Exploiting Power Differential for Energy Efficient Scheduling in Sensor Networks

Sha Liu, Kai-Wei Fan, Prasun Sinha
{liusha,fank,prasun}@cse.ohio-state.edu
Department of Computer Science and Engineering
The Ohio State University

Abstract

In sensor networks it is critical to conserve energy as replacing batteries is an expensive manual operation. Energy needs to be saved when there is no activity in the sensor field and also when events are being detected. Current packet scheduling approaches for sensor networks that address energy consumption require high coordination traffic leading to high latencies and low channel utilization. Our goal is to design an energy efficient scheduling protocol for sensor networks while maintaining high throughput and low latency. We propose BSMac, a scheduling protocol based on a new architecture, that leverages the difference in power at the sink and the sensor nodes. In this architecture, termed BoostNet, the sink is attached to a base-station with an unlimited power supply. We use a simple sequential coloring scheme starting with the least used color on the first link. Time is divided into cycles where each cycle consists of a DCF slot and k dedicated colored slots. The BoostNet architecture naturally provides support for optimizing the number of colors by conducting experiments with different values of k. The high power base-station is useful for synchronizing time and for globally coordinating optimization experiments. We have evaluated the performance of BSMac using simulations and analysis. In comparison to CSMA based approach, BSMac can reduce energy consumption by up to 80% while maintaining similar throughput and latency.

I. Introduction

"Touching" a deployed sensor for replacing batteries is an expensive manual operation especially for large deployments. Consequently, power conservation is crucial in designing wireless sensor networks with long network lifetime. For hazardous environments such as waste management facilities, and hostile environments such as enemy territories, replacing batteries is simply too dangerous. For certain applications, deployed sensors are impossible to physically recover. One such example is the Glacsweb project [1] where sensors are embedded 60 feet below the surface of the glaciers for studying their movement patterns. Thus, it is critical to design protocols that conserve energy in wireless sensor networks.

Energy conservation is important during periods with no activity and also during occurrence of events. It is critical to reduce packet collisions and save packet transmissions during occurrence of events. But nodes also consume significant energy in idle listening, which can be minimized if nodes can determine when they are expected to send and receive packets. For example, current drawn in Mica2 motes is 8 mA in the idle state as compared to 25 mA during packet transmissions [11]. To facilitate energy savings during event occurrence, smart scheduling can allow nodes to sleep for short periods when a node is neither transmitting nor receiving.

Although packet scheduling in ad-hoc and sensor networks has been an active area of research, scheduling to conserve energy in sensor networks has not received much attention. In the context of ad-hoc networks, the problem of scheduling has been well modeled as a graph theory problem with solutions based on local coordination and backoffs [2][3]. However, such protocols are not energy-aware and are incapable of determining the periods when the node can go to sleep. In [4], authors have proposed centralized per-packet scheduling based on information gathered from all links. Such global coordination requires excessive messaging and cause delays in link scheduling. To overcome the drawbacks of centralized scheduling, TRAMA [5] proposes distributed scheduling at each node based on information collected within a fixed number of hops. Although TRAMA can conserve energy, the scheduling delays and the message overhead of local coordination results in latencies that exceed 100 times the latency of CSMA/CA based approaches. Thus TRAMA is useful only in scenarios where latency is not a metric of performance, which is hardly the case in most sensor networks. **Our goal is to design an energy efficient scheduling protocol for sensor networks while maintaining high throughput and low latency.**

We propose a scheduling protocol called BSMac, that is based on a new architecture, that leverages the difference in power at the sink and the sensor nodes. In this architecture, termed BoostNet (see Figure 1), the sink is attached to a base-station with an unlimited power supply. An unconstrained power source is often available at the sink except in applications where the sink may be mobile [6]. We assume that with its high power, the base-station can reach all the sensors. The sensors communicate in the same channel as the base-station but at a much lower power resulting in a multi-hop network topology. All the sensors can receive packets directly from the base-station, but not all sensors can communicate directly to the base-station. Although in this paper we focus on an architecture with a single base-station, we briefly discuss a scalable extension with multiple such base-stations in Section VII, for the case when the base-station can not reach the entire network.

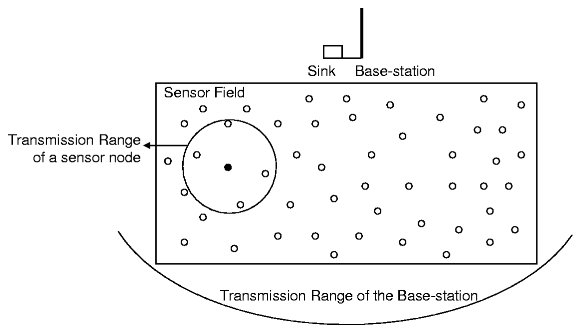

Fig. 1. BoostNet Architecture: The high-power base-station uses the same channel as the sensors and can reach the entire field.

There are two main uses of the high-power base-station. It helps synchronize the sleeping patterns of the nodes when there is no activity. The sensors wake up at pre-defined intervals for a short period to listen to the base-station's beacons and synchronize their clocks if they have drifted. Observe that this architecture facilitates time synchronization without requiring any sensors to transmit. The other use of the base-station is in optimizing global parameters for communication protocols. Several parameters are dependent on the traffic pattern resulting from the event which can only be optimized during the occurrence of the event. We explore the dynamic optimization of one such parameter, namely the number of colors used to schedule transmissions in our scheduling approach.

In our approach, time is divided into cycles. Each cycle consists of one DCF (Distributed Coordination Function [7]) slot in which any node can transmit, and k dedicated (colored) slots in which transmissions are allowed only on the respective colored links. Initially, nodes wake up only in the DCF slots to send or receive packets. Upon observing an event, nodes start communicating to coordinate their colors using information piggybacked in the data packets. We use a GPSR [8] like greedy routing protocol for forwarding packets, but the work easily generalizes to other routing protocols as well. The first link is colored with the least utilized color in the neighborhood. The colors in the subsequent links are sequentially assigned till the packet reaches a link that is already colored. All nodes remain awake in the DCF, transmission, and reception slots. The base-station periodically (after several cycles) commands the entire network to change k, to probe its optimal value. Once discovered, the optimal value of k is used for a certain duration before re-initiating the search.

The contributions of the paper are as follows:

- We propose BSMac, a energy conserving scheduling approach that conserves energy during event occurrence and does not require any transmissions by the sensors during periods of inactivity.

- Our proposed BoostNet architecture leverages the power differential of the base-station and the sensor nodes. This architecture can also be used for optimizing parameters of other protocols and for centrally controlling and coordinating other network activities.
- We analytically evalaute the performance of BSMac and show that it conforms to the simulation results.
- We present simulation based comparison of our approach with CSMA. We observe that BSMac reduces energy consumption by up to 80% while improving the throughput and maintaining the latency.

The rest of the paper is organized as follows. Section II motivates the problem and our approach. Section III presents the details of our protocol. Section IV analyzes the throughput in our architecture and compares it to the results from the simulations. In Section V, we present results from simulations comparing our approach to a CSMA based approach. Section VI summarizes related work on this topic. Section VII presents discussion on future extensions to the work. Finally, Section VIII concludes the paper.

II. Motivation

In this section we present the limitations of fine-grained scheduling and the limitations of apriori parameter optimization. To motivate the design of BSMac, we also outline the advantages of both CSMA and TDMA, and our technique of leveraging the power differential between the base-station and the sensors.

Limitations of Fine-grained Scheduling: Ideally, in a sensor network, a scheduling protocol must determine a transmission schedule for each packet that avoids collisions and that conserves energy. Such fine-grained scheduling can be performed centrally or distributedly. For central computation, the overhead of control messages and the delay in scheduling is often prohibitive [4]. Although an out-of-band channel and an extra radio could be used to facilitate the scheduling, the additional cost, and the added complexity of managing the power in the second radio makes it a less attractive alternative. Distributed computation of fine-grain scheduling is also faced with problems of excessive messaging and high latency. In [5], authors designed a local messaging based scheduling approach that may increase latency by a factor exceeding 100. In contrast, sensor networks require low message overhead to conserve energy and increase network lifetime. Also, most sensor network applications such as those based on detection and tracking require low latency. So, fine-grained scheduling is not suited for sensor networks.

In this paper, we explore the design of a coarse scheduling approach that uses local coordination with upstream and downstream nodes only. Limiting coordination only with upstream and downstream nodes allows the coordination information to be piggybacked in the data packets.

Limitations of Apriori Parameter Optimization: Most protocol parameters can be optimized during initial simulations and field tests. However, there is often a set of critical parameters that can not be optimized apriori due to the following four key reasons. First, certain parameters may be a function of the generated traffic pattern, that can not be determined before the occurrence of the event. For example, parameters for a scenario with an event triggering a few sensors may be quite different from an event that triggers a large number of sensors. Second, parameters at different protocol layers often have complex interactions that are hard to predict. Third, the application's prime metric of interest may change dynamically during an event's lifetime. For example, in an event tracking sensor network, the latency may be the primary metric for initial event detection, but throughput may become the primary metric after the initial detection. The critical metric may be application dependent or may even be controlled by a human in the loop. And fourth, changes in the network topology due to node failures, may render apriori optimizations useless.

In this work, we study the optimization of the maximum number of colors used for scheduling links. We have observed that the optimal value of the number of colors depends on factors such as the number of traffic sources, which can not be known apriori. We optimize the parameter by dynamic exploration with multiple settings during occurrence of the event.

Leveraging the Power Differential between Sensors and Base-station: In order to perform parameter tuning during the occurrence of an event, global coordination is required to perform multiple explorations with different parameter settings. As mentioned earlier, a 2-radio solution can be used to coordinate using an out-of-band channel. However, the additional cost of the secondary radio module in each sensor and the added complexity of managing the sleeping cycle of that radio poses questions on the practicality of the 2-radio approach.

In our design, we leverage the power differential by controlling the parameter exploration from the sink using the base-station. However, unlike the complex 2-radio approach outlined above, we assume that the base-station is using the same channel as the sensors. Our design requires infrequent messaging from the base-station as discussed later in Section V.

TDMA versus CSMA: Energy conservation is critical for battery-operated sensor networks. Although CSMA based protocols have gained popularity in WLANs, TDMA based protocols have several advantages in sensor networks. In TDMA, slot assignments can be done in a smart way to allow nodes to sleep for a few packet durations. Such sleeping between packet transmissions can conserve significant energy. Coordinating transmission schedules of nodes to avoid collisions is challenging. As discusses above, for TDMA, a centralized approach requires excessive communication, and a distributed approach incurs significant coordination overhead and often leads to inefficient scheduling.

Our scheduling approach uses the benefits of both TDMA and CSMA. We use globally coordinated time-slots, that are synchronized with the high power transmissions of the base-station. However in each transmission slot, that can handle multiple packet transmissions, nodes use CSMA to compete for the channel in order to avoid colliding with nearby nodes that may be contending in the same slot.

An Example: We show a simple example to illustrate the motivation for the problem and our approach. We consider a linear network topology with 12 hops where adjacent nodes are separated by 200 m. The interested reader may see Section V for other details of the simulation environment. Time is divided into cycles, and each cycle is divided into one DCF and k dedicated colored slots. Figure 2 shows that by choosing the optimal number of colors (four) the throughput can be optimized. More interestingly, we observe that optimizing the throughput also optimizes the normalized energy consumption (Figure 3). Normalized energy is defined as the total energy consumption divided by the number of received packets. In comparison to CSMA, in our architecture, optimization of the number of colors can reduce the energy consumption by 40% while improving the total throughput.

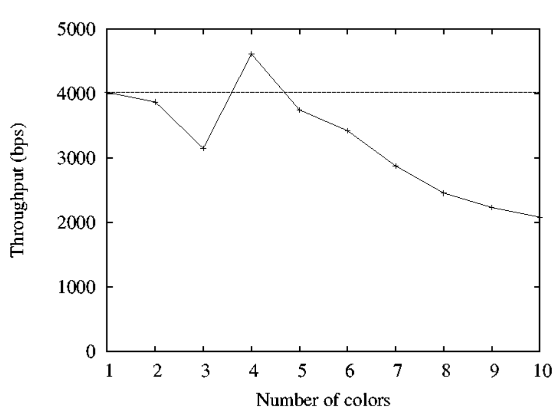

Fig. 2. Throughput for the 12-hop chain topology. (1 color = CSMA)

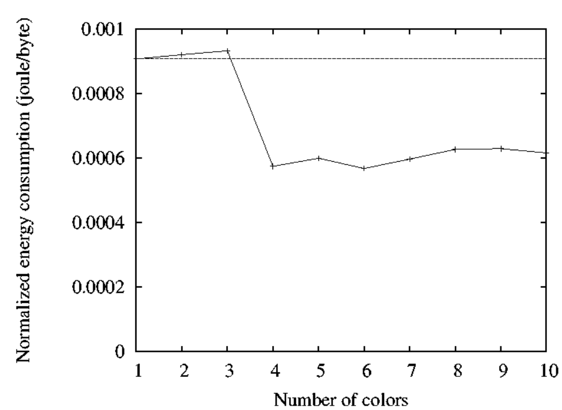

Fig. 3. Normalized energy consumption for the 12-hop chain topology. (1 color = CSMA)

III. Protocol Description

In this section, we present an overview of BSMac followed by its detailed description.

Overview: In BSMac, initially sensor nodes operate at a low duty cycle where they periodically wake up for a fixed duration. In these fixed periods, referred to as DCF (based on the Distributed Coordination Function in 802.11 [7]), nodes can transmit data as they know that other neighbors are awake. Also, the base-station sends a small packet in these fixed periods to synchronize the time if the local clocks have drifted. Once an event is observed, the sensors communicate in the DCF durations and inform the downlink node the color of the additional slot that they will start using. The value of the color is piggybacked in the data packet transmitted in the DCF slots. Each node wakes up in the transmitting colored slot, receiving colored slot (slot dedicated to receiving packets from upstream nodes), and the DCF slots, and sleeps in the remaining slots. We use a simple sequential coloring scheme as it does not require any coordination with nodes other than the downstream node. The color on the first link is chosen as the least used color and the colors on the subsequent links are chosen in a ordered cyclic fashion. After an epoch, which is a fixed number of cycles, the base-station asks the entire network to use a different number of colors. After a few epochs with various numbers of colors the base-station determines its optimal value, and directs the entire network to start using the optimal value. The whole process can be repeated to re-optimize if needed.

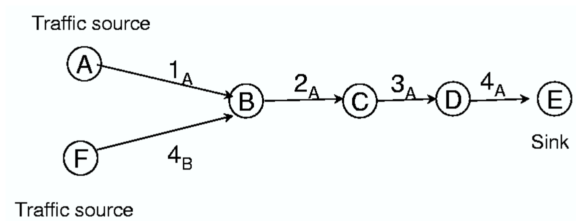

Fig. 4. Color assignment for converging flows. The subscripts represent the corresponding source of the flow.

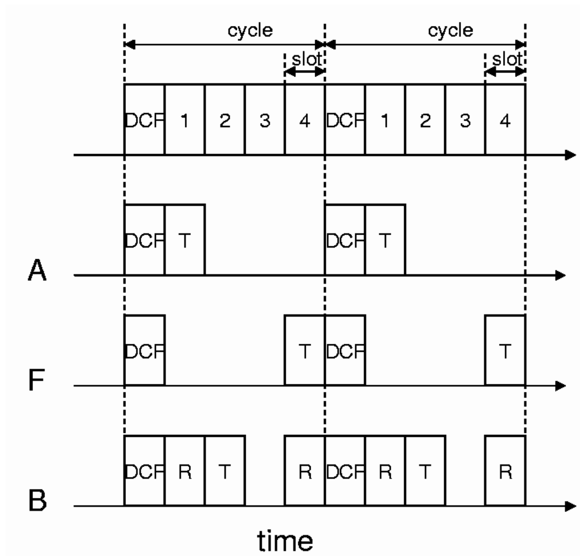

Fig. 5. Slots to wakeup. T is a transmitting slot and R is a receiving slot. All nodes can transmit in the DCF slot.

In BSMac, time is divided into cycles. Each cycle is divided into equal size slots. The default value of number of colors is four. Initially the nodes wakeup only during the DCF periods. Thus initially the duty cycle is 20%. Each slot can accomodate several packet transmissions (our simulations were based on a value of 5).

If an event happens during a non-DCF slot, the event reporting is delayed to the next DCF slot. Since multiple packets can fit in a single DCF slot, the first packet travels a few hops before it encounters a cycle-long delay . The traffic source picks up a color that is least used in its neighborhood. It can be learned by the color information contained in other nodes' packets transmitted in the DCF period. This requires snooping packets in the DCF slot. To optimize the end-to-end delay, and to avoid color coordination messages with all the neighbors, a simple sequential coloring scheme is used. The colors are assigned to links on the route to the sink in an increasing order, repeating the colors from the beginning if needed. Thus, each node learns of the colored slots in which it will receive data and the colored slot in which it will transmit. Nodes wake up in the DCF, transmission, and reception slots, and sleep in the remaining slots.

When two flows merge, the coloring beyond the merging point depends on the first flow that colored it. As the color on the first link of the two flows are chosen independently, there is a chance that the links to the branching point are assigned the same color. Figure 4 shows two merging flows where flow A-B-C-D-E was colored first. Subsequently, traffic source F started coloring with the least used color in its neighborhood which happened to be color 4. Figure 5 shows the periods in which nodes A, F, and B remain awake.

If a source has no more traffic to send, the slot assignment along the flow path from the source to the sink will timeout. Thus nodes can adjust their sleep schedules according to the change of traffic patterns, which is important for energy conservation in the case of moving events. If some other flows converge with this one later, then after the slot assignment timeout for the first flow, other flows can color the common path according to some specific criteria. In our simulations, the nodes at the converging points will select the slot assignment of the most heavily loaded upstream node to re-color the downstream common path.

At the outset, each sensor node operates using default number of colors (k). However, for a new event the default k may be suboptimal. The base-station evaluates the throughput (other metrics could also be used) by experimenting with a value of k for one epoch, which consists of several cycles. In simulations, we have observed that 10 sec is enough for evaluating an assignment of the parameter. A small epoch may not be sufficient to evaluate the performance for that assignment of the parameter, and a large epoch will require long time to converge to an optimal value. In different experiments depending on the number of active sources, we have observed that the number of optimal colors could be anywhere between 2 and 5. BSMac experiments with these four values of k before converging to the optimal value.

IV. Analysis

In this section, we analytically evaluate the throughput of our scheduling protocol for a linear network topology. Our analysis is based on Bianchi's seminal work [9] on performance analysis of WLANs. Although in this section we focus only on a linear topology, we believe that the methodology presented here can be used to analyze more complex topologies. We assume that the interference range is twice the transmission range. As shown in Figure 6, when the number of colors is 2, the number of contending nodes in the same scheduled slot is 3. According to [9], the probability that a node transmits a packet in a random time slot is given by:

$$\tau = \frac{2(1-2p)}{(1-2p)(W+1)+pW(1-(2p)^m)} \tag{1}$$

where p is the conditional collision probability that a node's packet collides with packets from other nodes, W is the minimum contention window, and m is the number of backoff stages. In steady state the conditional collision probability p is given by

$$p = 1-(1-\tau)^{n-1}+(1-\tau)^{n-1}\times P_{hc} \tag{2}$$

where n is the number of contending nodes, and P_{hc} is the conditional collision probability that is caused by hidden terminals. P_{hc} is 0 if there are no hidden terminals. If there is a hidden terminal, as Figure 7 shows, the collision will happen if the hidden terminal transmits a packet during the sender's transmission. To simplify the analysis, we assume that the collision probability caused by the hidden terminal is low, and we ignore its effect. By solving Equations 1 and 2, we can get the transmission probability τ.

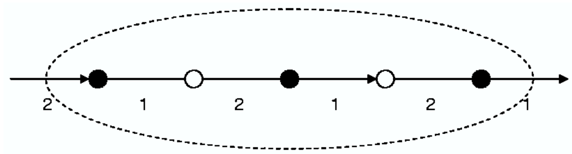

Fig. 6. The number of nodes in interference range.

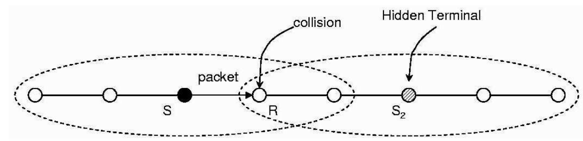

Fig. 7. The node that causes the hidden terminal problem.

In order to calculate the theoretical throughput, we need to know how much time is spent in successful transmissions. There are four possible channel conditions: idle, successful transmission, collision, and busy (other nodes occupying the channel). The probabilities of each channel condition that a node senses are P_i, P_s, P_c and P_b respectively, and $P_i + P_s + P_c + P_b = 1$. We can derive these probability using the transmission probability τ:

- P_i is the probability that no node is transmitting, therefore $P_i = (1-\tau)^n$.
- P_s is the probability that the sender is transmitting, and others are not. Therefore $P_s = \tau(1-p)$.
- P_c is the probability that the sender is transmitting, and at least one of its neighbors is transmitting, too. $P_c = \tau p$.
- P_b is the probability that the sender is not transmitting, and at least one of the sender's neighbors is transmitting. Therefore $P_b = (1-\tau)(1-(1-\tau)^{n-1})$.

Table IV lists these probabilities for different number of color assignments.

k	n	τ	P_i	P_s	P_c	P_b
1	5	0.048953	0.778055	0.040048	0.008904	0.172992
2	3	0.055192	0.843394	0.049268	0.005924	0.101414
3	1	0.062500	0.937500	0.031250	0.031250	0.000000
4	1	0.062500	0.937500	0.062500	0.000000	0.000000
5	1	0.062500	0.937500	0.062500	0.000000	0.000000
6	1	0.062500	0.937500	0.062500	0.000000	0.000000
7	1	0.062500	0.937500	0.062500	0.000000	0.000000
8	1	0.062500	0.937500	0.062500	0.000000	0.000000
9	1	0.062500	0.937500	0.062500	0.000000	0.000000
10	1	0.062500	0.937500	0.062500	0.000000	0.000000

TABLE I

PROBABILITIES FOR DIFFERENT NUMBER OF COLOR ASSIGNMENT. ($k = 1$ MEANS CSMA/CA)

The throughput S can then be computed as the the number of bytes transferred by a node per unit time. S is the size of a packet transmitted successfully, divided by the sum of time spent in idle, successful transmission, collision, and busy periods. Using the above probabilities, along with the packet transmission time, collision time and idle time, we can derive the throughput as:

$$S = \frac{S_p}{T_I + T_S + T_B + T_C} \times \frac{1}{k} \tag{3}$$

where S_p is the packet size, T_I is the time a node senses the channel as idle, T_S is the time a node used for a successful transmission, T_B is the time the channel is busy, T_C is the time a node's transmission causes a collision, and k is the number of scheduled slots ($k = 1$ means CSMA/CA). $S_p = P_s \times E[P]$, where $E[P]$ is the expected packet size. $T_I = P_i \times \sigma$ where σ is the length of a time slot . $T_S = P_s \times T_s$ where T_s is the time to successfully transmit a packet, which includes the time to transmit a packet, an acknowledgement, DIFS, SIFS, and propagation delay. $T_C = P_c \times T_s$ because the sender will not know of the collision until the transmission timeout, and the length of the timeout is T_s. Therefore, we use T_s as the time for collision when the sender is transmitting.

The throughput $S = \frac{S_p}{T_I+T_S+T_B+T_C}$ is the theoretical throughput the network can achieve with the specific number of contending nodes when all nodes can transmit 100%. Therefore the actual throughput must be divided by the number of colors because nodes can only transmit in its scheduled slot.

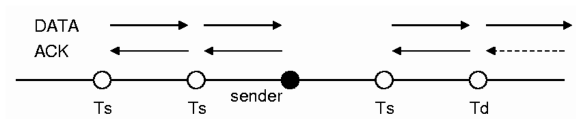

Fig. 8. The time a node senses the channel as busy while other nodes' transmissions are successful. The sender only senses the packet transmission time from the right node. It will not hear the acknowledgement packet.

Figure 8 shows the time the node senses the channel as busy for successful transmissions from other nodes. Transmission from different nodes have different channel occupation time in the node's point of view. Furthermore, other nodes' transmission may also collide with each other, which may have different channel occupation time in different situations. To simplify the analysis, we assume that a collision happens whenever two neighbor nodes are transmitting concurrently. Therefore, T_B can be calculated as follow:

- For 1 color assignment, the T_B is:

$$3 \times P_s \times T_s + P_s \times T_d + (P_b - 4 \times P_s) \times T_c \tag{4}$$

  where T_d is the time to transmit a data packet and T_c is the time the node sensing the channel as busy in collision caused by other nodes. We use the time to transmit a packet, the DIFS and propagation delay as both T_d and T_c.
- In the 2 color assignment scenario, concurrent neighbor nodes' transmission won't collide with each other, therefore, T_B in 2 color assignment scenario is:

$$\tau \times (1-\tau) \times T_s + (P_b - \tau \times (1-\tau)) \times T_d \tag{5}$$

- However, the analysis is not suitable for the scenario when the number of colors is three. When the number of colors is 3, the node that are 3-hop away will transmit concurrently. Consider that the number of slots for transmitting a data packet, T_d/τ, is usually hundreds of slots. The initial backoff stage, however, is only 31 in our simulation. Therefore a node's transmission will always collide with packet from the downstream node 3-hop away. As long as there is packet in the downstream node, the upstream node will not be able to transmit successfully. Therefore, not all nodes have packets to send in 3 color scenario, and the above model can not be used.
  Since a node can not send a packet if its 3-hop downstream node has packets to send, we can consider that these two nodes that are 3-hop away share the channel. Only one node can transmit packets at the same time, and the downstream node has higher priority. Therefore, the throughput is one half of the original throughput without the hidden terminal problem.

Using the above equations, we can compute the theoretical throughput by replacing the parameters we used for simulation. In our simulation, the value of $W = 31$, $m = 7$, the packet size (including the packet header) is 62 bytes, the ACK packet size is 40 bytes, the time of a slot is $20\ \mu s$, and the channel bandwidth is $38.4\ Kbps$. Figure 9 shows the result of the analytical data for different number of color assignment.

Figure 9 shows the throughput of analytical and simulation results. We can see that when the number of color is 1, i.e. no scheduled time slot and all nodes are contending with each other (CSMA), the

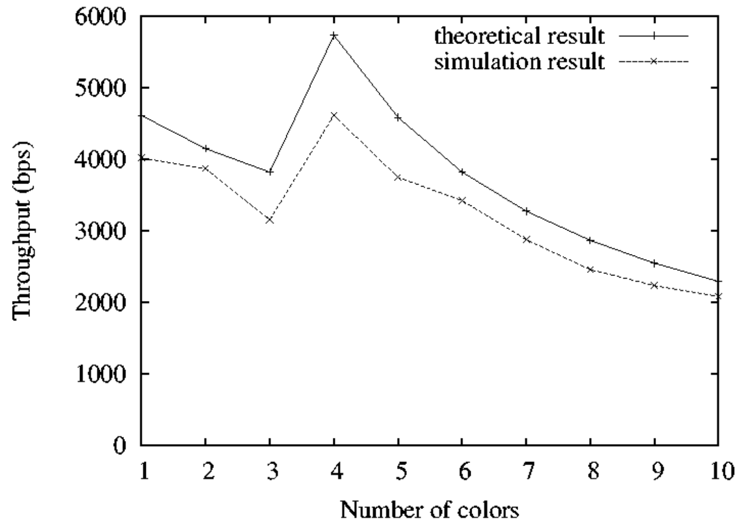

Fig. 9. Analytical throughput for different number of color assignment. (1 color = CSMA)

throughput is not optimal. When the number of colors is 4, the throughput is maximized because there is no contending node. When the number of colors is greater than 4, no node contends the channel with others. As the network throughput is divided by the number of scheduled slots, the throughput decreases gradually when the number of scheduled slots (or colors) increases beyond 4.

We observe that the throughput observed in simulation is slightly lower than the analytic throughput. In our analysis we have assumed that all nodes are always backlogged. However, due to the contention in the first few hops, the subsequent hops may not always have data to send. In addition, our analysis assumes that each cycle consists of only colored slots but the DCF slot, while the DCF slot takes $\frac{1}{k}$ of the time in BSMac protocol. Thus we observe lower throughput in our simulations. But our analysis has accurately captured the veriation in throughput with number of colors.

V. Performance Evaluation

In this section, we evaluate the performance of BSMac in various scenarios. The three main performance metrics are throughput, latency and normalized energy consumption. Normalized Energy is defined as the energy consumed to deliver one byte to the sink. As energy consumption is critical in sensor nodes, this metric is of utmost importance.

The simulations are conducted in ns2 [10]. Our approach is implemented over the IEEE 802.11 code of ns2, where the corresponding backoff timer is paused when the transmission slot or DCF slot ends, and is resumed when one of these two slots occurs again. In addition, nodes are put to sleep when they are not in their transmission, reception or DCF slots. The choice of the colored transmission slot of each node is carried in the packet header, and it introduces 2 bits of extra overhead compared to IEEE 802.11. However, since IEEE 802.11 is designed for WLANs (Wireless LANs), some fields are redundant for wireless sensor networks, which means these 2-bit overhead can use the reserved fields in the MAC header.

The transmission range and interference range are left at the default values of 250 m and 550 m respectively, which are similar to the best results of the Mica2 [11] radio. But our approach does not rely on the specific values. Thus it can be easily applied to wireless sensor networks with smaller or larger transmission interference ranges.

For BSMac, the length of each time slot is set to 110 ms, which can accommodate the transmission of 5 packets, and the default number of colored slots is set to four. The other significant parameters are summarized in Table II.

TABLE II
SIMULATION SETTINGS

Parameter	Value	Parameter	Value
DATA packet size	62 bytes	ACK packet size	40 bytes
IEEE 802.11 DATA size	60 bytes	IEEE 802.11 ACK size	38 bytes
Transmission range	250m	Carrier sense range	550m
Bandwidth	38.4 Kbps	Data rate	50 packets/sec.
Transmission Power	0.075W	Receiving/idle Power	0.025W
Interface Queue length	50	Routing protocol	Greedy routing
Slot time length	110ms	Default number of slots	4

To validate the BSMac protocol, we conduct performance measurement under three network topology configurations: the simple chain topology, a randomly generated large topology with multiple simultaneous traffic sources and a grid topology with one moving event. We compare the performance of BSMac with IEEE 802.11 (without RTS/CTS). The highlights of our simulation based evaluation is as follows:

- Our approach saves the normalized energy consumption by 40-45% and improves throughput by about 20% for the chain topology.
- BSMac saves 50-80% normalized energy, while achieving about 80% throughput of IEEE 802.11 DCF mode in case of multiple events. For small number of events (one or two), BSMac obtains higher throughput that 802.11. In addition, latency in BSMac is also comparable to that of IEEE 802.11.
- For the scenario with one moving event, BSMac saves normalized energy up to 60%, and can achieve more than 80% throughput of CSMA/CA when the moving speed is less than 10m/s, and 75% when the speed is between 15m/s and 20m/s.
- We observe that a probing interval (epoch) of 10 seconds suffices for determining the best value of a parameter.

A. Chain Topology

In this section, we considered a linear topology of 13 nodes. Simulations are run for 1000 second to obtain a stable performance measurement. In addition, we also study the impact of different node densities.

From Figure 10 we can see that after finding the optimal parameter of number of slots, the throughput can be improved by about 20%, and the latency is only 30% higher than CSMA/CA protocols (Figure 11) although packet transmission is confined to specific time slots. As for the energy consumption, the BSMac protocol consumes less than 60% of CSMA/CA (Figure 12), which can be anticipated since nodes sleep in some slots. Taking optimal number of slots for 200 m hop distance as an example, the optimal number is 4, thus a cycle is composed of 5 slots, and each node is awake in the DCF, one transmission slot and one reception slot, which leads to about $\frac{3}{5}$ energy consumption compared to CSMA/CA. In addition, since the optimal number of slots alleviate the contention, the BSMac can use less than 60% energy to deliver more packets.

Actually, in the chain topology, each node participates in the packet transmission and reception, which is not the case for a large sensor networks where only nodes along the path from the source to the sink will wake up in their transmission and reception slots, while other nodes will only wake up in the DCF slots. Thus we anticipate more energy saving in Section V-B.

B. Randomly Generated Network with Multiple Simultaneous Events

To test the performance of BSMac in a large deployed network, we generate a network topology by placing 400 nodes uniformly randomly in a $2000m \times 2000m$ area, and randomly select some places to trigger events. One to ten flows are simulated to show the efficiency of the BSMac protocol.

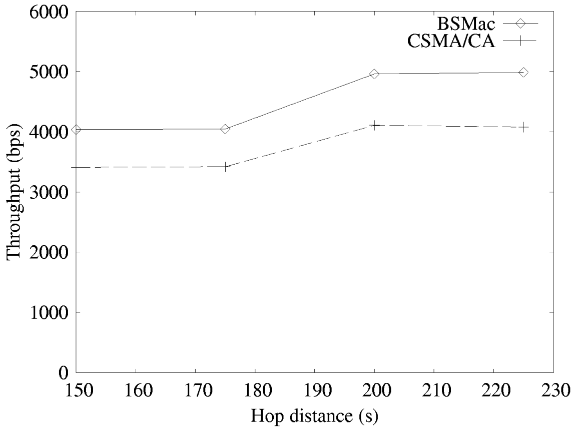

Fig. 10. Throughput for the 12-hop chain topology in 1000s

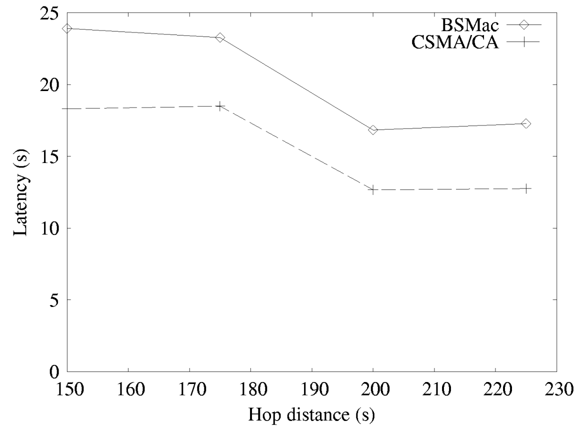

Fig. 11. Latency for the 12-hop chain topology in 1000s

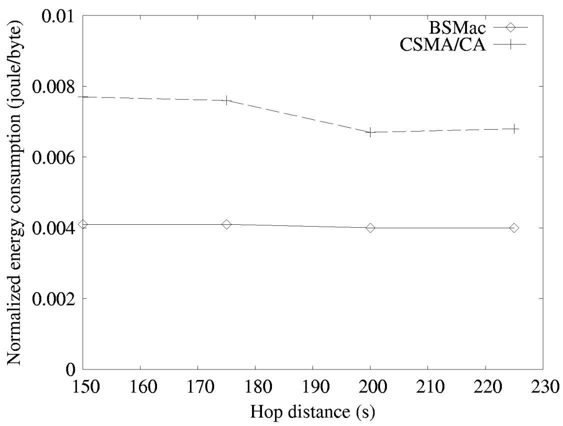

Fig. 12. Normalized energy consumption for the 12-hop chain topology in 1000s

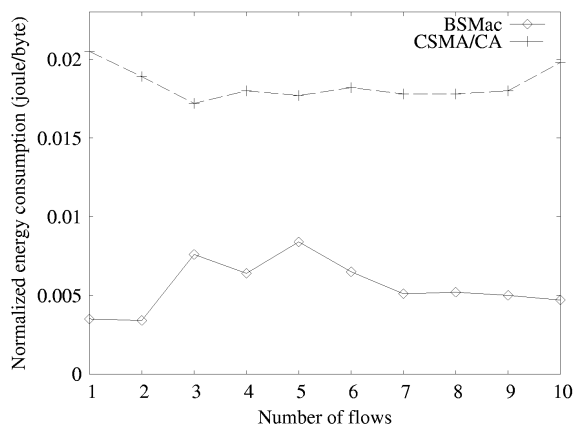

Fig. 13. Normalized energy consumption for the randomly generated topology with multiple events in 1000s

All simulations are run for 1000 seconds to achieve stable results. Figure 13 exhibits the normalized energy consumption of the BSMac and CSMA/CA. We observe that when there are only 1 or 2 flows, the normalized energy consumption is reduced by more than 82%. Even in the case of more than 2 flows, the energy consumption per byte is in the range of 20-45% compared to CSMA/CA. Figure 14 summarizes the throughput of BSMac together with that of IEEE 802.11 DCF node. It can be seen that if the number of flows is very small (1 or 2), the BSMac protocol improves the throughput by 5-10%. If the number of flows is more than 2, the BSMac protocol can achieve more than 85% throughput of CSMA/CA. Figure 15 shows that the latency of BSMac with slotted time is comparable with the that of CSMA/CA. The three metrics together validate that the BSMac protocol is suitable for the energy-constrained wireless sensor networks.

C. Grid Network with One Moving Event

In this section, we constructed a 10x10 grid network in a $2000m \times 2000m$ area. We study the performance of BSMac together with CSMA/CA under different moving speeds. Figure 16 shows that BSMac impairs the throughput by less than 20% when the moving speed is slower than 10 m/s. Although the throughput degradations are about 25% when the moving speed is increased to 20 m/s, we observe from Figure 17 that BSMac saves 60% normalized energy. In addition, Figure 18 shows that BSMac has almost the same latency performance as CSMA/CA. Therefore, BSMac may not be better for monitoring fast moving events with high data rate, it still exhibits its significant advantage in energy savings. For such dynamic

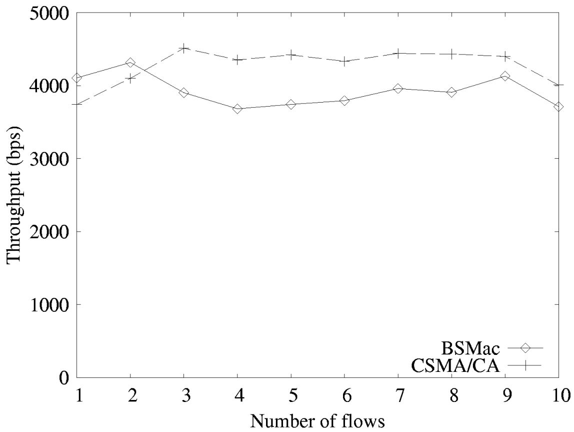

Fig. 14. Throughput for the randomly generated topology with multiple events in 1000s

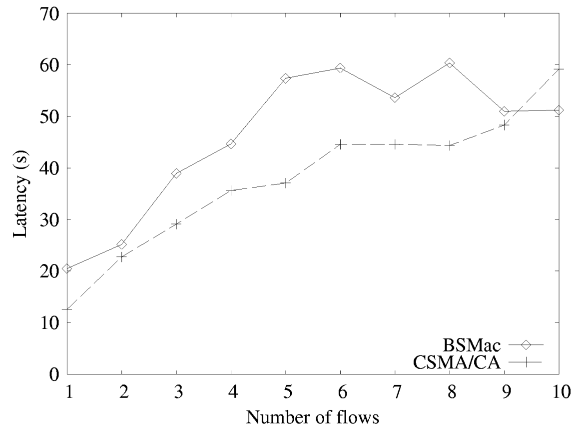

Fig. 15. Latency for the randomly generated topology with multiple events in 1000s

scenarios, it may be better to wakeup all nodes on the route with a 100% duty cycle. We plan to explore this limitation of our protocol further.

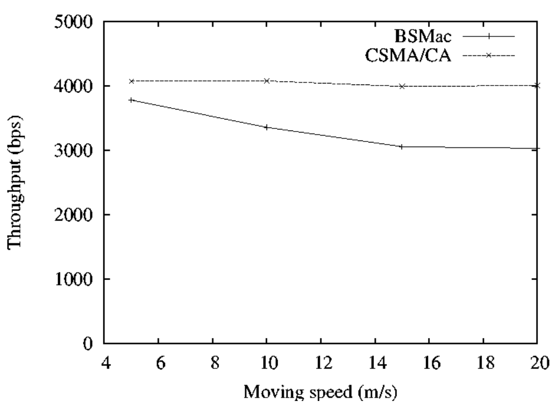

Fig. 16. Throughput for the scenario of moving event in 1000s

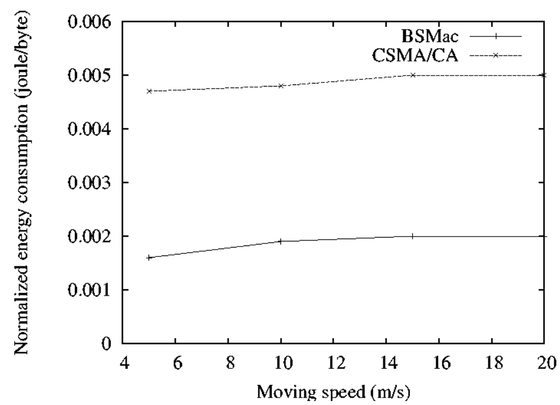

Fig. 17. Normalized energy consumption for the scenario of moving event in 1000s

D. Parameter Probing Interval

To probe the optimal parameter, namely the number of colors for the network, parameter probing messages are broadcast by the sink to the whole network. To quickly get the optimal value, the probing interval should be small. But with a very small interval, it is hard to gather sufficient evidence to discard one value or adapting to another. In addition, frequent probing will surely impair the throughput of data traffic because of the asymmetric links between each sensor node and the sink.

To find an appropriate value for the probing interval (also referred to as epoch interval), we conduct a simple simulation with the chain topology used in Section II. The throughput observed by the sink is shown in Figure 19. From the figure we can see that after 10 s, using 4 time slots leads to the best performance, and the throughput observed by the sink is stable. In our simulations, we have cautiously used a larger probing interval of 20 s for stability. It also means less overhead and collisions with on-going data-traffic since the link between a common sensor node and the sink is asymmetric. We have given a higher priority to the broadcasts from the base-station in order to reduce collisions with ongoing data transmissions in the DCF slot. The higher priority has been implemented as a lower value of DIFS and no backoff if the channel is found to be available. Although the probing interval, can itself be optimized by utilizing the BoostNet architecture, it has not been studied in this paper.

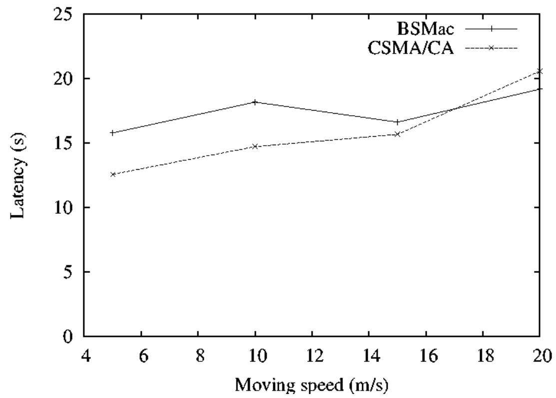

Fig. 18. Latency for the the scenario of moving event in 1000s

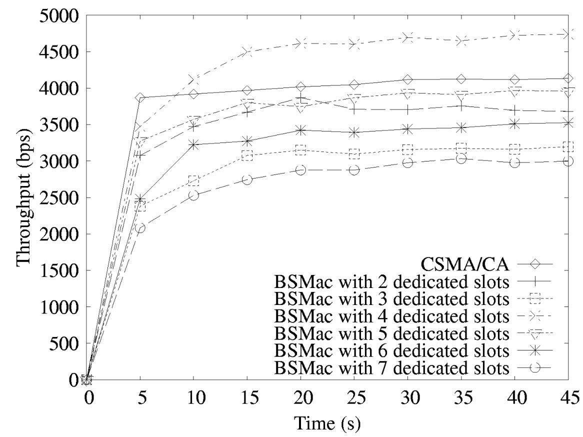

Fig. 19. Throughput observed by the sink vs. runtime

VI. Related Work

Much work has been conducted on the topic of energy conservation in wireless sensor networks, PAMAS [12] uses out-of-channel signaling to avoid over-hearing among neighbor nodes. But adding the second interface to the sensor may not be cost-efficient and may lead to complexity of power management on the second channel.

Woo and Culler [13] propose an adaptive rate control mechanism to achieve the fairness of bandwidth allocation while being energy-efficient for both low and high duty cycles. S-MAC [14] synchronizes neighbor nodes' sleeping schedule to make nodes work at low duty cycle. These protocols deal with the energy conservation based on the CSMA/CA mechanism, and do not utilize the weapon of scheduling to put nodes to short sleep when they have packets to send.

Appropriate link scheduling mechanism can facilitate energy savings in wireless sensor networks while reducing the collision rate. Central coordination mechanisms suffer from the inefficiency of information gathering at fine-grained packet level. In [4], the authors propose three heuristic link coloring and scheduling schemes to solve the optimal parallel communication scheduling problem. They compare the three heuristic color selection policies: minimal weight color heuristic, random color selection heuristic, and least used color heuristic. However this work is based on the assumption that there is a central coordinator which knows the exact topology, interference range, and all traffic information at per-packet level, which is not practical.

Researchers have also proposed distributed approaches to energy conserving scheduling protocols in ad hoc networks [2][3][15] and sensor networks [5][16]. In [3], authors use the concept of cliques to represent the conditions of collision-free transmission. A backoff adjustment mechanism is devised according to the general framework to ensure proportional fairness in wireless shared channel. But the performance analysis in [3] considered only some special topologies, and the backoff adjustment based approach can not put nodes to sleep because they have to observe the transmission failure ratio to adjust the backoff. [2] analyzes the contention problem in the same manner, but its purpose is to achieve maximum allocation of the shared wireless channel while assuring minimum throughput and delay bounds for each flow. Since the approach in [2] is also backoff-based, it fails to fulfill the design goal of energy-saving for wireless sensor networks.

NAMA [15] also divides the time into cycles, where each cycle consists of several hundred slots. Nodes communicate with their 2-hop neighbor nodes to coordinate their slot selection. The authors design a hash function and an exquisite local negotiation mechanism to decide which node has the highest priority to transmit. TRAMA [5] is actually an extension of NAMA. In TRAMA, nodes will sleep in slots in which they do not transmit or receive packets. Unfortunately, the assumption of collision-free transmission in 2 hops is not appropriate since such a number is different for different node densities. In addition, the latency of TRAMA may exceed 100 times the latency of CSMA/CA, which may not be desirable if the latency performance is important.

In [16], the authors propose another TDMA-based MAC scheduling protocol. There are two kinds of slots, one is for transmission and the other is for reception, and nodes negotiate their slot selection locally. This approach is similar to our BSMac protocol. However, this work is only suitable for low data rate networks and thus throughput is not a consideration. Therefore it can not provide high throughput compared to our BSMac protocol.

VII. Future work and Discussions

In this section we discuss some ongoing research issues and future work, that have not been addressed in this paper.

- **Towards a Scalable Design:** By using long antenna and high power, the base-station's range can be made much larger than the sensors range. However, for very large deployments a more scalable approach is needed. We are investigating a hierarchical design with multiple base-stations scattered across the network in such a way that they cover the entire sensor field. These high power base-stations also need an out-of-band channel using which they communicate and coordinate with the base-station. Through coordination, collisions between transmissions of base-stations can be eliminated. As the set of all base-stations cover the entire sensor field, the parameter probing can proceed in a manner similar to that outlined in this paper. This scalable architecture is also suited for fault tolerance that is currently lacking in the single base-station approach.
- **A Multi-channel Alternative:** In Section II we discussed an alternate architecture requiring an additional receiver radio module in each sensor node. Although we did not consider that architecture in this paper, it has some advantages over the proposed architecture. Using a frequency that is more suited for long range such as FM or AM, we can create very large sensor networks by using only a single base-station. However, as mentioned before, the energy consumption on that radio must be optimized as well with smart coordination between the two channels.
- **Leveraging BoostNet for Optimizing Parameters in other Protocols:** We proposed a simple sequential scheduling approach that does not require coordination with nodes other than the immediate upstream and downstream nodes. However, other scheduling algorithms can also leverage the architecture to optimize their performance. In fact, other network protocols can also leverage the BoostNet architecture to statically or dynamically optimize parameters and improve performance.

VIII. Conclusion

This paper proposes an energy conserving scheduling protocol, called BSMac. It uses a new architecture called BoostNet, to dynamically adjust the network parameters and improve performance. BSMac divides time into cycles, where each cycle consists of a DCF slot and k scheduled colored slots. Nodes can only transmit packets in DCF or their scheduled transmission slots. During periods with no activity, nodes wakeup only during DCF slots. During occurrence of events, nodes wakeup during transmission, reception, and DCF slots, thus conserving energy. The BoostNet architecture allows the base station with unlimited power supply to command all sensors to change the number of slots k and to probe for its optimal value. By dynamically adjusting the parameter, BSMac can improve the throughput and maintain low latency while saving up to 80% energy compared to traditional CSMA protocols. Using BoostNet has two advantages. First the base station can send messages to synchronize sensors obviating the need for the sensors to transmit any messages for time synchronization. The second advantage is that the sink can measure the runtime network performance, and the attached base-station can update the network parameters to quickly adapt to different traffic patterns. Based on the simulation results, we conclude that BSMac is highly suited for networks that require high throughput, low latency, and long network lifetime. Our analytical modeling of throughput has closely predicted the observed throughput in the simulations.

REFERENCES

[1] "Envisense," http://envisense.org/glacsweb.htm.

[2] Haiyun Luo, Songwu Lu, Vaduvur Bharghavan, Jerry Cheng, and Gary Zhong, "A Packet Scheduling Approach to QoS Support in Multihop Wireless Networks," *Mobile Networks and Applications*, vol. 9, no. 3, pp. 193–206, June 2004.

[3] Thyagarajan Nandagopal, Tae-Eun Kim, Xia Gao, and Vaduvur Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," in *Proceedings of ACM MOBICOM 2000*, August 2000, pp. 87–98.

[4] Huan Li, Prashant Shenoy, and Krithi Ramamritham, "Scheduling Communication in Real-Time Sensor Applications," in *Proceedings of IEEE RTAS 2004*, May 2004, pp. 10–18.

[5] Venkatesh Rajendran, Katia Obraczka, and J.J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks," in *Proceedings of ACM SENSYS 2004*, November 2003, pp. 181–192.

[6] H.-S Kim, T. A. Abdelzaher, and W.-H. Kwon, "Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks," in *Proceedings of ACM SENSYS 2004*, November 2003.

[7] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," ISO/IEC 802-11:1999, 1999.

[8] Brad Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," in *Proceedings of ACM MOBICOM 2000*, August 2000, pp. 243–254.

[9] Guiseppe Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, March 2000.

[10] "The Network Simulator – ns-2," http://www.isi.edu/nsnam/ns/.

[11] "Mica2," http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/6020-0042-06_B_MICA2.pdf.

[12] Suresh Singh and C. S. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad hoc networks," *ACM SIGCOMM Computer Communication Review*, vol. 28, no. 3, pp. 5–26, July 1998.

[13] Alec Woo and David E. Culler, "A transmission control scheme for media access in sensor networks," in *Proceedings of ACM MOBICOM 2001*, 2001, pp. 221–235.

[14] Wei Ye, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, June 2004.

[15] L. Bao and J. J. Garcia-Luna-Aceves, "A new approach to channel access scheduling for ad hoc networks," in *Proceedings of ACM MOBICOM 2001*, 2001, pp. 210–221.

[16] Zhihui Chen and Ashfaq Khokhar, "TDMA Based Energy Efficient MAC Protocols for Wireless Sensor Networks," in *Proceedings of IEEE SECON 2004*, October 2004.