = M - 1$ down to 0 :
	Compute $V_{i,M}$ from (61).
	For $j = M - 1$ down to $i + 1$:
	Compute $V_{i,j}$ recursively from (62)
	$J_i = \min_{i+1 \le j \le M} V_{i,j}$
	$u_i^* = \arg\min_{i+1 \le j \le M} V_{i,j}$
	$z_i^* = \tau_{u_i^*} - \tau_i$

Then, $V_{i,u}$ can be expressed in terms of τ_i values:

$$V_{i,u} = c + \frac{\tau_u (u - i) - \frac{1}{2} \sum_{j=i}^{u-1} \left(\tau_j + \tau_{j+1}\right)}{M - i} + J_u \frac{M - u}{M - i}. \tag{60}$$

We can see that, expressed in quantiles, the Bellman equations are simpler to evaluate. Now we can solve (59) numerically. The direct, exhaustive algorithm to solve (59) has complexity $O(M^3)$, because one has to compute $V_{i,u}$ for all possible i and u, and to compute $V_{i,u}$ for a single (i, u) pair is an $O(M)$ task due to the inner sum, as seen in (60). However, the computation can be simplified noting that $V_{i,u}$ can be easily obtained from $V_{i,u+1}$ without computing the integral again. To see this, take the difference between $V_{i,u}$ and $V_{i,u+1}$. Thus,

$$V_{i,M} = c + \frac{\tau_M (M - i) - \frac{1}{2} \sum_{j=i}^{M-1} \left(\tau_j + \tau_{j+1}\right)}{M - i} \tag{61}$$

$$V_{i,u} = V_{i,u+1} - \frac{(\tau_{u+1} - \tau_u)(u - i) + \frac{1}{2}(\tau_{u+1} - \tau_u) - J_{u+1} + (J_{u+1} - J_u)(M - u)}{M - i}, \tag{62}$$

so we can obtain $V_{i,u}$ from $V_{i,u+1}$ in $O(1)$ time.

To solve (59), we evaluate $V_{i,u}$ backwards in time. From $V_{M-1,M}$, obtained with (61), we obtain J_{M-1} and $u_{M-1}^* = M$. We compute $V_{M-2,M}$, and then $V_{M-2,M-1}$ with (62), obtaining J_{M-2} and u_{M-2}^*, and so on. Therefore, we can solve (59) in $O(M^2)$ complexity. The process is summarized in the dynamic programming (DP) algorithm in Table I.

Recall that in Section 5.2 we have derived a set of ODEs (28), (31), and (32) that characterizes the optimal solution $u^*(t)$. We can obtain $u^*(t)$ by integrating the differential equations backwards using a routine ODE solver. Complexity-wise, this is generally an $O(M)$ task, which is very efficient. However, the drawback is that because the group of ODEs is only a necessary condition for the optimality of $u^*(t)$, it may not be the global optimal solution. Numerical stability is another concern. Moreover, the ODEs require differentiability of $f(\cdot)$, which cannot be satisfied if $f(\cdot)$ is not analytically known. Therefore, it is still desirable to use the DP algorithm to compute a globally optimal solution at the expense of more computation effort.

8. NUMERICAL RESULTS

In this section we present some numerical examples to illustrate our dynamic sleep time control approaches and quantify their benefit compared to fixed

Table II. Parameters of the Distribution Examples. All Distributions are Truncated at $T_{\max} = 60$ and Normalized

Distribution	Parameters
Uniform	$[0, 60]$
Weibull	$\alpha = 20, \beta = 2$
Bi-modal Gaussian	$\mu_1 = 15, \mu_2 = 48, \sigma_1 = \sigma_2 = 3$, modal probability = 0.5

sleep time control. In Section 8.1, we use some distributions that are known to the receiver. We will see that our fixed expected preamble duration approach dominates the conventional fixed sleep time control, and that our total energy minimization approach achieves better energy savings comparing to the best possible fixed sleep time control. In Section 8.2 we show an example where the distribution is unknown, and how the dynamic sleep time controller learns the distribution and improves performance as time proceeds. In Section 8.3 we extend the total energy minimization approach to a multihop network and evaluate the performance. In Section 8.4 we implement the total energy minimization approach in a commercially available sensor node and evaluate the computation time.

8.1 Known Distribution Examples

We begin with examples where distribution information is known. The distribution examples are: (1) uniform distribution, (2) Weibull distribution, (3) bimodal Gaussian distribution. The parameters of the distributions used are listed in Table II. We set $M = 1000$, which is the number of quantiles in the distribution approximation.

We first compare the fixed expected preamble duration approach to fixed sleep time control. Recall that the fixed expected preamble duration approach is used when one wants to impose stringent requirements on the average preamble duration. Therefore, we compare this approach with fixed sleep time control under the criterion that both approaches yield the same average preamble duration per message. The comparison results are illustrated in Figures 6, 7, and 8, where the Pareto frontiers of both the fixed expected preamble duration approach and the fixed sleep time control are plotted. In fixed sleep time control, by varying the sleep time, we can plot the average preamble duration per message versus the average number of samplings per message. In the fixed expected preamble duration approach we vary $\bar{D}$ and plot the same trade-off relationship.

In Figures 7 and 8, we see that the Pareto frontier of the dynamic sleep time control dominates the fixed sleep time control. In other words, compared to fixed sleep time control, with the same average preamble duration, dynamic sleep time control needs fewer channel samplings. On the other hand, with the same average channel samplings per message, the preamble duration in dynamic sleep time control is shorter. In Figure 6, however, we see that the Pareto frontier of both fixed sleep time control and the fixed expected preamble duration approach overlap. This is to be expected because in (5), if we let $f(t) = 1/T_{\max}$, $F(t) = t/T_{\max}$, we will immediately get $u(t) = 1$, which means $z(t) = 0$, a fixed sleep time until $u(t) = T_{\max}$.

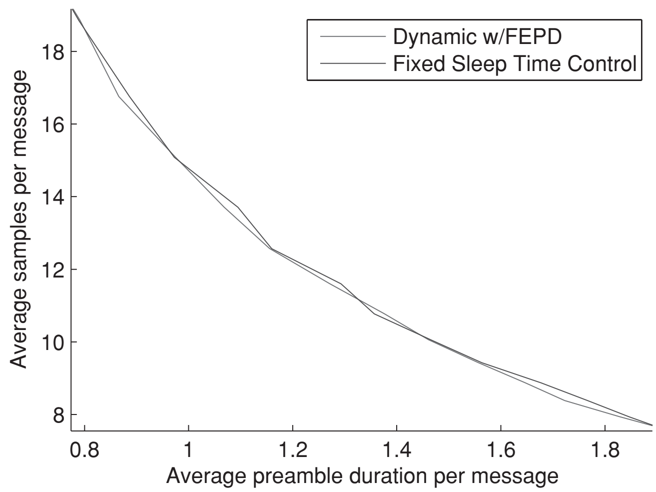

Fig. 6. Pareto frontier plot under the uniform distribution.

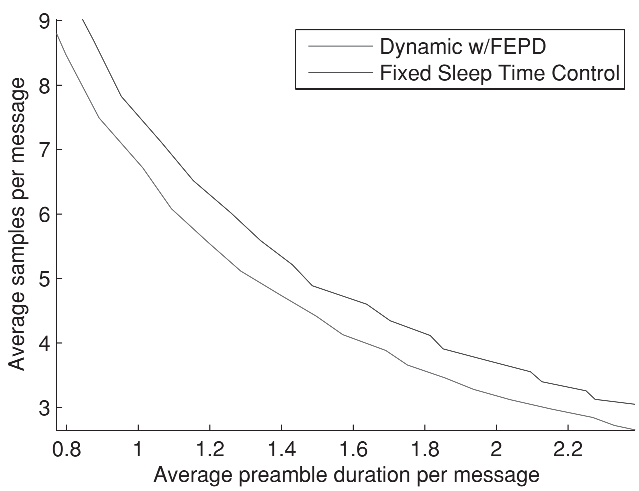

Fig. 7. Pareto frontier plot under the Weibull distribution.

Next, we compare the best fixed sleep time control, and our total energy minimization (TEM) approach. The best fixed sleep time control is obtained by using Monte Carlo simulation and exhaustive search to find the best sleeping interval that minimizes energy usage. We set $c = 0.1$ in this example. The data are obtained by simulating a 10,000 message sample path. Table III shows the energy cost comparison in different distribution examples with different sleep time control approaches.

Table III shows that depending on different distributions, our total energy minimization approach achieves a considerable amount of energy savings. To further analyze how these savings are achieved, we plot the optimal sleep time

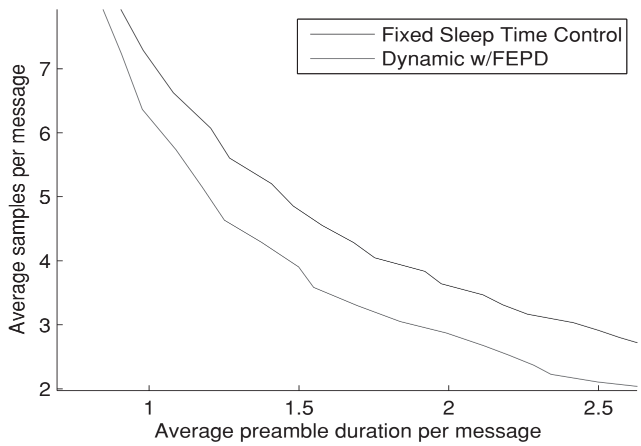

Fig. 8. Pareto frontier plot under the bimodal gaussian distribution.

Table III. Energy Cost Comparison in Different Distributions. The Values are the Average Energy Cost per Message

Distribution	Best Fixed	TEM	Save%
Uniform	2.4272	2.2976	5.34%
Weibull	2.0593	1.8947	7.99%
Bi-modal Gaussian	2.5050	1.5984	36.19%

$z^*(t)$ and the optimal expected cost-to-go function $J(t)$ as a function of the age t, shown in Figures 9, 10, and 12. The Direct curve is obtained by solving the Bellman equation with known $F(\cdot)$ and $f(\cdot)$. The ODE curve is obtained by solving the necessary optimality conditions. The DP curve is obtained by approximating $F(\cdot)$ using quantiles, and applying the DP algorithm presented in Section 7.2.

First, in Figure 9, we see that as t increases, the sleep time $z^*(t)$ decreases gradually, and approaches 0 at $t \to T_{\max}$. Recall that the sleep time at t is calculated based on the fact that no event has occurred during the last t amount of time. Therefore, as t increases, the possibility that the event takes place in the near future increases as well, which leads to a shorter sleep time. We also see that while DP overlaps with the Direct curve, the ODE curve has a small deviation. This is due to the numerical instability of the ODE approach.

The Weibull distribution models random events with an increasing hazard rate with respect to the age of the event. We see that in Figure 10, the optimal sleep time control $z^*(t)$ is again decreasing as t increases, but with a different shape from Figure 9. In this case, the ODE curve overlaps with the direct curve, while the DP curve deviates in $t \in [40, 60]$. This is because near the thin tail of the distribution, the difference $\tau_{i+1} - \tau_i$ becomes larger. Because in the DP algorithm, the minimum sleep time at state i is $\tau_{i+1} - \tau_i$, the controller does not have any other choice but to sleep for this whole large interval. Thus, the control granularity is inevitably larger, resulting in a deteriorating performance.

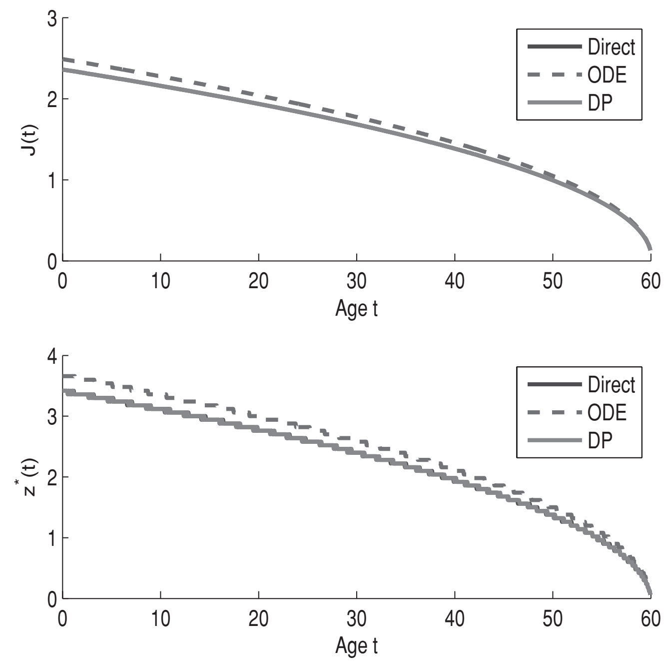

Fig. 9. The optimal sleep time control obtained under Uniform distribution.

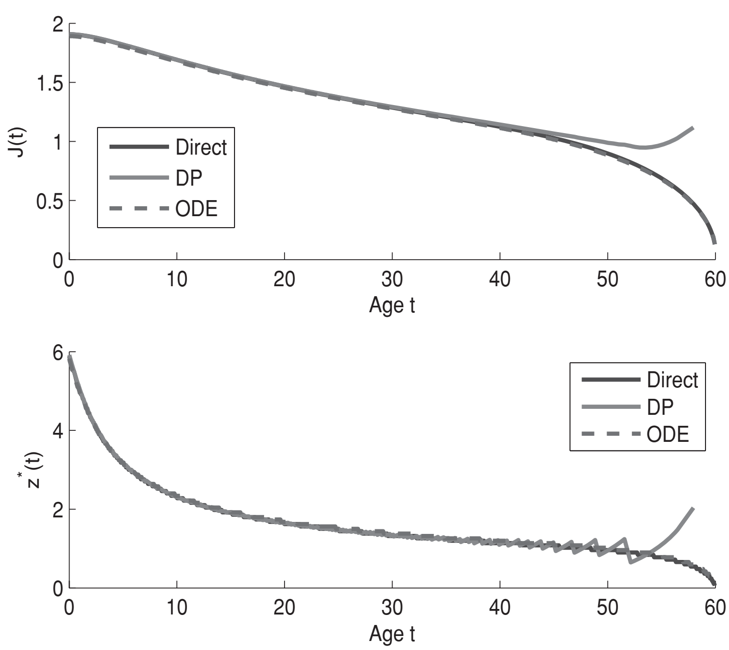

Fig. 10. The optimal sleep time control obtained under Weibull distribution.

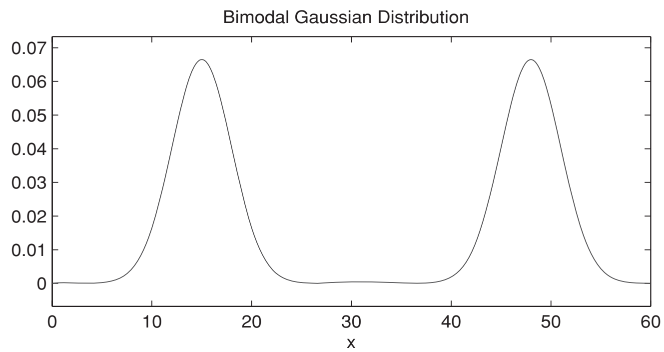

Fig. 11. The bimodal Gaussian distribution.

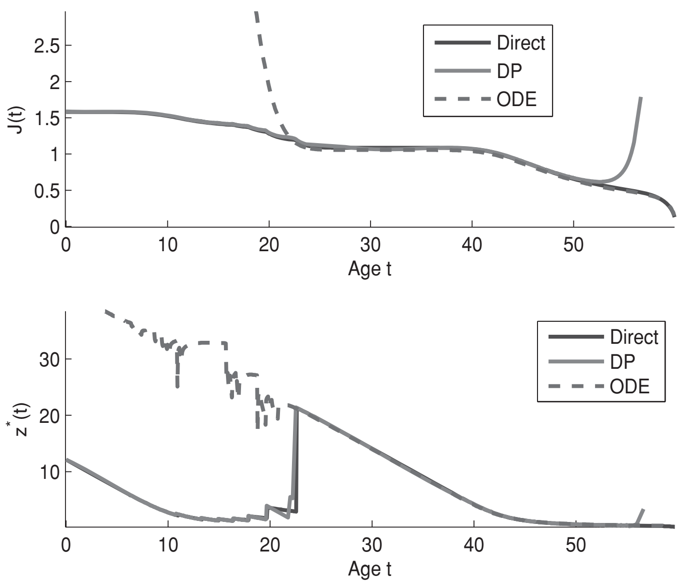

Fig. 12. The optimal sleep time control obtained under bimodal Gaussian distribution.

The bimodal Gaussian (BG) distribution is shown in Figure 11. This pdf has two peaks corresponding to two modes of the distribution. The interval of approximately $[20, 40]$ shows a valley where the event arrival probability is small. The optimal policy obtained is shown in Figure 12. We first notice that there is a discontinuity in the optimal sleep time curve of Direct and DP. From $t = 0$ to around $t = 15$, we see the sleep time is decreasing. However, because the probability of an event occurring in the interval of $[20, 40]$ is small, the optimal sleep time is to skip over the period in order to save energy. We also notice that ODE completely fails to handle the jump, which is expected. The explanation of the occurrence of such a jump is that there exist multiple local minima in $G(t, u)$, so (24a) and (24b) no longer provide a globally optimal solution.

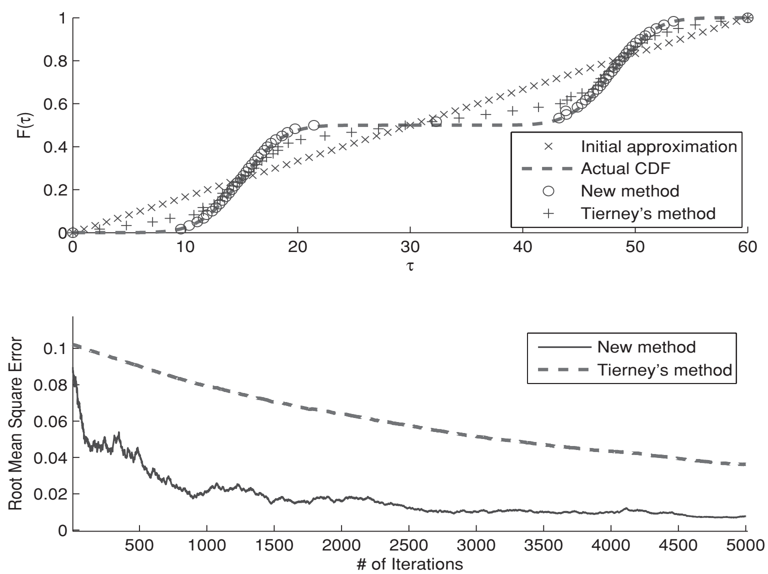

Fig. 13. Comparison of the convergence speed between the new method and Tierney's method.

8.2 Distribution Learning

When the distribution is unknown to the dynamic sleep time controller, we use the algorithm presented in Section 6.2 to learn it by observing samples sequentially. Figure 13 shows simulation results for a system with the bimodal Gaussian distribution. The process begins with an approximation using a uniform distribution $U[0, 60]$, labeled as Initial approximation in Figure 13. The actual cdf of the bimodal Gaussian distribution is labeled as Actual CDF. We compare the result of our learning algorithm, labeled as New method and the result using Tierney's method [Tierney 1983], labeled as Tierney's method. The two methods are computed separately with the same sample sequence. We can see that compared to Tierney's method, the method we have proposed has a significant advantage in convergence speed, that is, the accuracy of the learning. This is a very important feature since the quality of control increases faster as the root-mean-square error (RMSE) decreases. As the number of iterations increases, we can see that the RMSE of both methods tends to zero.

In Figure 14, the total power is measured by taking a moving average of the instantaneous power. The process begins with $\{\tau_i\}_0^M$, initialized using a uniform distribution $U[0, 60]$. Since the approximation error is large, the total power during this period is high. As time evolves and the interarrival times are collected, the approximation is updated and better controls are generated, which subsequently reduces the total power.

In the total energy minimization approach, because of the mismatch in distribution the sleep time controller cannot attain the maximum energy savings at first. Figure 15 shows the average cost per message as time proceeds. We see a decreasing cost as the receiver learns the distribution, adjusts its control policy, and approaches the level obtained by off-line computation.

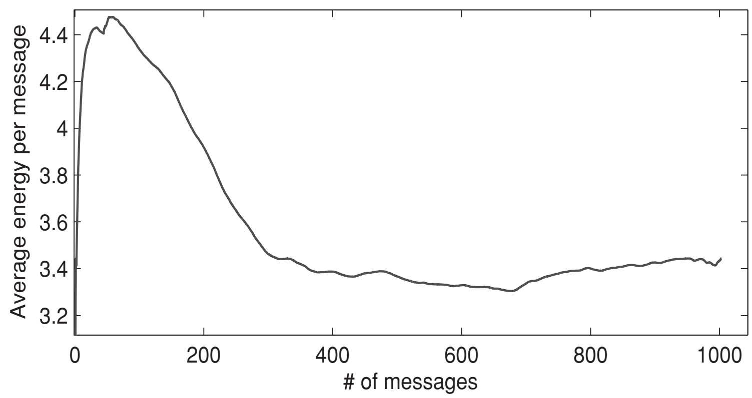

Fig. 14. Average power in the fixed expected preamble duration approach with unknown distribution.

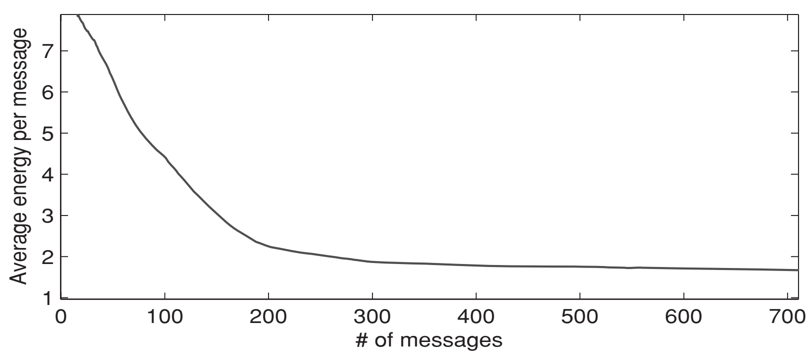

Fig. 15. Average power in the total energy minimization approach with unknown distribution.

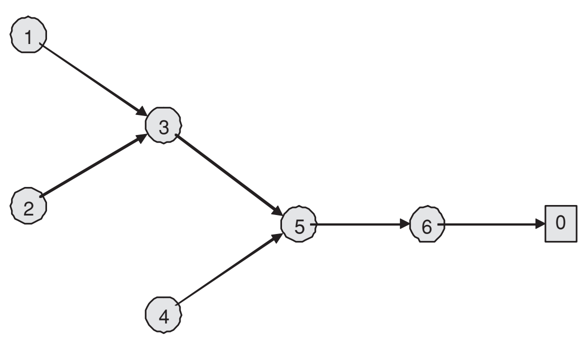

Fig. 16. Network topology.

8.3 Multi-Hop Total Energy Minimization

So far, our approaches focus on single-hop performance. It is interesting to see how they perform in a multihop environment. We consider a network with six nodes, as shown in Figure 16, where node 0 is the base station. Nodes 1–6 are

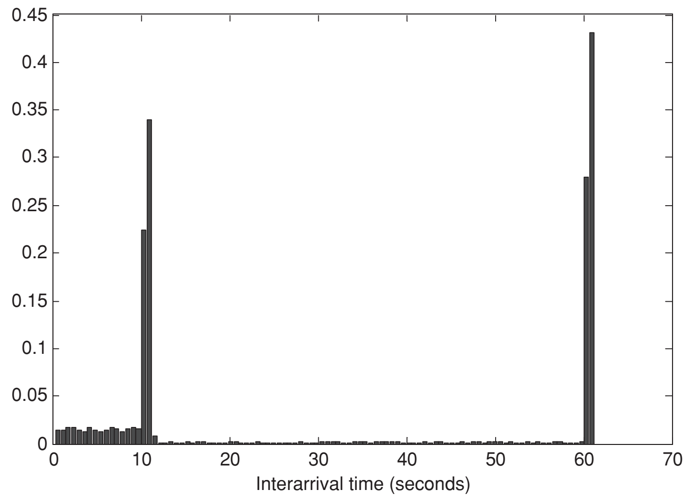

Fig. 17. Histogram of the aggregated interarrival time distribution of the mixed arrival streams.

homogeneous. The dynamic sleep time controller will be applied at nodes 3, 5, and 6, which receive messages from upstream nodes.

To make the experiment more realistic, we consider the following scenario. We assume that the WSN operates in two modes: stand-by mode and active mode. This emulates a typical event detection sensor network, where during stand-by mode, only occasional status report messages are transmitted. When the network detects the event of interest, it switches to active mode, where frequent data packets are transmitted. Therefore, two classes of messages are involved. We assume that the receiver does not know this mode-switching scheme. We also assume the interarrival time of the status report messages is uniformly distributed $U[60, 61]$, which models a constant frequency status report message with 1-second jitter. The active message is triggered by an external event, which occurs according to an exponential distribution with mean $\mu = 300$ (seconds). Once triggered, the active message process continues with interarrival time distribution $U[10, 11]$ for 10 messages. The aggregated interarrival time pdf is shown in Figure 17. We can see two peaks corresponding to two operating modes. When a node receives messages from more than one sender (e.g., nodes 3, 5), the interarrival time is for the aggregated arrival process, that is the superposition of the output processes at nodes 1 and 2 (for node 3) and nodes 3 and 4 (for node 5), respectively. Because the arrival process is complicated, there is no analytical form of $F(\cdot)$ available. Therefore, $F(\cdot)$ needs to be approximated and learned online.

Figure 18 shows the comparison results. On the left is the percentage savings observed on nodes 3, 5, 6 and the total energy savings. On the right is the comparison of total energy amounts. We see that in the multihop environment, although the dynamic sleep time control is still superior in terms of energy savings, the margin is not as large. This is because the closer to the base station, the more messages a node will observe. Because they are all

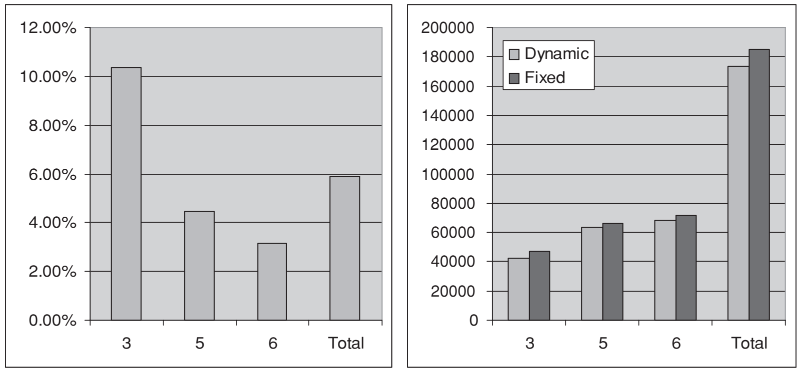

Fig. 18. Performance comparison with fixed sleep time control on various nodes.

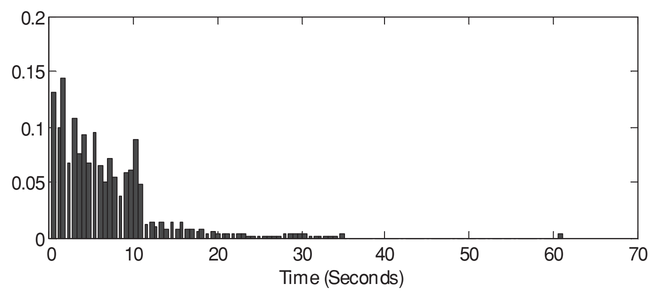

Fig. 19. Interarrival time distribution observed at Node 6.

aggregated into a single arrival stream, the interarrival time distribution resembles an exponential distribution, the worst case in our approach. Figure 19 shows the histogram of the interarrival time at Node 6 which highly resembles an exponential distribution. We can see that the gain from directly extending the single-link energy minimization is limited because it does not consider the topological information.

8.4 Computation Time Experiment with Total Energy Minimization

To demonstrate the feasibility of our total energy minimization approach, we implemented the optimal sleep time computation on a TMote wireless sensor, manufactured by Sentilla, Inc. The TMote is equipped with an 8 MHz MSP430 16-bit microcontroller, manufactured by Texas Instruments, Inc. We measured the CPU time used to compute the optimal sleep time policy using the ODE and DP and show our results (in seconds) as a function of the time resolution M in Table IV. The ODE row describes the time used to compute the optimal sleep time by solving the ODEs (28), (31), and (32) using the basic Euler method. The DP row describes the computation time using the $O(M^2)$ DP algorithm described in Table I.

Due to the limitation of on-board memory (10KBytes), we were unable to experiment with larger M. However, the final case of $M = 300$ is more than enough; if the interevent time does not exceed 60 seconds (typical in WSNs

Table IV. Computation Time Experiment. CPU Times are Measured in Seconds

M	50	100	200	300
ODE	1.5	3	6	9
DP	3	11	45	97

because of heartbeat/health/routing messages), this produces a resolution of 0.2 seconds. We believe that a value in the vicinity of $M = 100$ offers a good trade-off between performance and resource usage. It is important to emphasize that this computation is performed only once to generate the optimal sleep time policy, unless the interevent time distribution is significantly changed. Therefore, the amortized computation time overhead imposed on each message is very small (e.g., <0.1 seconds) compared to the typical interevent time.

We also notice that ODE is much faster than DP thanks to its linear time growth. However, as mentioned before, ODE has numerical stability problems and cannot reliably handle cases like bimodal Gaussian distributions.

9. CONCLUSIONS AND FUTURE WORK

We have presented two dynamic sleep time control approaches to reduce control packet overhead incurred in LPL. In the first approach, fixed expected preamble duration, we determine a dynamic sleep time such that a design requirement on the expected preamble duration is satisfied. The computation of the sleep time uses the interarrival time distribution, and results in fewer channel samplings per message compared to a fixed sleep time control that produces the same average preamble duration.

In the second approach, total energy minimization, we address a different problem, which aims to minimize the energy consumption incurred by the total control packet overhead. We formulated, and managed to solve, the continuous time Bellman equation and derived a set of differential equations characterizing the optimal solution.

In both approaches, we obtained the dynamic sleep time as a function of the event age, with the function posed as the solution to an ODE with initial condition fully specified. The knowledge of the interarrival time distribution is crucial to the computation. If the form of $F(t)$ is known in advance, depending on what that function is, one can often obtain the solution in closed form, otherwise it can be obtained through numerical means. Solving ODE numerically is well studied and easily implemented. However, in practice $F(t)$ is usually not known in advance. Therefore, we have developed a quantile-based distribution approximation and learning algorithm, which exploits the structure of the problem and is an improvement over existing literature. We have created algorithms of both approaches that capitalize on this quantile-based approximation. In both approaches, this approximation simplifies the computation of the dynamic sleep time. In fixed expected preamble duration, the computation of sleep time is even reduced to a closed-form formula. In the total energy minimization approach, the computation of sleep time requires a DP algorithm that ends in finite time.

For future work, there are two extensions. The first is batching. Our approaches only focus on the next message. However, it is possible that the sender can intentionally delay sending the preamble so that a few messages can be batched together and share the same preamble, see Ning and Cassandras [2007]. On the receiver's side, it can also intentionally wake up later, hoping that there will be more messages queued up at the sender's side. In order to investigate this problem, some augmentations to the existing model are needed, allowing the receiver to see beyond the next message.

The second extension is network-level sleep time control. In our numerical example section, we have naively extended our approach to a multihop network. Although it does have some advantages over the best fixed sleep time control, the margin is smaller than in a single hop situation. Clearly, the total energy minimization approach developed for a single link does not minimize the end-to-end control packet overhead in a multi-hop network. Our observation is that the interarrival time distribution gradually conforms to an exponential distribution, a known asymptotic result, which happens to be our worst case. Therefore, at a network level, in addition to what is proposed in this article, some coordination effort is required, which should utilize the topology and possibly more application-related information to achieve better energy savings.

REFERENCES

AVVENUTI, M., CORSINI, P., MASCI, P., AND VECCHIO, A. 2006. Increasing the efficiency of preamble sampling protocols for wireless sensor networks. In *Proceedings of the 1st Mobile Computing and Wireless Communication International Conference (MCWC'06)*. 117–122.

BUETTNER, M., YEE, G. V., ANDERSON, E., AND HAN, R. 2006. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys'06)*. 307–320.

CHEN, F., LAMBERT, D., AND PINHEIRO, J. C. 2000. Incremental quantile estimation for massive tracking. *Knowl. Discov. Data Mining*. 516–522.

EL-HOIYDI, A. AND DECOTIGNIE, J.-D. 2004. WiseMAC: An ultra low power MAC protocol for multihop wireless sensor networks. Lecture Notes in Computer Science, vol. 3121, 18–31. Springer-Verlag.

HILL, J. L. AND CULLER, D. E. 2002. MICA: a wireless platform for deeply embedded networks. *IEEE Micro 22*, 6, 12–24.

JOE, I. AND RYU, H. 2007. A patterned preamble MAC protocal for wireless sensor networks. In *Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN'07)*. 1285–1290.

JURDAK, R., BALDI, P., AND LOPES, C. V. 2005. Energy-aware adaptive low power listening for sensor networks. In *Proceedings of the 2nd International Workshop on Networked Sensing Systems (INSS'05)*. San Diego, CA.

JURDAK, R., BALDI, P., AND LOPES, C. V. 2007. Adaptive low power listening for wireless sensor networks. *IEEE Trans. Mobile Comput. 6*, 8, 988–1004.

KUSHNER, H. J. AND YIN, G. G. 2003. *Stochastic Approximation and Recursive Algorithms and Applications* 2nd Ed. Springer-Verlag.

MAHLKNECHT, S. AND BOCK, M. 2004. CSMA-MPS: a minimum preamble sampling MAC protocol for low power wireless sensor networks. In *Proceedings of the IEEE International Workshop on Factory Communication Systems*. 73–80.

MANGIONE-SMITH, B. 1995. Low power communications protocols: paging and beyond. In *Proceedings of IEEE Symposium on Low Power Electronics*. 8–11.

MEGERIAN, S. AND POTKONJAK, M. 2003. *Wireless Sensor Networks*. Wiley Encyclopedia of Telecommunications. Wiley-Interscience, New York, NY.

MOON, S., KIM, T., AND CHA, H. 2007. Enabling low power listening on IEEE 802.15.4-based sensor nodes. In *Proceedings of the IEEE Wireless Communication and Networking Conference (WCNC'07)*.

NING, X. AND CASSANDRAS, C. G. 2006. Dynamic sleep time control in event-driven wireless sensor networks. In *Proceedings of the 45th IEEE Conference on Decision and Control*. 2722–2727.

NING, X. AND CASSANDRAS, C. G. 2007. Message batching in wireless sensor networks–a perturbation analysis approach. In *Proceedings of the 46th IEEE Conference on Decision and Control*. 939–944.

NING, X. AND CASSANDRAS, C. G. 2008. Optimal dynamic sleep time control in wireless sensor networks. In *Proceedings of the 47th IEEE Conference on Decision and Control*. 2332–2337.

POLASTRE, J., HILL, J., AND CULLER, D. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*. 95–107.

ROBBINS, H. AND MONRO, S. 1951. A stochastic approximation method. *Annals Math. Stat. 22*, 400–407.

SHNAYDER, V., HEMPSTEAD, M., CHEN, B., ALLEN, G. W., AND WELSH, M. 2004. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*. 188–200.

SICHITIU, M. 2004. Cross-layer scheduling for power efficiency in wireless sensor networks. In *Proceedings of IEEE INFOCOM*. 1740–1750.

SOHRABI, K., GAO, J., AILAWADHI, V., AND POTTIE, G. 2000. Protocols for self-organization of a wireless sensor network. *IEEE Pers. Comm. 7*, 5, 16–27.

TIERNEY, L. 1983. A space-efficient recursive procedure for estimating a quantile of an unknown distribution. *SIAM J. Sci. Stat. Comput. 4*, 4, 706–711.

YE, W., HEIDEMANN, J., AND ESTRIN, D. 2004. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw. 12*, 3, 493–506.

Received December 2008; revised March 2009; accepted July 2009