

Spatiotemporal Delay Control for Low-Duty-Cycle Sensor Networks

Yu Gu and Tian He

Department of Computer Science and Engineering
University of Minnesota, Twin Cities, USA
{yugu,tianhe}@cs.umn.edu

Mingen Lin and Jinhui Xu

Department of Computer Science and Engineering
State University of New York at Buffalo, USA
{mlin6,jinhui}@cse.buffalo.edu

Abstract—Data delivery is a major function of sensor network applications. Many applications, such as military surveillance, require the detection of interested events to be reported to a command center within a specified time frame, and therefore impose a real-time bound on communication delay. On the other hand, to conserve energy, one of the most effective approaches is to keep sensor nodes in the dormant state as long as possible while satisfying application requirements. Obviously a node can not communicate if it is not active. Therefore, to deliver data in a timely manner for such extremely low duty-cycle sensor networks, communication needs to be carefully managed among sensor nodes.

In this work, we introduce three different approaches to provide *real-time guarantee* of communication delay. First, we present a method for increasing duty-cycle at individual node. Then we describe a scheme on placement of sink nodes. Based on previous two methods, we discuss a hybrid approach that shows better balance between cost and efficiency on bounding communication delay. Our solution is global optimal in terms of minimizing the energy consumption for bounding pairwise end-to-end delay. For many-to-one and many-to-many cases, which are NP-hard, we propose corresponding heuristic algorithms for them. To our knowledge, these are the most generic and encouraging results to date in this new research direction. We evaluate our design with an extensive simulation of 5,000 nodes as well as with a small-scale running test-bed on TinyOS/Mote platform. Results show the effectiveness of our approach and significant improvements over an existing solution.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have been applied in many real-time application domains such as military surveillance [1], assisted living [2] and infrastructure monitoring [3], [4]. Many of these applications are designed for long-term operations, which may require energy-limited sensors to last as long as several years. Beside energy constraints, these applications require real-time data delivery, which usually has delay constraints in the form of end-to-end (E2E) communication deadlines. For example, a surveillance system may require the positions of intruders to be reported within several seconds, so that timely pursuit actions can be initiated. To resolve the conflict between the limited power supply of sensor devices and the stringent data delivery deadline of applications, it is essential to design communication schemes that can achieve designated data delivery deadline with the minimum energy consumption.

In this work, we focus on *extremely low duty-cycle sensor networks*, in which energy management protocols [5]–[8] aggressively reduce energy consumption by allowing a very low duty cycle for each single sensor device (e.g., $< 1\%$). Essentially, during the operation of sensor applications, sensor nodes activate very briefly and stay in sleep state for a long time. Due to the extremely limited energy budget, it is infeasible to maintain a ready-to-use communication backbone continuously. For any given time frame, this type of networks may be fragmented (partitioned) and network connectivity (topology) becomes time-dependent. In other words, when a node has packets ready to be sent, all of its neighboring nodes may be in the dormant state and the sender may have to wait for one of its neighbors to wake up in order to forward its packet. The time spent on waiting for a neighbor to wakeup at the sender is called *sleep latency*. The sleep latency is usually in the order of seconds, which is much longer than other delivery latency (e.g., processing delay, transmission delay and propagation delay) with a delay in the order of milliseconds during the packet transmission. Therefore, the sleep latency dominates the E2E delay in such extremely low duty-cycle sensor networks.

While many emerging protocols are yet to be developed for this new type of networks, in this work we are aiming at *providing energy efficient schemes to guarantee real-time delay bound for extremely low duty-cycle sensor networks*. Specifically, the intellectual contributions of this work lie in:

- To the best of our knowledge, we are the first to design generic algorithms to bound the communication delay in extremely low duty-cycle sensor networks. We introduce three energy efficient solutions that optimize (i) duty-cycle at individual node, (ii) number of necessary sink nodes, and (iii) balance between cost and efficiency for integrating previous two approaches. These solutions effectively provide real-time guarantee of pairwise and many-to-many communications.
- We demonstrate an optimal solution for bounding pairwise communication. In addition, we propose suboptimal algorithms for many-to-one and many-to-many communications, which are NP-hard to obtain optimal results.

- We evaluate our solutions with a running system implementation as well as a 5000-node large scale simulation. The results from both evaluation platforms show significant better energy efficiency for bounding E2E delays than a state-of-the-art solution [9].

The rest of the paper is organized as follows: Section II describes the network model. Section III proposes three solutions to bound the E2E delay in extremely low duty-cycle networks. Practical issues are discussed in Section IV. We describe our simulation results in Section V, followed by system implementation and evaluations in Section VI. Section VII briefly discusses the related works. Section VIII concludes the paper.

II. MODELS AND ASSUMPTIONS

Before presenting our design in detail, we state the models and assumptions used in this work. To simplify description, we present our design with time slots in a synchronized mode. However, our design works without time slots and requires only very low-cost local synchronization. In other words, our design works in CSMA networks where nodes are duty-cycled by upper-layer scheduling algorithms. In Section IV, we will further discuss those issues in detail.

A. Network Model

We assume a network with N nodes. At any time t , each node can be either in an active or a dormant state. When a node is in the active state, it can sense the environment, transmit a packet or receive a packet. When a node is in the dormant state, it turns off all function models except the a timer to wake itself up. Specifically, a dormant node transits to the active state either when it is scheduled to switch to the active state or it has a packet to send to a neighboring node that is active at that time. Formally, we denote the network topology at time t using $G(t) = (V, E(t))$, where V is a complete set of nodes with cardinality of N , and $E(t)$ is a set of directed edges at time t . An edge e_{ij} belongs to $E(t)$ if and only if node n_i can communicate with node n_j , and n_j is active at time t . Essentially, $G(t)$ represents the possible traffic within the network at time t . In other words, a node can *transmit a packet at either active or dormant state, however can only receive packets when it is at the active state.*

We represent the states of each nodes n_i with a working schedule $\Gamma_i = (\omega_i, \tau)$.

- ω_i is an infinite binary string, in which 1 denotes the active state and 0 denotes the dormant state. Consequently, the duty cycle of a node is the percentage of 1s in the binary string. Since the working schedules of the sensor nodes are normally periodic, the infinite binary string ω_i can be described using a regular expression. For example, a regular expression $(101)^*$ represents a periodic active-dormant-active working schedule.

- The state transitions between active and inactive states are time-driven. We use τ to denote the time-span of a bit in the binary string ω_i represents. For example, the total time-span of the binary string (00101) with τ of 2 seconds is 10 seconds.

We note that this simple 2-tuple (ω_i, τ) is generic enough to represent arbitrary working schedules. Theoretically, when $\tau \rightarrow 0$, ω_i can precisely characterize any on/off behavior of a node n_i . For the sake of clarity in presentation, we begin our design with a simplified assumption that it takes at most time τ to transmit one packet between two nodes successfully. We will address the practical issues related to packet loss and queuing delays in Section IV.

B. Delay in Data Delivery

Obviously, if all nodes are always in active state, E2E delay in the above network model equals $H\tau$, where H is the number of hops between a source and a destination¹. However, if nodes in a network have certain working schedules $\Gamma = \{\Gamma_1, \Gamma_2, \dots, \Gamma_N\}$, transmission at each hop could be delayed, waiting for the receiver to wake up. Consequently, the sleep latency between nodes i and j at time t , $d_{ij}(t)$, can be defined as: if t is the *packet ready time* at node i , then $t + d_{ij}(t)$ is the *packet arrival time* at node j . In other words, $d_{ij}(t)$ is the time difference between t and the first active bit in ω_j after time t . For example, assuming sender i and receiver j has working schedule $\Gamma_i = ((10000)^*, \tau)$ and $\Gamma_j = ((00010)^*, \tau)$ respectively, the sleep latency at time 0 ($d_{ij}(0)$) is $3\tau - 0\tau = 3\tau$ since the first wakeup time after time 0 at receiver j is at time 3.

III. BOUNDING COMMUNICATION DELAY

In an extremely low duty-cycle sensor network, the full connectivity is not always guaranteed. Consequently, the time spent on delivering packets over multiple hops would be excessive if without coordination. Such a delay may be tolerable for applications which do not require timely data delivery, however, for other applications (e.g., military surveillance and hazardous monitoring), the sensed events must be received within specified time frames to meet application requirements. However, many energy management protocols such as sensing coverage [10], [11] and target tracking [12], [13], although are very effective to minimize energy consumption within the network, they *rarely consider the impact of resulting node working schedules on communication delay.* In this work, we take a step further and aim at bridging the gap between energy management protocols and effective communication in sensor networks.

Under light traffic load, the E2E delay in extremely low duty-cycle sensor networks is dominated by sleep latency along a multi-hop route instead of congestion. Motivated

¹In extremely low duty-cycle sensor networks, traffic loads are very low. Therefore the queuing delay and collision/interference can be effectively ignored for analysis purposes

by this observation, we propose two effective schemes to bound the E2E delay while preserving the original node activities that are specified by the power management protocols (e.g., sensing patterns), namely active bits augmentation (Section III-A) and sink augmentation (Section III-B). The former addresses the issue from the *temporal (time)* perspective, by adding new active bits to the existing schedule of individual nodes, in order to reduce the sleep latency. The latter addresses the issue from the *spatial (space)* perspective by adding a minimum number of sinks, in order to reduce the number of communication hops. After demonstrating that both schemes have their advantages as well as limitations, in Section III-C we propose a hybrid *spatiotemporal* scheme that achieves good balance between cost and energy efficiency while satisfying real-time requirements.

A. Active Bits Augmentation Scheme

In this section, we provide E2E delay guarantee by adding extra active bits to nodes within a sensor network $G = (V, E)$ whose node working schedules have been generated by energy management protocols. For example, in a linear network shown in Figure 1, the original sleep latency between node A and node B is 2τ if node A has a packet ready to be sent at time 1 (the first active bit). However, after adding an extra active bit to node B and changing its working schedule from $(100)^*$ to $(101)^*$, we reduce the sleep latency to τ and maintain the same level of node behaviors generated by the energy management protocols (node B is still in the active state at the first time interval). Therefore, our active bits augmentation scheme attempts to bound the E2E delay while augmenting a minimum number of active bits in the network. Formally, we define the problem as *Minimum Bits Augmentation for Bounded Delay (MBABD)*:

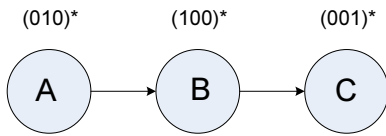


Figure 1. A Linear Network

MBABD Problem: Given a network $G = (V, E)$ with the node working schedule $\Gamma = (\omega, \tau)$, find how to augment a minimum number of active bits into working schedule ω , so that E2E delay from each K designated reporting sensor nodes to sink d is below a E2E delay bound D .

The augmentation problem we dealt with here is more generic than an allocation problem, since allocation is one special case of augmentation with empty configuration.

1) *Bounding Pairwise Communication Delay:* When $K = 1$, we need to bound the delay for pairwise communication between a designated reporting node and a sink. For this case, we introduce an *optimal polynomial* solution using dynamic programming.

Before introducing our optimal solution, we first present two observations on how active bits should be augmented so as to achieve optimality: 1) since a node only senses surrounding environment and receives packets from its neighbors at its active state, the node only has a packet ready to be sent at its active state. 2) All augmented bits should reduce the length (sleep latency) of an original edge to τ . This observation is based on FIFO property of the network model described in section II. In a FIFO network, the packet arrival order at destination is the same as sending order at origin. Since waiting in FIFO networks can never reduce E2E delay [14], an intermediate relay node should forward the packet as soon as possible. Therefore, an augmented active bit should reduce the delivery delay between a sender and a receiver to a minimum possible value τ .

To apply dynamic programming, we first need to discuss the *optimal substructure* of MBABD problem for pairwise communication. In order to represent intermediate states of the MBABD, we define the following function:

$l_{ij}(m, h)$: The earliest time a packet arrives at node j after traversing at most m edges from node i . Among m edges, the sleep latencies of h edges are reduced to τ by adding h active bits along the path.

To further illustrate above intermediate state function, let us take linear network in Figure 1 as an example. Assuming node A has a packet ready to be sent at its active bit (time 1), then $l_{AB}(1, 0)$ represents the earliest packet arrival time at node B by traversing one edge e_{AB} without any active bit augmentation; $l_{AB}(1, 1)$ means the earliest packet arrival time at node B by traversing the edge e_{AB} with exact one active bit augmentation at node B that reduces the edge length of e_{AB} to τ ; while $l_{AC}(2, 1)$ denotes by traversing two edges e_{AB} and e_{BC} , the earliest packet arrival time at node C with the edge length of either e_{AB} or e_{BC} reduced to τ . The final solution of MBABD problem, therefore can be represented as $l_{id}(n - 1, h)$, where n is the number of nodes in the network.

- **Initial State:** For the simple shortest path of a n -node network, the maximal number of edges (m) could be traversed from a node to another node is $n - 1$. When $m = 1, h = 0$, for any pair of node i and node j , $l_{ij}(1, 0)$ simply represents the sleep latency from node i to node j (d_{ij}). Similarly if $m = h$, the value of $l_{ij}(m, h)$ is just $m\tau$ since we add h active bits along the path from node i to node j without any extra sleep latency. Consequently, we can have following initial states:

$$l_{ij}(m, h) = \begin{cases} d_{ij} & m = 1, h = 0 \\ m\tau & m = h \end{cases} \quad (1)$$

- **Recursive Solution:** We define the cost of an optimal solution recursively in terms of the optimal solutions

to subproblems. For any node j , the earliest packet arrival time from the source node i consists of two possibilities:

- The packet could be firstly transmitted from node i to a certain intermediate node p (possibly through multiple hops), and then go directly from node p to node j through one single edge without augmenting any additional active bit to node j .
- The packet could also firstly be delivered to the immediate node p (possibly through multiple hops), and then add an active bit to node j so as to reduce sleep latency between node p and node j to τ if the path from node i to node p has less than h augmented bits.

Therefore, to solve the MBABD problem, we iteratively calculate $l_{id}(n-1, h)$ by increasing the value of h , until the E2E delay bound D has been met. For finding the value of $l_{id}(n-1, h)$, we need to solve its subproblems of finding solutions from node i to all the neighboring nodes (P) of sink node d with maximal number of augmented bits being h (i.e., $l_{ip}(n-2, h), p \in P$) and $h-1$ (i.e., $l_{ip}(n-2, h-1), p \in P$), respectively. According to above analysis, we have following recursive equation:

$$l_{ij}(m, h) = \text{Min} \begin{cases} l_{ij}(m, h), \\ l_{ip}(m-1, h-1) + \tau, & m > h > 0 \\ l_{ip}(m-1, h) + d_{pj}(l_{ip}(m-1, h)) \end{cases} \quad (2)$$

To further illustrate above dynamic programming process, we provide a simple walkthrough of the described algorithm on the linear network shown in Figure 2. In Figure 2, we assume node A has a packet ready to be sent to node C at time 0 and the delay bound D is 3τ .

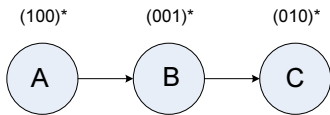


Figure 2. A Linear Network

For initial state, according to the given node working schedules, the sleep latency between node A and node B (d_{AB}) is 2τ . In addition, when the number of edges traversed from node A to other nodes is the same as the number of augmented active bits along the path, there is no extra sleep latency. Therefore, we can obtain following initial states:

$$l_{AB}(1, 0) = 2\tau, \quad l_{AB}(1, 1) = \tau, \quad l_{AC}(2, 2) = 2\tau.$$

Since $l_{AC}(2, 2) = 2\tau < 3\tau$, we assure that delay bound D is achievable. Built upon above initial states, the communication delay from node A to node C through intermediate node B without any bit augmentation is:

$$l_{AC}(2, 0) = l_{AB}(1, 0) + d_{BC}(l_{AB}(1, 0)) = 2\tau + 2\tau = 4\tau$$

Because of $l_{AC}(2, 0) = 4\tau > 3\tau$, we can conclude that in order to achieve the designated delay bound, we have to augment at least one active bit along the path from node A to node C. Therefore, for the case of augmenting only one active bit, we have the options of adding the active bit at either node B or node C. According to Equation 2, we have:

$$\begin{aligned} l_{AC}(2, 1) &= \text{Min} \begin{cases} l_{AB}(1, 0) + \tau, \\ l_{AB}(1, 1) + d_{BC}(l_{AB}(1, 1)) \end{cases} \\ &= \text{Min} \begin{cases} 2\tau + \tau, \\ \tau + d_{BC}(\tau) \end{cases} \\ &= \text{Min} \begin{cases} 3\tau, \\ \tau + 3\tau \end{cases} \\ &= 3\tau \end{aligned}$$

From above equations we can see by augmenting an extra active bit to node C (represented as $l_{AB}(1, 0) + \tau$), and changing its working schedule from $(010)^*$ to $(110)^*$, we meet the delay bound 3τ with a minimum number of active bits augmentation. The total running time of the complete dynamic programming algorithm is $\mathcal{O}(n^3 H^2)$, where n is the number of nodes in the network and H is the network diameter.

2) *Bounding many-to-many Communication Delay:* Besides the pairwise communication, many-to-one and many-to-many communication are another two typical traffic patterns in sensor network applications where single or multiple sinks are used to gather data from the sensor networks, which represent the cases where $K > 1$. However, both many-to-one and many-to-many versions of MBABD problem are NP-hard, which can be deduced from planar 3-Satisfiability problem. The detailed proof of NP-Hardness is omitted due to the space constraints.

Since it is NP-hard to solve the MBABD problem for the many-to-one and many-to-many cases, we resort to a greedy heuristic algorithm. This algorithm incrementally augments active bits into the network. For each augmentation, the algorithm chooses the active position that maximally decreases the sum of E2E delays between all K designated reporting nodes and Q sink nodes. The computational complexity of this algorithm is $\mathcal{O}(KQn^3 H^2)$, if $KQ < n$, otherwise $\mathcal{O}(n^4 H^2)$.

B. Sink Augmentation Scheme

To bound E2E delay, one can either reduce the per-hop delay (by augmenting active bits) or reduce the total number of E2E hops (by augmenting more sinks). In this section, we demonstrate how to bound E2E communication delays by augmenting sinks. In practice, in order to add additional sink nodes, we can attach radio modems with additional power supplies to regular nodes and convert them to sink nodes. For example, Y-Lynx long range radio modem [15] has very small form factor with $77 \times 29 \times 5mm$ in size, requires as low as 2.4V for power supply, has a current draw of around

500mA while the communication range reaches as long as 30,000 meters in open space, which is usually sufficient long enough to reach final data-gathering point.

Formally, we define the problem *Minimum Sink Augmentation for Bounded Delay* (MSABD) as:

MSABD Problem: Given a sensor network $G = (V, E)$ with the node working schedule $\Gamma = (\omega, \tau)$, select a minimum subset of nodes $B \subseteq V$ as sink nodes, so that for any of K designated reporting nodes (a subset of nodes $I \subseteq V$), the E2E delay to one of the sinks in B does not exceed delay bound D .

This problem is NP-hard as well. The proof is straightforward. For each designated reporting node $i \in I$, by running the algorithm in Section III-A with zero active bit augmentation (i.e., the shortest path algorithm for time-dependent networks), we can obtain a set of nodes such that the E2E delay from i to any node in that set is smaller or equal to delay bound D . Then for any node j in the network, we can create a collection of sets C_j , such that each element in C_j is the node that can reach node j within delay bound D . The problem of MSABD, then can be transformed to:

Given a collection of sets C_j (SC), where $\bigcup C_j = I$ and $j = 1, 2, \dots, |V|$, find a minimum number of sets among SC , such that $\bigcup C_m = I$, $m \in \{1, 2, \dots, |V|\}$.

The new problem definition then is identical to the well-known NP-hard *set cover problem*. To better explain the problem transformation from MSABD to set cover, we provide a walkthrough example below.

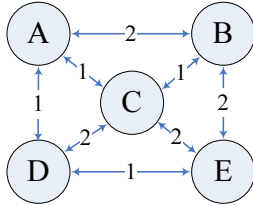


Figure 3. Example of MSABD to Set Cover Transformation

As shown in Figure 3, we have a 5-node network that each node needs to report their readings to a sink within a delay bound 1. For the purpose of illustration, in Figure 3 we assume the sleep latencies between pairs of nodes are bi-directional. The values of sleep latencies are labeled on the edges connecting them. After running the shortest path algorithm at all nodes with given sleep latency, at each individual node, we can create a set that records the nodes that can reach itself within delay bound 1 and list as below:

$$C_A = \{A, C, D\}, \quad C_B = \{B, C\}, \quad C_C = \{A, B, C\} \\ C_D = \{A, D, E\}, \quad C_E = \{D, E\}$$

Therefore, the MSABD problem transforms to the problem of finding a minimum number of sets from C_A to C_E such that node A to node E are all covered, which is a typical set cover problem setting.

To solve the set cover problem heuristically, we adopt the rule that repeatedly choose the set that contains the largest number of uncovered element, until a full cover has achieved. Alon et al. [16] have proved the inapproximability of set cover problem. This heuristic, therefore, is essentially the best-possible polynomial time approximation algorithm for set cover problem under plausible complexity assumptions. Briefly, the algorithm goes as follows: for each designated reporting node $i \in I$, we calculate its shortest path to all other nodes in the network. Upon above information and application specified communication delay bound, for each node $j \in V$ in the network, we compute the designated reporting nodes that can be covered by node j so as to meet their communication delay bounds. Finally we continuously chooses the sink that can cover most uncovered reporting nodes until all of them can meet their deadlines.

According to the output of algorithm, we can add these sinks and the reporting nodes transmit their packets to the nearest sink that meets their communication deadline.

C. Hybrid Scheme

As shown in the previous sections, both active bits augmentation scheme and sink augmentation scheme are effective in bounding E2E delay. On the other hand, energy efficiency is another critical metric for measuring the performance of low duty-cycle sensor networks. In this section, we demonstrate the deficiency of applying either active augmentation or sink augmentation alone, and introduce a hybrid design that exploits the advantages of both schemes while minimizes their limitations.

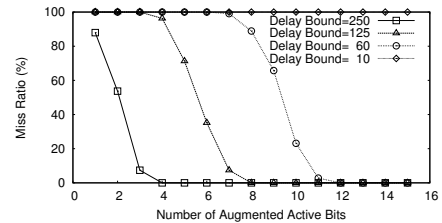


Figure 4. Deadline Miss Ratio vs. Number of Augmented Active Bits(Node Density=1)

1) *Energy Efficiency of the Active Bits Augmentation Scheme:* In order to understand the energy efficiency of active bits augmentation scheme, we studied the E2E miss ratios under different energy budgets and different communication delay bounds in simulation. The simulation setup is consistent with the evaluation presented in Section V. Figure 4 illustrates the miss ratios of messages delivered between reporting nodes and a sink under 15 different energy consumption budgets with delay bounds vary from 10 to 250 units of time. From the figure, it is clear that the longer delay bound is, the more energy effective for active bits augmentation scheme. For example, under delay bound of 250 units of time, the miss ratio drops to zero percent when the number of augmented active bits reaches 4. In contrast,

under delay bound of 60 units of time, 12 augmented active bits are necessary to deliver all the messages in time.

More crucially, we can see for very tight delay bound, such as 10 units of time, no matter how many active bits we add into network, the miss ratio for message delivery is always 100%. This indicates the active bits augmentation scheme is *not effective* when a designated delay bound is too small for a reporting node to forward its message continuously to a sink.

2) *Energy Efficiency of the Sink Augmentation Scheme:* Figure 5 studies the energy efficiency of the sink augmentation scheme in simulation and shows the average number of reporting nodes covered by augmented sinks in the order of augmentation sequence. From Figure 5, we can see the average number of reporting nodes that covered by augmented sinks drops significantly as number of augmented sinks increases. For example, when delay bound is 30 units of time, the first augmented sink can cover 154.4 reporting nodes, while the tenth sink can only cover approximately 0.68 reporting node. Such a huge gap of node coverage among early and late augmented sinks reveals the inefficiency and unbalanced cost of sink augmentation scheme. This energy inefficiency of sole sink augmentation scheme, therefore, further motivates the design of hybrid scheme in the next subsection.

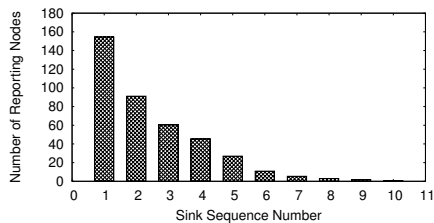


Figure 5. Number of Nodes Coverage vs. Sink Sequence Number (Delay Bound=30)

3) *Hybrid Design:* As discussed in the previous two subsections, both active bits and sink augmentation schemes have certain drawbacks and limitations. In active bits augmentation scheme, if the designated communication deadline is shorter than the minimum time to forward a packet without sleep latency, no matter how many active bits are added, the packet will miss the deadline. While for the sink augmentation scheme, the last a few augmented sinks may only cover a very small number of nodes, leading to a high-cost network.

To address both limitations mentioned above, in this section, we propose a hybrid solution, which tries to maximize the benefits of active bits augmentation and sink augmentation while overcoming the drawbacks and limitations of them. The major advantage of sink augmentation scheme is by carefully augmenting sinks, it can effectively bound E2E delay for a fairly large group of sensor nodes and there is no change of working schedules for those nodes. However, augmenting sinks requires additional hardware cost and human

intervention. On the other hand, active bits augmentation requires little human intervention at the cost of increasing single node energy consumption. Such increase in energy consumption may be wasteful if the presence of interested events in network is rare and the node is turned on without receiving or forwarding any message most of the time. We note that especially in the extremely low duty-cycle sensor networks, nodes are very sensitive to the increase in duty cycles, even a single added active bit may affect the lifetime of single node considerably. For example, if a 1% duty cycle node has a working schedule with a length of 100 bits, one augmentation of active bit at this node would cut node life time to half of its original life time. Therefore, finding a balanced configuration between active bits augmentation and sink augmentation is essential to achieve efficient power and cost management during the deployment of sensor networks. In order to find such balance point, depending on specific applications and physical environment, we can decide the cost ratio of augmenting a sink over augmenting one active bit based on the hardware cost, lifetime of sinks and sensor nodes, human intervention cost and so on.

With the known cost ratio, the hybrid scheme goes as follows. Firstly, identical to the transformation from MSABD to set cover in sink augmentation section, each node in the network maintains a set of nodes (cover set) that can reach itself within the delay bound D . Starting with the node i that has the largest cover set, we calculate the total cost of meeting delay bound for all nodes in C_i by augmenting active bits. Then we compare the sink augmentation cost with the active bit augmentation cost. The scheme with smaller cost is chosen and all nodes in C_i are removed from cover sets contain them. This comparing and selecting process repeats until all designated reporting nodes can meet the delay bound.

IV. PRACTICAL CONSIDERATION

This section completes our design with five important practical design issues concerning time synchronization, CSMA networks, distributed implementation of active bits augmentation scheme, packet loss and packet queuing. Due to space constraints, we can only provide design insights without too many details.

A. Local Time Synchronization

In Section III, we describe our design in a synchronized fashion. However, our network model and corresponding designs do not depend on time slots or global time synchronization. To obtain optimal solutions, our design only requires that each node knows the *working schedules* of its forwarding nodes. To interpret the working schedule of the forwarding node, only local synchronization is necessary. Techniques such as MAC-layer time-stamping [17] can achieve $2.24\mu s$ accuracy with an overhead of a few

bytes of packets exchange among neighboring nodes. Since the expected τ value ranges from $2,000\mu s$ to $20,000\mu s$ (according to the data rates of different radio chips), an accuracy of $2.24\mu s$ is by far sufficient.

B. Working under CSMA Networks

Unlike TDMA networks, where each node has designated time slots to transmit its packets, our solution allows each node to initiate its transmission at any time instance. In addition, in terms of performance, CSMA is a more favorable choice in extremely low duty-cycle sensor networks where traffic is very low, since a node in TDMA networks has to wait for its time slots to transmit and becomes inefficient. Finally, since we do not assign specific time slots to each individual nodes, for a given time instance, packets sent from different senders may collide at their same destined receiver. In such scenarios, CSMA is helpful in avoiding potential but rare collisions in low traffic low duty-cycle sensor networks.

C. The Impact of Packet Loss

In the basic design, we assume that it takes τ (a unit of time) to successful transmit a packet. In reality, this assumption could not hold well especially when the communication quality is very low (hence multiple retransmissions would occur). In traditional networks, packet loss normally does not affect the routing path on the fly. However, in extremely low duty-cycle sensor networks, packet loss would introduce a delay that is much longer than the retransmission time, because of the need to defer the transmission to the time when the receiver wakes up again. Therefore, we should find out another optimal route by switching from original receiver to another receiver if the latter activates earlier [18]. We note that if the maximum number of retransmissions is known at each node, we can extend the current real-time design to accommodate the impact of packet loss. The main idea is to assign the worst-case edge length $e(i, j)$ based on the maximum number of retransmissions allowed by a network protocol.

V. LARGE-SCALE SIMULATION

In order to understand the performance of the proposed schemes under numerous network settings and exploit the scalability of different design approaches, in this section, we provide simulation results with 5,000 nodes.

A. Simulation Setup

In the simulation, up to 5,000 sensor nodes are randomly deployed in a $300m \times 300m$ square field. From all deployed sensor nodes, we randomly choose 200 nodes as designated reporting nodes. The sensing range and communication range are 10m and 20m respectively. The working schedules of sensor nodes have lengths of 200 bits with the duty cycles of one percent. The actual positions of active bits are generated randomly to test a wide range of configurations.

All experiments are repeated 100 times with different random seeds, node deployments and nodes working schedules.

B. Performance for Active Bits Augmentation Scheme

In this section, we investigate the effectiveness of adding active bits to bound the communication delay among reporting nodes and sinks. In order to better reveal the insights of the active bits augmentation scheme, in this experiment, we set the locations of the sending nodes to be on the leftmost margin of the sensed field. The sink which is always active and ready to receive messages, is placed on the rightmost side of field. To compare the performance of the proposed scheme, the following baseline is adopted:

- **Baseline: Streamlined Wake-up Scheme:** Cao [9] proposes a streamlined wake-up scheme to minimize the E2E delay for a single sink case. This scheme first labels the sensor nodes according to the shortest hop count to the sink. They then build a route in which every node on the route is able to successfully transmit the packet to the node that is one-hop closer to the sink right after receiving the packet. This idea is analogous to turn traffic lights green just in time for the arrival of vehicles from previous intersections.

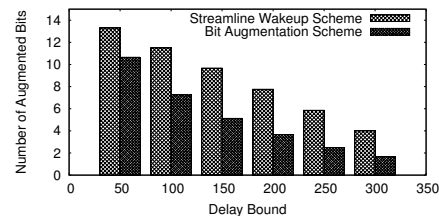


Figure 6. Number of Augmented Bits Vs. Delay Bound(Node Density=1)

1) Impact of Designated Delay Bound on Energy Budget:

Figure 6 shows the impact of delay bound on minimum number of active bits which have to be added so as to achieve the delay bound between the sensing node and the sink. From Figure 6, we can see as the delay bound increases, the minimum number of added active bits decreases accordingly for both the streamlined wake-up scheme and the active bits augmentation scheme. However, at each delay bound, our active bits augmentation scheme outperforms the baseline scheme significantly. When the delay bound is between 150 and 300 units of time, which requires relatively a smaller amount of delay reduction from the original delay, the number of added bits for the baseline almost doubles that of the active bit augmentation scheme. For example, when a delay bound such as 300 units of time is required, compared with 4.01 bits for the streamlined wake-up scheme, the active bit augmentation scheme only adds an average of 1.66 active bits to the network, which is almost 2.5 times more energy efficient. For tighter deadlines such as 50 and 100 units of time, we observe less performance gain, this is because that such a tight delay bound enforces the addition of one

active bit to almost each node on the shortest path from the reporting node to the sink. This essentially builds a traffic light-like path, which leaves little design space for the optimal bit augmentation algorithm to exploit. Even under such extreme conditions, we still gain better performance than the baseline scheme, with an average of 10.66 added active bits in comparison to 13.30 bits for the streamlined wake-up at the delay bound of 50 units of time.

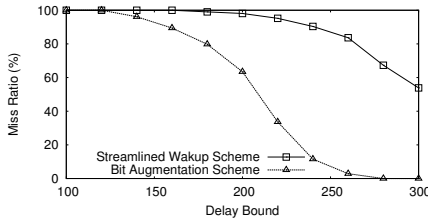


Figure 7. Deadline Miss Ratio Vs. Delay Bounds (# Augmented Bits=3)

2) *Impact of Designated Delay Bound on Miss Ratio:* In this experiment, we investigate the scenario where only a fix number of active bits can be added. Under a given energy budget, Figure 7 shows the miss ratios of the active bit augmentation scheme and the baseline scheme under different designated delay bounds. For both schemes, the miss ratios decrease as we extend the bound on the communication delay. Curves in Figure 7 further demonstrate the optimality of our pairwise active bits augmentation solution. For every single delay bound, the miss ratio for the bit augmentation scheme is much lower than that of the streamlined wake-up approach. When the delay bound is above 273 units of time, our active bits augmentation scheme can guarantee the 100% delivery of messages from the reporting node to the sink within the designated transmission deadline, while the baseline scheme can only deliver approximately 29% of the messages on time. This result, along with the findings in Figure 6, demonstrate that simply building a traffic light-like path from the sink to the reporting node, which is adopted in the streamlined wake-up scheme [9], is not energy efficient. In order to minimize the additional energy consumption, one should explore all potential nodes that can be used to relay a message, with the cost of larger computational complexity.

C. Performance for Sink Station Augmentation Scheme

This section investigates the efficiency of augmenting additional sinks to bound the communication delay between the reporting nodes and the sinks.

1) *Impact of Designated Delay Bound:* Figure 8 shows the number of augmented sinks under different delay bounds. As the delay bound becomes longer, the number of required sinks drops sharply. As delay bound increases from 10 units of time to 100 units of time, the number of augmented sinks decreases from 31.78 to 1. Such a trend implies the sink augmentation scheme is very effective when the delay bound is not tight.

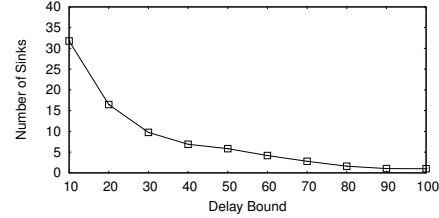


Figure 8. Number of Sinks Vs. Delay Bound(Node Density=5)

Figure 9 depicts the E2E miss ratios under different delay bounds while the number of added sinks increases. Consistent with previous results, a longer designated delay bound leads to a smaller miss ratio. For example, when only one sink is allowed to place in the field, the miss ratios for delay bounds of 10, 20, 30 and 40 units of time are 89.17%, 77.59%, 61.38% and 41.15% respectively.

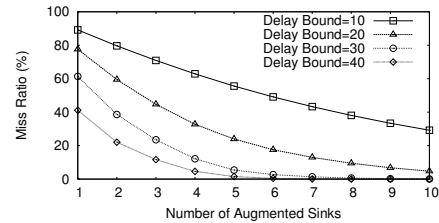


Figure 9. Deadline Miss Ratio Vs. Number of Augmented Sinks (Node Density=5)

D. Performance for Hybrid Scheme

As stated earlier, both active bits augmentation and sink augmentation schemes have their limitations and disadvantages. In this section, we evaluate the performance of our third scheme, a combined approach of above two schemes. During the experiment, we use cost ratio from 5 to 25 for augmented sink over augmented active bits in the network. According to the simulation configuration, where each sensor node has a 200-bit working schedule with duty cycle of 1% (2 active bits), the cost ratio for sink node is also equivalent to the lifespan of 2.5 nodes to 12.5 nodes.

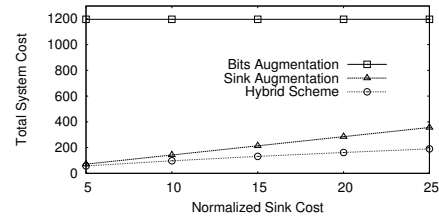


Figure 10. Total System Cost Vs. Normalized Sink Cost(Density=1)

Although all three schemes are able to bound E2E delay, the costs to achieve it are different. In Figure 10, we compare the total system cost for all three proposed schemes. From Figure 10, it is clear that under all the normalized sink cost values, the total system cost for the hybrid scheme is the smallest. For example, when normalized sink cost

is 25, the total system cost for active bits augmentation, sink augmentation and hybrid schemes are 1196.8, 356.3 and 190.0 respectively. This result justifies our design goal of hybrid scheme, eliminating disadvantages of previous solutions and providing better energy efficiency.

Figure 11 shows under different normalized sink costs, the average number of active bits and sinks augmented. As normalized sink cost increases, the number of augmented active bits increases linearly, while the number of elected sinks drops steadily. From normalized sink cost of 5 to 25, the number of augmented active bits rises from 11.2 to 57.8, while the number of sinks decreases from 9.2 to 5.3. Such trend is consistent with energy efficiency results from previous two sections, as the normalized sink cost becomes larger, sink augmentation scheme becomes less effective than the active bits augmentation scheme.

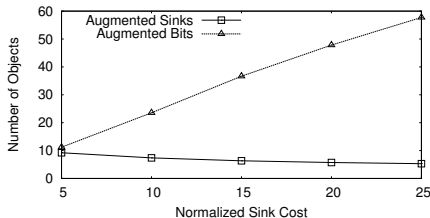


Figure 11. Number of Sinks and Augmented Active Bits Vs. Normalized Sink Cost(Density=1)

VI. SYSTEM IMPLEMENTATION AND EVALUATION

Extensive simulation results in Section V demonstrate the effectiveness of our three proposed schemes on bounding the communication delay between the reporting nodes and the sinks. In order to evaluate the applicability of our design on real sensor networks, we have experimented proposed protocols on our large indoor test-bed composed of six 4-foot by 8-foot boards, as illustrated in Figure 12. Each board in the system can be used as an individual sub-system, which is powered, controlled and metered separately. The protocols are implemented using NesC. The compiled image of a full implementation only adds 838 bytes of code memory and 12 bytes of data memory to the existing sensing applications which switch the state of sensor nodes according to the node working schedules. A simple timer-driven Finite Automaton logic is implemented to turn a mote on/off according to the node working schedules. The nodes in the network are synchronized by FTSP [17], whose accuracy is at tens of microseconds.

To disseminate node schedules to the network and evaluate the performance of our protocols, we implemented an evaluation engine, written in Java and running on a laptop, takes the node working schedules as input, disseminates them into the network, and retrieve the communication delay between the reporting node and the sink.

Due to the space and physical constraints, during the test-bed evaluation, we only evaluate the effectiveness of active



Figure 12. Experiment Setup

bits augmentation scheme in an indoor environment, and leave experiments for sink augmentation and hybrid schemes as our future work. In the experiment, we tuned down node transmission power and built a 5-hop linear sensor network, which consists of six micaZ motes with the leftmost node in the network trying to report the sensed events back to the sink, which is the rightmost mote whose states are always active. The original duty cycle of sensor nodes is set to 1%.

To understand the effectiveness of active bits augmentation scheme to bound the communication delay between the reporting nodes and the sink, we demonstrate the impact of the delay bound on number of augmented active bits added to the network. Figure 13 shows the time for packet delivery with the time-span of a bit (τ) in the node working schedule ω sets to 300ms. Figure 13 shows for a given energy budget and designated delay bound, the active bits augmentation scheme effectively bounds the actual communication delays between reporting node and sink. For example, when the delay bound is 4100ms, by adding one active bit, the active bits augmentation scheme reduces the communication delays to 4024ms, 4037ms, 4059ms, 4056ms and 4030ms under five runs of system experiments. All packet delivery time are below the designated delay bound 4100ms and therefore we achieve our design goal of real-time communication delay guarantee.

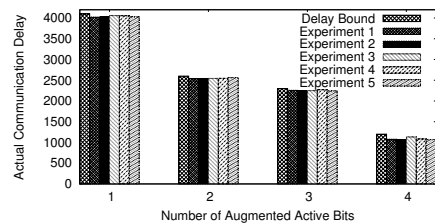


Figure 13. Actual Communication Delay Vs. Delay Bound

VII. RELATED WORK

Real-time communication allows sensor networks to interact with the physical world effectively. In general, three types of approaches are used to achieve this goal. First, timely communication can be achieved by carefully regulating the amount of traffic transmitted locally and globally. Vasudevan [19] proposes an application-specific compression for time delay estimation in sensor networks, and

He [20] adaptively performs application-independent data aggregation in a time sensitive manner. Second, feedback-based real-time communication designs are widely used to cope with the unpredictable nature of wireless communication. RAP [21] uses velocity monotonic scheduling (VMS) to distributedly schedule real-time traffic with the assistance from a differentiated MAC Layer. SPEED [22] uses feedback-based adaptation to enforce per-hop speed in face of unpredictable traffic. As an enhancement to SPEED, Felemban [23] presents a multi-path and multi-speed routing protocol, for probabilistic QoS guarantee in wireless sensor networks. Third, various techniques are proposed to schedule data traffic carefully in order to prevent unbounded collisions and hence provide real-time guarantee. Li [24] proposes a SLF message scheduling algorithm to exploit spatial channel reuse, so that deadline misses can be reduced. Carley [25] designs a periodic message scheduler to provide a contention-free predictable medium access control.

Real-time protocol designs in sensor networks in general are highly diversified. Mobicast [26] disseminates just-in-time information to nodes in a mobile delivery zone. Yang [27] proposes a wakeup scheme that balances energy savings with E2E delays. Nam [28] builds a time-parameterized sensing task model for real-time tracking. Somasundara [29] proposes a mobile agent scheduling algorithm to collect the buffered sensor data, before the buffer overflow occurs at individual sensor nodes. The Lightning protocol [30] localizes the acoustic source with a bounded delay regardless of the node density.

Besides protocol designs, several notable real-time analysis methods have been proposed for sensor networks. Lu [31] shows how to minimize the communication latency given that each sensor has a duty cycling requirement of being awake for only $\frac{1}{k}$ time slots on average. Mohan et al. [32] provides a cycle-accurate WCET analysis tool for the applications running on the Atmega Processor Family. Abdelzaher [33] derives a real-time capacity bound for multi-hop wireless sensor networks.

Recently, there is also a surge of research on low-duty-cycle networks. Keshavarzian et al. propose a multi-parent forwarding technique for low-duty-cycle networks [34]. DSF introduces a concept of dynamic switch-based forwarding [18]. Su et al. discuss both on-demand and proactive algorithms for routing packets in low-duty-cycle networks [35]. TwinStar system provides a working prototype to support sustainable sensor operations by harvesting energy from surrounding environment and storing energy in super capacitors [36]. ESC studies energy synchronization between energy supply and node working schedule in energy-harvesting networks [37]. In addition, several flooding schemes for low-duty-cycle networks also have been investigated [38], [39].

Our work is original and valuable, because it is the first to address real-time issues under an emerging and practical

sensor network model with extremely low duty-cycle. As the first step, we study how to provide real-time guarantee through duty cycle allocation (bit augmentation) and base allocation (placement). Studies on how to apply various techniques in the literature in this new network are yet to be explored in the future.

VIII. CONCLUSION

Extremely low duty-cycle sensor networks have been used in many long-term application scenarios with limited power supplies. These networks can not afford to maintain a ready-to-use communication backbone, leading to a potential fragmented connectivity at any given point of time. Therefore, without coordination, communication would suffer excessive E2E delay due to the sleep latency. In this work, we present three schemes for bounding the communication delay between designated reporting nodes and sink nodes. We derive an optimal dynamic-programming-based solution to bound the pairwise communication, and propose heuristic algorithms for many-to-one and many-to-many communications which are NP-hard to obtain optimal solutions. We demonstrate the effectiveness of our solutions through algorithmic analysis, a running test-bed as well as a 5,000-node simulation. To our knowledge, these are the most generic and encouraging results to date to provide the real-time guarantee of communication delay in extremely low duty-cycle sensor networks.

ACKNOWLEDGEMENT

This research was supported in part by NSF grants CNS-0626609, CNS-0626614 and CNS-0720465.

REFERENCES

- [1] T. He, S. Krishnamurthy, L. Luo, T. Yan, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. A. Stankovic, T. F. Abdelzaher, J. Hui, and B. Krogh, "VigilNet: An Integrated Sensor Network System for Energy-Efficient Surveillance," *ACM TOSN*, 2006.
- [2] J. Ko, T. Gao, and A. Terzis, "Empirical study of a medical sensor application in an urban emergency department," in *BodyNets*, 2009.
- [3] A. Rowe, D. Goel, and R. Rajkumar, "Firefly mosaic: A vision-enabled wireless sensor networking system," in *RTSS 2007*, 2007.
- [4] Y. Kim, T. Schmid, Z. M. Charbiwala, J. Friedman, and M. B. Srivastava, "Nawms: nonintrusive autonomous water monitoring system," in *SenSys '08*, 2008.
- [5] S. Bhattacharya, G. Xing, C. Lu, G.-C. Roman, O. Chipara, and B. Harris, "Dynamic wake-up and topology maintenance protocols with spatiotemporal guarantees," in *IPSN '05*, 2005.
- [6] W. Shu, X. Liu, Z. Gu, and S. Gopalakrishnan, "Optimal sampling rate assignment with dynamic route selection for real-time wireless sensor networks," in *RTSS '08*, 2008.

- [7] A. Lachenmann, P. J. Marrón, D. Minder, and K. Rothermel, "Meeting lifetime goals with energy levels," in *SenSys '07*, 2007.
- [8] Y. Gu, J. Hwang, T. He, and D. H.-C. Du, "uSense: A Unified Asymmetric Sensing Coverage Architecture for Wireless Sensor Networks," in *ICDCS '07*, 2007.
- [9] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards Optimal Sleep Scheduling in Sensor Networks for Rare Event Detection," in *IPSN'05*, 2005.
- [10] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," in *SenSys'03*, 2003.
- [11] Z. Yuan, R. Tan, G. Xing, C. Lu, Y. Chen, and J. Wang, "Fast sensor placement algorithms for fusion-based target detection," in *RTSS '08*, 2008.
- [12] S. Subramaniam, V. Kalogeraki, and T. Palpanas, "Distributed real-time detection and tracking of homogeneous regions in sensor networks," in *RTSS '06*, 2006.
- [13] M. Li, Y. Liu, and L. Chen, "Non-threshold based event detection for 3d environment monitoring in sensor networks," in *ICDCS '07*, 2007.
- [14] R. K. Ahuja, J. B. Orlin, S. Pallottino, and M. G. Scutell, "Dynamic shortest paths minimizing travel times and costs," University of Pisa, Tech. Rep., 2001.
- [15] *YLX - TRM8053-500 Long Range 500mW Radio Modem*, Y-Lynx, available at www.y-lynx.com/socket_modems_TRM8053-500.php.
- [16] N. Alon, D. Moshkovitz, and S. Safra, "Algorithmic Construction of Sets for k-restrictions," *ACM Trans. Algorithms*, 2006.
- [17] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The Flooding Time Synchronization Protocol," in *SenSys'04*, Nov. 2004, pp. 39–49.
- [18] Y. Gu and T. He, "Data Forwarding in Extremely Low Duty-Cycle Sensor Networks with Unreliable Communication Links," in *SenSys '07*, 2007.
- [19] L. Vasudevan, A. Ortega, and U. Mitra, "Application-Specific Compression for Time Delay Estimation in Sensor Networks," in *ACM SenSys 2003*, 2003.
- [20] T. He, B. M. Blum, J. A. Stankovic, and T. F. Abdelzaher, "AIDA: Adaptive Application Independent Data Aggregation in Wireless Sensor Networks," *ACM TECS, Special issue on Dynamically Adaptable Embedded Systems*, 2004.
- [21] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks," in *IEEE RTAS*, 2002.
- [22] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Ad Hoc Sensor Networks," in *ICDCS'03*, May 2003.
- [23] E. Felemban, C. Lee, E. Ekici, R. Boder, and S. Vural, "Probabilistic QoS Guarantee in Reliability and Timeliness Domains in Wireless Sensor Networks," in *IEEE INFOCOM 2005*, 2005.
- [24] H. Li, P. Shenoy, and K. Ramamritham, "Scheduling Messages with Deadlines in Multi-hop Real-time Sensor Networks," in *RTAS'05*, 2005.
- [25] T. W. Carley, M. A. Ba, R. Barua, and D. B. Stewart, "Contention-free Periodic Message Scheduler Medium Access Control in Wireless Sensor/actuator Networks," in *IEEE RTSS*, 2003.
- [26] Q. Huang, C. Lu, and G.-C. Roman, "Spatiotemporal Multicast in Sensor Networks," in *SenSys 2003*, November 2003.
- [27] X. Yang and N. H. Vaidya, "A Wakeup Scheme for Sensor Networks: Achieving Balance between Energy Saving and End-to-end Delay," in *IEEE RTAS 2004*, 2004.
- [28] M.-Y. Nam, C.-G. Lee, K. Kim, and M. Caccamo, "Time-Parameterized Sensing Task Model for Real-Time Tracking," in *IEEE RTSS 2005*, 2005.
- [29] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines," in *IEEE RTSS*, 2004.
- [30] Q. Wang, R. Zheng, A. Tirumal, X. Liu, and L. Sha, "Lightning: A Fast and Lightweight Acoustic Localization Protocol Using Low-End Wireless Micro-Sensors," in *IEEE RTSS*, 2004.
- [31] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *IEEE INFOCOM 2005*, 2005.
- [32] S. Mohan, F. Mueller, D. Whalley, and C. Healy, "Timing Analysis for Sensor Network Nodes of the Atmega Processor Family," in *IEEE RTSS*, 2004.
- [33] T. F. Abdelzaher, S. Prabh, and R. Kiran, "On Real-Time Capacity Limits of Multihop Wireless Sensor Networks," in *IEEE RTSS*, 2004.
- [34] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup Scheduling in Wireless Sensor Networks," in *MobiHoc*, 2006.
- [35] L. Su, C. Liu, H. Song, and G. Cao, "Routing in intermittently connected sensor networks," in *ICNP*, 2008.
- [36] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z.-L. Zhang, "Leakage-Aware Energy Synchronization for Wireless Sensor Networks," in *MobiSys '09*, 2009.
- [37] Y. Gu, T. Zhu, and T. He, "ESC: Energy Synchronized Communication in Sustainable Sensor Networks," in *ICNP '09*, 2009.
- [38] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links," in *MobiCom '09*, 2009.
- [39] F. Wang and J. Liu, "Duty-cycle-aware broadcast in wireless sensor networks," in *INFOCOM'09*, 2009.