Topology Management for Sensor Networks: Exploiting Latency and Density

Curt Schurgers Vlasios Tsiatsis Saurabh Ganeriwal Mani Srivastava ‡

Networked and Embedded Systems Lab (NESL), Electrical Engineering Department, UCLA
56-125B Eng. IV, UCLA -EE Dept., Los Angeles, CA 90095
{curts, tsiatsis, saurabh, mbs}@ee.ucla.edu

‡ *Please send all correspondence to Mani Srivastava*

ABSTRACT

In wireless sensor networks, energy efficiency is crucial to achieve satisfactory network lifetime. In order to reduce the energy consumption of a node significantly, its radio needs to be turned off. Yet, some nodes have to participate in multi-hop packet forwarding. We tackle this issue by exploiting two degrees of freedom in topology management: the path setup latency and the network density. First, we propose a new technique called Sparse Topology and Energy Management (STEM), which aggressively puts nodes to sleep. It provides a method to wake up nodes only when they need to forward data, where latency is traded off for energy savings. Second, STEM integrates efficiently with existing approaches that leverage the fact that nearby nodes can be equivalent for traffic forwarding. In this case, an increased network density results in more energy savings. We analyze a hybrid scheme, which takes advantage of both setup latency and network density to increase the nodes' lifetime. Our results show improvements of nearly two orders of magnitude compared to sensor networks without topology management.

Keywords : Sensor networks, energy efficiency, topology management.

1. INTRODUCTION

1.1. Sensor Networks

Advances in microelectronic fabrication have allowed the integration of sensing, processing and wireless communication capabilities into low-cost and small form-factor embedded systems called sensor nodes [1][2]. The need for unobtrusive and remote monitoring is the main motivation for deploying a sensing and communication network (sensor network) consisting of a large number of these battery-powered nodes. For example, such systems could be used either outdoors in inhospitable habitats, disaster areas, or indoors for intrusion detection or equipment monitoring. The nodes gather various sensor readings, process them and forward the processed information to a user or, in general a data sink. This forwarding typically occurs via other nodes using a flat or clustered multi-hop path [3][9]. Thus a node in the network essentially performs two different tasks: (1) sensing its environment and processing the information and, (2) forwarding traffic as an intermediate relay in the multi-hop path.

However, the convenience of autonomous remote monitoring comes at a price: an extreme design focus must be placed on energy efficiency as the sensor nodes operate on a small battery with limited capacity [1][2][3]. It is important to view the problem as one of extending the lifetime of the network, rather than just that of the individual nodes. Thus, in addition to improving the efficiency of the nodes, techniques that tackle the problem on the level of the entire network are necessary. This is especially true for the traffic forwarding functionality of the network, as the main energy consumer in a node is the communication subsystem [1][3][4]. Our paper explores this category of network-wide techniques, more specifically dealing with topology management.

1.2. Topology Management

Topology management is an important issue because the only way to save power consumption in the communication subsystem is to completely turn of the node's radio, as the idle mode is almost as power hungry as the transmit mode [4]. However, as soon as a node powers down its radio, it is essentially disconnected from the rest of the network topology and therefore can no longer perform packet relaying. For simplicity, we refer to this state as the node being asleep, although only its radio is turned off. The sensors and processor can still be active, as they are much less power hungry.

The goal of topology management is to coordinate the sleep transitions of all the nodes, while ensuring that data can be forwarded efficiently to the data sink. Existing topology management schemes, such as the ones described in references [5] and [6], are based on the observation that in typical scenarios, some nodes can be asleep without sacrificing significant data forwarding capacity. As density increases, more nodes can be sleeping, resulting in further energy savings. However, major savings would require extremely dense networks, as we will illustrate in this paper.

We propose a different approach to topology management, which exploits the time dimension rather than the density dimension. Strictly speaking, nodes only need to be awake when there is data to forward. We refer to this situation as the network being in the ***'transfer state'***, and in many practical scenarios, this is a rather infrequent event. Most of the time, the sensor network is only monitoring its environment, waiting for an event to happen, and nodes can be asleep. For a large subset of sensor net applications, no data needs to be forwarded to the data sink in this ***'monitoring state'***. Consider for example a sensor network that is designed to detect brush fires. It has to remain operational for months or years, while only sensing if a fire has started. Once a fire is detected, this information should be forwarded to the user quickly. Even when we want to track how the fire spreads, it probably suffices for the network to remain up only for an additional week or so. Similar observations hold for applications such as surveillance of battlefields, machine failures, room occupancy, or other reactive scenarios, where the user needs to be informed once a condition is satisfied.

In the monitoring state, no communication capacity is needed, in principle at least. As there is no data to forward, the communication energy could be completely eliminated, by simply turning off the radios of all nodes. If the need for data forwarding is very rare, the energy savings could be phenomenal. However, there is a crucial caveat: if a node detects an event, it cannot forward the data to the user since all the nodes on the multi-hop path are asleep. If a node has turned off its radio, it will stay completely oblivious of the efforts of other nodes to communicate with it. This is the main dilemma in topology management for sensor nets: a node's radio should be turned off to save energy, yet be left on so the node can know when other nodes need it to forward their traffic. Our topology management scheme, called ***STEM (Sparse Topology and Energy Management)***, solves this issue and trades off energy consumption versus latency of switching back to the transfer state.

Furthermore, we would like to develop a topology management scheme that marries the benefits of both classes discussed previously, namely those that exploit network density and those that exploit setup latency. Ideally, this hybrid solution combines the savings in both dimensions fully, such that a ten-fold energy reduction in both schemes separately would result in a combined hundred-fold reduction. This basically requires these base schemes to be orthogonal in using the independent dimensions of latency and density. We propose such a very effective hybrid scheme in this paper, by combining STEM with techniques that leverage the network density.

2. RELATED WORK

For routing in sensor networks, two alternative approaches have been considered: flat multi-hop and clustering. Although STEM is applicable to both of them, we mainly focus on flat multi-hop routing [3]0[8]. For clustered approaches [9], which are possibly hierarchical, our scheme can be used to reduce the energy of the cluster heads, although the gains are expected to be less dramatic here.

Recently, topology management techniques, called SPAN [5] and GAF [6], have been proposed for flat multi-hop routing. They operate on the assumption that the network capacity needs to be preserved. As a result, the energy consumption is approximately the same whether the network is in the transfer or monitoring state, as no distinction is made between them. Both techniques trade off network density for energy savings. The performance of STEM is independent of network density. It operates in an orthogonal dimension, that of setup latency. Our hybrid scheme, which we describe in section 6, leverages both network density and latency.

With SPAN [5], a limited set of nodes forms a multi-hop forwarding backbone that tries to preserve the original capacity of the underlying ad-hoc network. Other nodes transition to sleep states more frequently, as they no longer carry the burden of forwarding data of other nodes. To balance out energy consumption, the backbone functionality is rotated between nodes, and as such, there is a strong interaction with the routing layer.

Geographic Adaptive Fidelity (GAF) [6] exploits the fact that nearby nodes can perfectly and transparently replace each other in the routing topology. The sensor network is subdivided into small grids, such that nodes in the same grid are equivalent from a routing perspective. At each point in time, only one node in each grid is active, while the others are in the energy-saving sleep mode. Substantial energy gains are, however, only achieved in very dense networks. We will discuss this issue further on in this paper, when we integrate STEM with GAF.

An approach that is closely related to STEM is the use of a separate paging channel to wake up nodes that have turned off their main radio [10]. However, the paging channel radio cannot be put in the sleep mode for obvious reasons. This approach thus critically assumes that the paging radio is much lower power than the one used for regular data communications. It is yet unclear if such radio can be designed. STEM basically emulates the behavior of a paging channel, by having a radio with a low duty cycle radio, instead of a radio with low power consumption.

The work of McGlynn *et al* [14] describes an algorithm that resembles STEM. However, it is designed to discover the neighbors of all the nodes some time after the network deployment. The goal is to let the network be dormant during deployment, and once the discovery phase starts, learn the complete topology with a high probability. In principle, this algorithm could also be used to set up a path like STEM. However, it is less aggressive, and would result in much larger setup latency, as a node only sends out setup request probabilistically. Furthermore, it does not guarantee discovery of a link.

3. SPARSE TOPOLOGY MANAGEMENT

3.1. Basic Concept

In the application scenarios we consider in this paper, the sensor network is in the monitoring state the vast majority of its lifetime. Ideally, we would like to only turn on the sensors and some preprocessing circuitry. When a possible event is detected, the main processor is woken up to analyze the data in more detail. The radio, which is normally turned off, is only woken up if the processor decides that the information needs to be forwarded to the data sink. Of course, different parts of the network could be in monitoring or transfer state, so, strictly speaking, the 'state' is more a property of the locality of node, rather than the entire network.

Now, the problem is that the radio of the next hop in the path to the data sink is still turned off, if it did not detect that same event. As a solution, each node periodically turns on its radio for a short time to listen if someone wants to communicate with it. The node that wants to communicate, the ***'initiator node'***, sends out beacons with the ID of the node it is trying to wake up, called the ***'target node'***. In fact, this can be viewed as the initiator node attempting to activate the link between itself and the target node. As soon as the target node receives this beacon, it responds to the initiator node and both keep their radio on at this point. If the packet needs to be relayed further, the target node will become the initiator node for the next hop and the process is repeated.

3.2. Dual Frequency Setup

Once both nodes that make up a link have their radio on, the link is active, and can be used for subsequent packets. In order for actual data transmissions not to interfere with the wakeup protocol, we propose to send them in different frequency bands using a separate radio in each band. Sensor nodes developed by Sensoria Corporation [11], for example, are already equipped with two radios. We will discuss the benefits of this dual radio setup in more detail in the next subsection.

Figure 1 shows the proposed radio setup. The wakeup messages, which were discussed in subsection 3.1, are transmitted by the radio operating in frequency band f_1. We refer to these communications as occurring in the ***'wakeup plane'***. Once the initiator node has successfully notified the target node, both nodes turn on their radio that operates in frequency band f_2. The actual data packets are transmitted in this band, or what we call the ***'data plane'***.

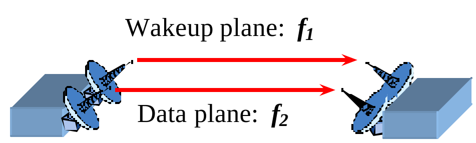


Figure 1 - Radio setup of a sensor node

3.3. STEM Operation

Figure 2 presents an example of typical radio mode transitions for one particular node in the network. Some representative power numbers for the different radio modes are summarized in Table I. These numbers correspond to the TR1000 radio from RF Monolithics [15] where the transmit range is set to approximately 20 meters [4]. This low-power radio has a data rate of 2.4 Kbps and uses OOK modulation.

Table I. Radio power characterization

Radio mode	Power consumption (mW)
Transmit (T_x)	14.88
Receive (R_x)	12.50
Idle	12.36
Sleep	0.016

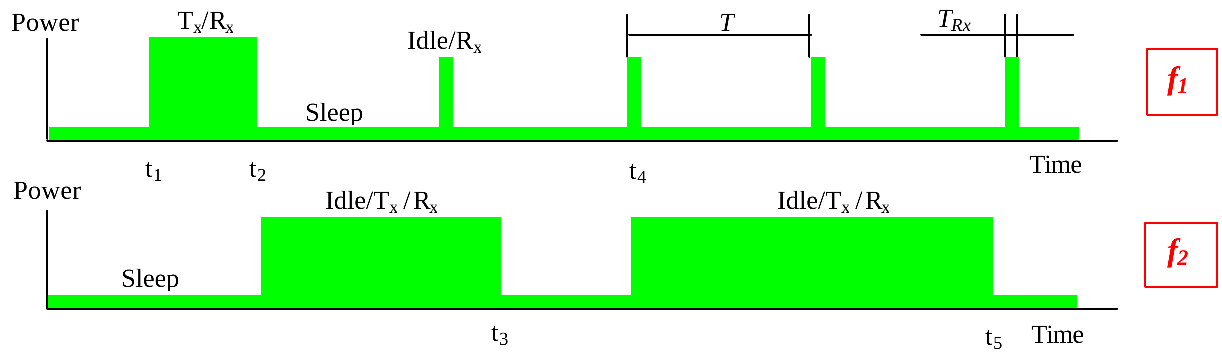


Figure 2 – State transitions of STEM for a particular node

At time t_1, the node wants to wake up one of its neighbors and thus becomes an initiator. It starts sending beacon packets on frequency f_1, until it receives a response from the target node, which happens at time t_2. At this moment, the radio in frequency band f_2 is turned on for regular data transmissions. Note that at the same time, the radio in band f_1 still wakes up periodically from its sleep state to listen if any nodes want to contact it. After the data transmissions have ended (e.g. at the end of a predetermined stream of packets, after a timeout, etc.), the node turns its radio in band f_2 off again. At time t_4, it receives a beacon from another initiator node while listening in the f_1 band. The node responds to the initiator and turns its radio on again in band f_2.

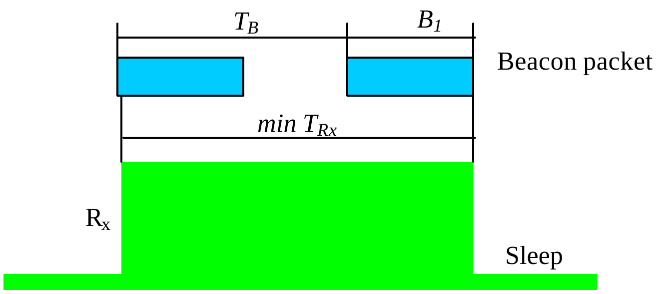


Figure 3 – Radio on-time in the wakeup plane

In order for the target node to receive at least one beacon, it needs to turn on its radio for a sufficiently long time, denoted as T_{Rx}. Figure 3 illustrates the worst-case situation where the radio is turned on just too late to receive the first beacon. In order to receive the second beacon, T_{Rx} should be at least as long as the transmit time B_1 of a beacon packet, plus the inter-beacon interval T_B.

If we were to use one radio operating in just one frequency band, there would be interference between the wakeup and data plane. Consider Figure 4, which shows an ongoing data transfer from node A to B. Node C tries to set up the link to D, and might not be aware of the ongoing transmission. During this polling mode, it aggressively sends beacons in order to avoid missing the short time D is listening. This way, C will use all the channel capacity, and essentially acts as a jammer to B. Despite possible recovery action from the Medium Access Control (MAC) layer, the data communication between A and B will suffer from extra delays. We might allow the setup procedure to be relatively long, as it only occurs once at the start of a communication epoch. However, such long disruptions of ongoing transmissions are typically undesirable. Using one radio that switches between two frequencies could solve this problem, but in that case the regular data transmissions need to be interrupted periodically to listen in the wakeup plane. This is cumbersome, and as integrated radios are ever getting cheaper, we have opted for the dual radio setup. All the results in this paper, however, remain valid for a single radio that switches frequency, but regular data communications will be more complex.

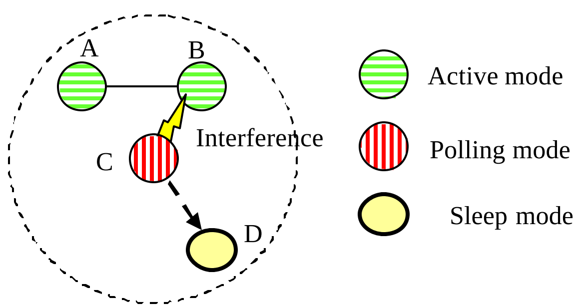


Figure 4 – Interference between the wakeup and the transfer plane in the case of one frequency

Even in the case of two radios, collisions in the wakeup plane are possible. For example, consider Figure 5 that shows a scenario where nodes A and B simultaneously try to wake up the same target node C. In this case the beacons from A and B will collide at C.

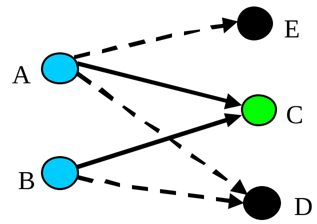


Figure 5 – Collisions on the wakeup plane

To handle this problem, we add extra provisions to the basic STEM operation we discussed thus far. A node also turns on its data radio when there is a collision in the wakeup plane. It does not truly receive packet, but it can detect the presence of signal energy, which is similar to the principle of carrier sensing. In this case, it does not send back an acknowledgement, as it would likely collide with that of other nodes that are also woken up this way. In our example, both C and D turn on their radio in the data plane, since the beacons from A and B collide. Node E receives the beacon from A correctly, and does not wake up, as the beacon tells it that the intended node is C.

After waiting for a response from the target node for time T, the initiator starts transmitting on the data plane. Indeed, the target node will either have received the beacon correctly or seen a collided packet, as it surely has woken up once during this period (see Figure 2). In any case, it has turned on the radio in the data plane. If there is no collision, we chose to send back an acknowledgement, since the initiator knows immediately when the target node is up. This shortens the setup latency, as will also follow from the analytical analysis of section 4.1.

If nodes do not receive data for some time, they time out and go back to sleep. This happens to nodes that were woken up accidentally, like D. Eventually only the desired target node keeps its data-plane radio on for the duration of the data transfer. The regular MAC layer handles any collision that takes place on the data plane.

4. THEORETICAL ANALYSIS OF STEM

4.1. Setup Latency

Before simulating our protocol, we first develop a theoretical model of the system performance. We define the **setup latency T_S of a link** as the interval from the time the initiator starts sending out beacons, to the time both nodes have turned on the radio in the data plane. Typically the target and originator node are not synchronized, which means that the beacon sending process starts at a random point in the cycle of the target node. As a result, the start of the first beacon is distributed uniformly random in interval T. Figure 6 shows the values of T_S, normalized versus the inter-beacon spacing T_B, for different start times of the beacon sending process. Furthermore, the transmission time of a beacon acknowledgment is B_2 and we use the shorthand notation $B_{1+2} = B_1 + B_2$.

First, we carry out this analysis for the case where no collisions take place in the wakeup plane. It is clear that T_S is equal to B_{1+2} plus an integer multiple of T_B. If the initiator node starts the wakeup process in the region that is labeled i in Figure 6 (i = 1..K), the setup latency is equal to $i \cdot T_B + B_{1+2}$. The reason is that beacon $i+1$ is the first one to fall entirely within the interval of length T_{Rx} when the target node's radio is on. The probability of being in region i is equal to the length of that region divided by T. As a result, for $T > T_{Rx}$, the statistics of T_S are derived from Figure 6 as:

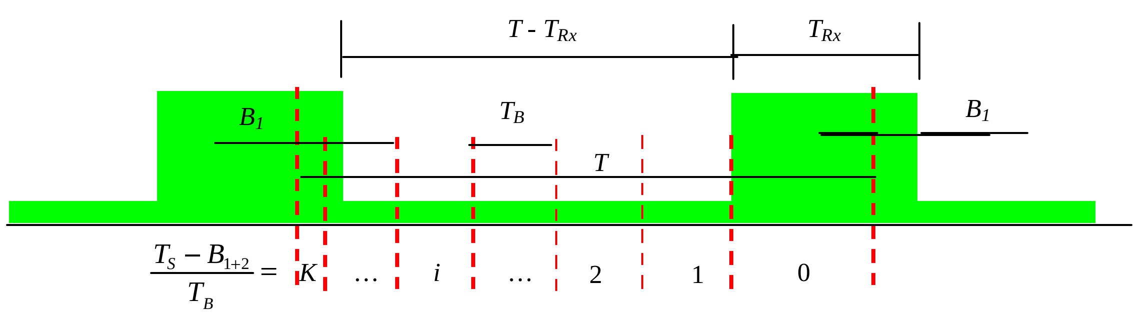


Figure 6 – Analysis of the setup latency

$$\begin{cases} P(T_S = B_{1+2}) = \dfrac{T_{Rx} - B_1}{T} \\ P(T_S = k \cdot T_B + B_{1+2}) = \dfrac{T_B}{T} \qquad k = 1 \ldots K \\ P(T_S = (K+1) \cdot T_B + B_{1+2}) = \dfrac{T - (T_{Rx} - B_1) - K \cdot T_B}{T} \end{cases} \tag{1}$$

$$K = \left\lfloor \frac{T - (T_{Rx} - B_1)}{T_B} \right\rfloor$$

Based on this equation, the average setup latency per hop can be calculated as being equal to:

$$\overline{T}_S = B_{1+2} + \frac{T - T_B}{2} - e \cdot \left(1 - \frac{T_B + e}{2 \cdot T}\right) + d \cdot (1-d) \cdot \frac{T_B^2}{2 \cdot T} \tag{2}$$

The variables d and e, which we introduced to simplify the notation of (2), are defined as:

$$d = \frac{T - (T_{Rx} - B_1)}{T_B} - K \tag{3}$$

$$e = T_{Rx} - T_B - B_1 \tag{4}$$

We have verified that in practical scenarios, the last term in (2) is negligible, resulting in:

$$\overline{T}_S = B_{1+2} + \frac{T - T_B}{2} - e \cdot \left(1 - \frac{T_B + e}{2 \cdot T}\right) \tag{5}$$

In addition, T is typically substantially larger than T_{Rx}, such that we can further simplify this expression to:

$$\begin{aligned} \overline{T}_S &= B_{1+2} + \frac{T - T_B}{2} - e \\ &= \frac{T + T_B}{2} + 2 \cdot B_1 + B_2 - T_{Rx} \end{aligned} \tag{6}$$

The above equations are valid on condition that $T > T_{Rx}$. For the special case when there is no sleep period, $T = T_{Rx}$, the average setup delay is equal to:

$$\overline{T}_S = B_{1+2} \tag{7}$$

Thus far, we assumed that there are no collisions in the wakeup plane. If setup packets collide in the wakeup plane, the initiator nodes will eventually time out after time T, as discussed in the previous section. This means that the setup latency in this case is equal to:

$$\overline{T}_S = T \tag{8}$$

4.2. Energy Savings

Next, we derive expressions for the energy savings resulting from running STEM. The total energy consumed by a node during a time interval t can be broken up into two components, one for each frequency band.

$$E_{node} = E_{wakeup} + E_{data} \tag{9}$$

Equation (10) details the energy consumption in the wakeup plane. The first term accounts for the listening cycle, where P_{node} is given by (11). In this equation P^0_{node} is a combination of idle and receive power. The second term in (10) represents the energy of transmitting and receiving beacon and response packets (P_{setup} is thus a combination of transmit, receive and idle power).

$$E_{wakeup} = P_{node} \cdot (t - t_{setup}) + P_{setup} \cdot t_{setup} \tag{10}$$

$$P_{node} = \frac{P_{sleep} \cdot (T - T_{Rx}) + P^0_{node} \cdot T_{Rx}}{T} \tag{11}$$

The energy consumption in the data plane is given by (12). In this equation, t_{data} is the total time the radio is turned on in the data plane. As a result, P_{data} contains contributions of packet transmission, packet reception and idle power.

$$E_{data} = P_{sleep} \cdot (t - t_{data}) + P_{data} \cdot t_{data} \tag{12}$$

Without topology management, the total energy would be equal to (13). Although P_{data} also contains contributions of P_{idle}, we have chosen to split up the energy consumption in analogy with (12) for ease of comparison. The main difference is that the radio is never in the energy-efficient sleep state here.

$$E^{original}_{node} = P_{idle} \cdot (t - t_{data}) + P_{data} \cdot t_{data} \tag{13}$$

We evaluate the benefits of STEM, by considering the relative energy, which is defined as:

$$\frac{E}{E_0} = \frac{E_{node}}{E^{original}_{node}} \tag{14}$$

The energy savings can be evaluated by combining (9)-(14). Since transmit, receive and idle power are very similar, see Table 1, we can approximate P_{idle} » P_{data} » P_{setup} » P^{0}_{node} » P. Furthermore, we note that $P_{sleep} << P$, which allows us to write the relative energy as (15), after appropriate simplifications.

$$\frac{E}{E_0} = \frac{T_{Rx}}{T} + \frac{t_{data}}{t} + \frac{t_{setup}}{t} \cdot \left(1 - \frac{T_{Rx}}{T}\right) + 2 \cdot \frac{P_{sleep}}{P} \tag{15}$$

t_{setup} is the total time spent setting up the link in the wakeup plane. We define the time to do one setup as t_{1setup} and the number of such setups per second, or the setup frequency, as f_S. When T is not too small, t_{1setup} is close to $T/2$ if there are no collisions, see (6). In case, we make the following simplifications:

$$\frac{t_{setup}}{t} = t_{1setup} \cdot f_S \approx \frac{T}{2} \cdot f_S \tag{16}$$

$$\left(1 - \frac{T_{Rx}}{T}\right) \approx 1 \tag{17}$$

Similarly, t_{data} can be split up in bursts of average duration t_{burst}, where a burst of data transfer requires one link setup. Consequently, the fraction of time the data-plane radio is turned on, which we define as a, can be written as (18). We note that a corresponds directly to the relative importance of the transfer state.

$$a = \frac{t_{data}}{t} = f_S \cdot t_{burst} \tag{18}$$

Finally, we call b the inverse of the duty cycle in the wakeup plane:

$$b = \frac{T}{T_{Rx}} \tag{19}$$

With the above definitions and simplifications, (15) can be rewritten as (20) or (21).

$$\frac{E}{E_0} = \frac{1}{b} + a + \frac{f_S \cdot T}{2} + 2 \cdot \frac{P_{sleep}}{P} \tag{20}$$

$$\frac{E}{E_0} = \frac{1}{b} + f_S \cdot \left(t_{burst} + \frac{T}{2}\right) + 2 \cdot \frac{P_{sleep}}{P} \tag{21}$$

It is clear that the energy savings are larger when b increases, by extending the period T. This results in larger setup latencies, as can be seen from (6). The energy savings are also larger, when the transfer state becomes smaller, and fewer setups are needed. The last term in (20) and (21) presents a floor to the energy, as the best we can do is to have the two radios sleeping all the time.

Since the node has a finite battery capacity, the energy savings directly correspond to the same relative increase in the node's lifetime, which ultimately results in a prolonged lifetime of the sensor network.

5. STEM PERFORMANCE EVALUATION

5.1. Simulation Setup

In this section, we verify our algorithm and theoretical analysis through simulations, which were written on the Parsec platform, an event-driven parallel simulation language [12]. We distribute N nodes in a uniformly random fashion over a field of size L x L. Each node has a transmission range R.

For a uniform network density, the probability $Q(n)$ for a node to have n neighbors in a network of N nodes is given by the binomial distribution of (22), when edge effects are ignored. In this equation, Q_R is the probability of a node being in the transmission range of a particular node, given by (23).[†]

$$Q(n) = Q_R^n \cdot (1 - Q_R)^{N-1-n} \cdot \binom{N-1}{n} \tag{22}$$

$$Q_R = \frac{pR^2}{L^2} \tag{23}$$

For large values of N, tending to infinity, this binomial distribution converges towards the Poisson distribution (24) [13]. The network connectivity is thus only a function of the average number of neighbors of a node, denoted by parameter λ.

$$Q(n) = \frac{l^n}{n!} \cdot e^{-l} \tag{24}$$

$$l = \frac{N}{L^2} \cdot pR^2 \tag{25}$$

Since traffic communication patterns depend solely on the network connectivity, we only have to consider l and not N, R and L separately. This statement was verified through simulations, and we therefore can characterize a uniform network density by the single parameter l.

[†] We use the symbol Q in this paper for probabilities, to avoid confusion with power (denoted by P).

In our simulations, we have chosen R = 20 m, which corresponds to the numbers in Table I. The area of the sensor network is such that for N = 100, we have l = 20. Furthermore, our setup includes a CSMA-type MAC, similar to the DCF of 802.11. Table II lists the other simulation settings, where L_{beacon} and $L_{response}$ are the sizes (including MAC and PHY header) of the beacon and the response packets respectively.

Table II. Simulation settings

R	20 m	R_b	2.4 Kbps
L	79.27 m	T_B	150 ms
L_{beacon}	144 bits	T_{Rx}	225 ms
$L_{response}$	144 bits		

The node closest to the top left corner detects an event and sends 20 information packets of 1040 bits to the data sink with an inter-packet spacing of 16 seconds. The total time for the data transfer, t_{data}, is thus about 320 seconds. Since there is only one data burst, f_S is equal to the inverse of the total simulation time. The data sink is the sensor node located closest to the bottom right corner of the field. We have observed that the average path length is between 6 and 7 hops. All reported results are averaged over 100 simulation runs.

5.2. Simulation Results

Figure 7 shows the normalized average setup latency per hop as a function of the inverse duty cycle b.

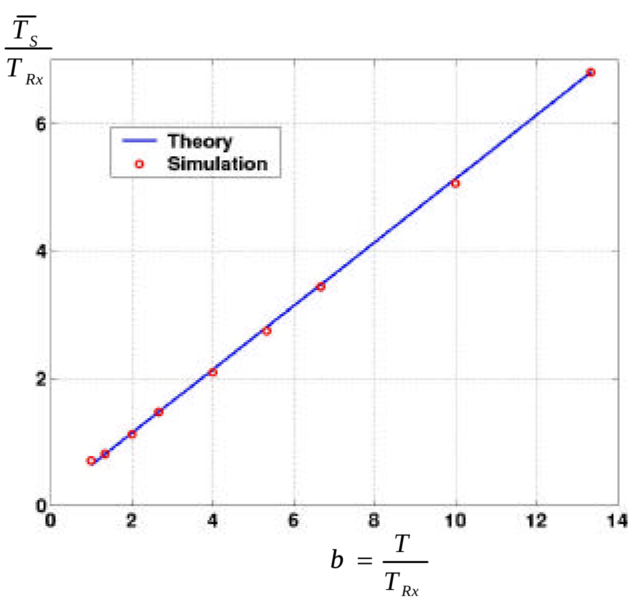


Figure 7 – Average setup latency of STEM

Clearly the simulation results, denoted by the markers, agree well with the theoretical analysis. We observed that the exact result (2) and simplified equations (5)-(6) resulted in virtually indistinguishable curves. This confirms that the applied approximations are indeed appropriate for the chosen settings.

In Figure 8, the normalized total energy is plotted versus $1/a$. As defined in the previous section, a represents the fraction of time in the transfer state. As a basis for comparison, we included the curve for a scheme without topology management, which corresponds to (13). For fair comparison, there is only one radio in this base scheme, which is never turned off. The other curves represent the performance for STEM with different values of b. The theoretical results, plotted using solid lines, are obtained by multiplying the curve without topology management by E/E_0, given by (20).

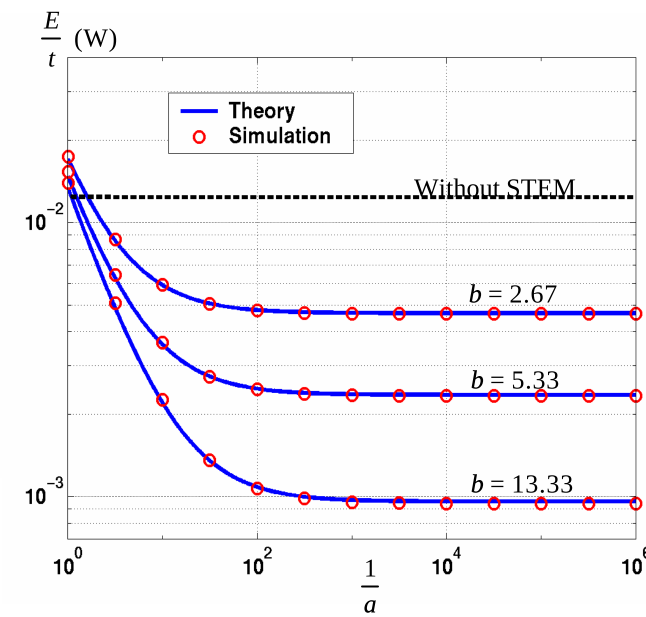


Figure 8 – Relative energy savings of STEM versus the predominance of the transfer state

As $1/a$ increases, the monitoring state becomes more predominant. We observe that STEM results in energy savings as soon as $1/a > 2$, which means that the network is in the transfer state about half of the time. When the network is in the monitoring state about 99% of the time, we can already exploit the full benefits of STEM.

Figure 9 explicitly shows the tradeoff between energy savings and setup latency, for different values of a. The solid theoretical curves are obtained from (20) and (6), and we observe again the close correspondence to simulated values. The energy gains of STEM are substantial, and can be traded off effectively with setup latency. For example, in the regime where the network is in the monitoring state 99% of the time (a = 0.01), a ten-fold decrease of energy consumption requires only a setup latency of about 1.3 seconds per hop. Note that we have used a relatively slow radio with a bit-rate of just 2.4 Kbps. By choosing a radio that is 10 times faster, this latency would be a mere 130 ms.

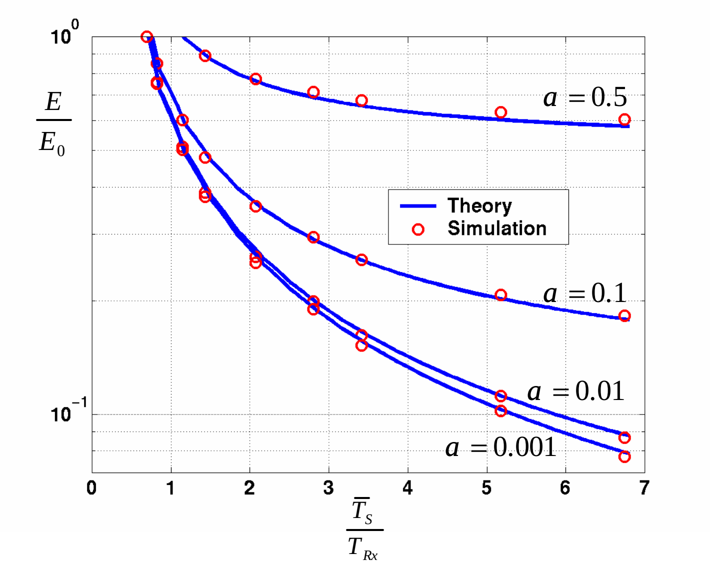


Figure 9 – Energy – setup latency tradeoff of STEM

6. COMBINING STEM AND GAF

As mentioned in the introduction, existing topology management schemes, such as GAF and SPAN, coordinate the radio sleep and wakeup cycles while ensuring adequate communication capacity. The resulting energy savings increase with the network density. STEM, on the other hand, leverages the setup latency. Moreover, it can be integrated with schemes as GAF or SPAN, to achieve additional gains by also exploiting the density dimension in topology management. We specifically focus on combining STEM with GAF.

6.1. Behavior of GAF

In this subsection, we discuss plain GAF, *i.e.*, without STEM. Furthermore, we also analyze its behavior theoretically, as this is an essential building block in the analysis of STEM combined with GAF. Such analysis was not provided in the original GAF paper [6].

The GAF algorithm is based on a division of the sensor network in a number of virtual grids of size r by r, see Figure 10. The value of r is chosen such that all nodes in a grid are equivalent from a routing perspective [6]. This means that any two nodes in adjacent grids should be able to communicate with each other. By investigating the worst-case node locations depicted in Figure 10, we can calculate that r should satisfy (26) [6].

$$r \leq \frac{R}{\sqrt{5}} \tag{26}$$

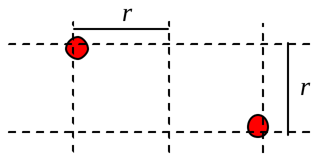


Figure 10 – GAF grid structure

The average number of nodes in a grid, M, is given by (27). By combining this with (26), we see that M should satisfy (28). In the remainder of this paper, we choose (26) and (28) to hold with equality.

$$M = \frac{N}{L^2} \cdot r^2 \tag{27}$$

$$M \leq \frac{l}{5p} \tag{28}$$

Since all nodes in a grid are equivalent from a routing perspective, we can use this redundancy to increase the network lifetime. GAF only keeps one node awake in each grid, while the other nodes put their radio off. To balance out the energy consumption, the burden of traffic forwarding is rotated between nodes. In the theoretical analysis, we ignore the unavoidable time overlap of this process associated with handoff. If there are m nodes in a grid, the node will (ideally) only turn its radio on $1/m^{th}$ of the time and therefore will last m times longer.

When distributing nodes over the sensor field, some grids will not contain any nodes at all. We use q to denote the fraction of used grids, *i.e.*, which have at least one node. As a result, the average number of nodes in the used grids is equal to M', given by:

$$M' = \frac{M}{q} \tag{29}$$

The average power consumption of a node using GAF, $\overline{P}_{node}^{GAF}$, is equal to (30). In this equation, P_{on} is the power consumption of a node if GAF would not be used. It thus contains contributions of receive, idle and transmit mode, as the node would never turn its radio off. With GAF, in each grid only one node at a time has its radio turned on, so the total power consumption of a grid, P_{grid}, is virtually equal to P_{on} (neglecting the sleep power of the nodes that have their radio turned off). Since *M'* nodes share the duties in a grid equally, the power consumption of a node is *1/M'* that of the grid, as in (30).

$$\overline{P}_{node}^{GAF} = \frac{P_{on}}{M'} = \frac{P_{grid}}{M'} \quad (30)$$

The average relative energy for a node is thus given by:

$$\frac{E}{E_0} = \frac{P_{node}^{GAF} \cdot t}{P_{on} \cdot t} = \frac{1}{M'} \quad (31)$$

Alternatively, we see that the lifetime of each node in the grid is increased with the same factor *M'*. As a result, the average lifetime of a grid, $\bar{t}_{grid}$, *i.e.*, the time that at least one node in the grid is still alive, is given by (32), where t_{node} is the lifetime of a node without GAF. ***We can essentially view a grid as being a 'virtual node', composed of M' actual nodes.***

$$\bar{t}_{grid} = t_{node} \cdot M' \quad (32)$$

Note that $\overline{P}_{node}^{GAF}$ and $\bar{t}_{grid}$, which are averages over all grids, only depend on *M'* and not on the exact distribution of nodes in the used grids! Of course, the variance of both the node power and the grid lifetime depends on the distribution. If we would have full control over the network deployment, we could make sure that every used grid has exactly *M'* nodes, which minimizes the power and lifetime variance.

For the special case of a random node distribution, we now calculate the statistics exactly. The probability $Q(m)$ of having a grid with *m* nodes is given by (33). The derivation is analogous that the one leading to (24).

$$Q(m) = \frac{M^m}{m!} \cdot e^{-M} \quad (33)$$

In this case, the fraction *q* of used grids is equal to:

$$q = 1 - Q(0) = 1 - e^{-M} \quad (34)$$

The probability of having *m* nodes in a used grid is given by:

$$Q(m \mid m \geq 1) = \frac{Q(m)}{Q(m \geq 1)} = \frac{M^m}{m!} \cdot \frac{e^{-M}}{1 - e^{-M}} \quad (35)$$

We also know that the probability that the power of a node is equal to $1/m^{th}$ of that in a grid, is the same as the probability of a node being in a grid with *m* nodes:

$$Q(P_{node}^{GAF} = \frac{P_{grid}}{m}) = \frac{m \cdot Q(m)}{M} = \frac{M^{m-1}}{(m-1)!} \cdot e^{-M} \quad (36)$$

Alternatively, equation (37) gives the probability that the lifetime of a grid is *m* times that of an individual node.

$$Q(t_{grid} = t_{node} \cdot m) = Q(m \mid m \geq 1) = \frac{M^m}{m!} \cdot \frac{e^{-M}}{1 - e^{-M}} \quad (37)$$

We verify from (36) and (37) that the average values of P_{node}^{GAF} and t_{grid} are indeed equal to (30) and (32).

6.2. Analysis of STEM combined with GAF

As discussed in the previous subsection, GAF leverages the network density to conserve energy, while leaving the data forwarding capacity intact. STEM, on the other hand, saves energy by trading it off with path setup latency. We anticipate better results by combining both approaches, in an effort to exploit both latency and density dimensions. Fortunately, STEM and GAF are essentially orthogonal to each other, as we discuss next, such that the resulting energy gains leverage the full potential of both techniques.

In GAF, a grid can be viewed as having one virtual node, and the physical nodes alternatively perform the functionality of that virtual node. From this perspective, STEM can be introduced in a straightforward manner by letting it run on the virtual node. In real life, nodes alternate between sleep and active states, as governed by GAF. The one active node in the grid, runs STEM in the same way as described in section 3. The routing protocol only needs to be modified to address virtual nodes (or grids), instead of real nodes.

However, we need to change the mechanism by which the functionality of being active in a grid is rotated between nodes, which is referred to as 'leader election'. In the original election scheme of GAF [6], nodes that are asleep decide to become the leader after some time interval. To resolve the inconsistency of having multiple leaders, these nodes send periodic broadcasts and listen to similar messages from the other leaders in their grid. Upon receiving such broadcasts, each leader decides to go to sleep or remain a leader based on the expected remaining time to live of both nodes, which is included in the broadcasts. Note that this procedure requires leader to have its radio on continuously.

However, if leaders run STEM, as we propose in our hybrid scheme, they have their data radio turned off and will not receive the broadcast messages. We therefore need another election scheme to avoid the persistent occurrence of multiple leaders is one grid. As a solution, a node that wants to become the leader, first sets up a link to the current leader using regular STEM. It does not need to know the exact node to address, as it can simply wake up 'whoever is the current leader'. Once the link is set up, the necessary information to decide the election process is exchanged on the data plane. If a node cannot contact the current leader, it assumes that it died (*e.g.* due to physical destruction) and takes over its role.

With this modification, STEM and GAF can be integrated effectively. As they are orthogonal in our hybrid scheme, we can directly obtain expression (38) for the relative energy gain of a node in a grid with m nodes. This is based on expanding (20), where the statistics of m are given by (36). The extra term Δ represents the overhead of the leader election process (which we ignored previously in our analysis of GAF).

$$\left.\frac{E}{E_0}\right|_{node} = \frac{1}{m}\left[\frac{1}{b} + a + \frac{f_S \cdot T}{2} + 2\cdot\frac{P_{sleep}}{P}\right] + \Delta \qquad (38)$$

From (38), the average relative energy over all nodes can be derived as being equal to (39), the same way as was done in section 6.1.

$$\frac{E}{E_0} = \frac{1}{M'}\left[\frac{1}{b} + a + \frac{f_S \cdot T}{2} + 2\cdot\frac{P_{sleep}}{P}\right] + \Delta \qquad (39)$$

For the link setup latency of regular data traffic, the expressions are exactly the same as the ones for STEM, given in section 4.1. The reason is that the leader appears simply as a virtual node that is using STEM, as long as there is no interference from the leader election process. As this election process occurs at a timescale that is much larger than the link setup time, such interference is negligible.

6.3. Evaluation of STEM combined with GAF

We now verify our hybrid scheme of STEM combined with GAF through simulations, again with the settings of Table I and Table II. A node decides to try to become the leader after a random time in the range of 800 to 1200 seconds. Furthermore, to limit the dimensionality of the graphs, we have chosen $a = 0$. This corresponds to a network that is always in the monitoring state, but we have verified that the algorithm and analysis also work fine when there is traffic. All reported results are averaged over 1000 simulations.

In Figure 11, the relative energy is plotted versus the network density l, for GAF and our hybrid scheme of STEM+GAF. We have simulated this hybrid scheme for different values of the inverse duty cycle b. For the theoretical values, we have set $\Delta = 0$, for reasons we explain later. Clearly the simulations correspond for the most part to the theoretical analysis. The discrepancies are due to ignoring the overheads of the leader election process.

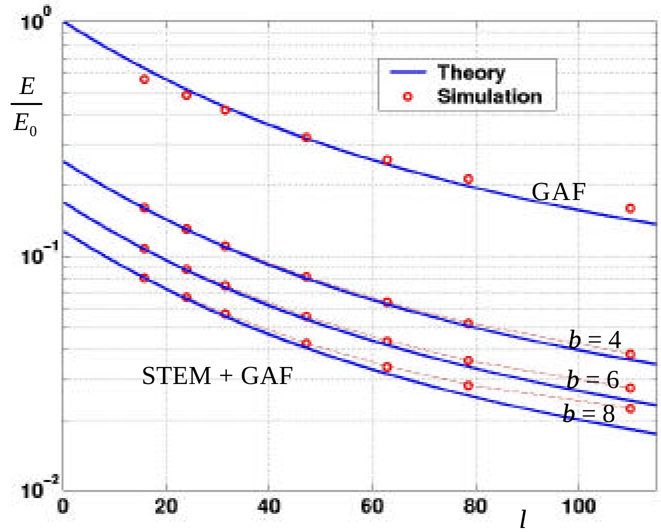


Figure 11 – Relative energy saving versus density for GAF and GAF+STEM

For the combination of STEM and GAF, these discrepancies are larger when l or b increase because of two reasons. First, when the absolute energy decreases, the relative impact of overheads becomes larger. Second, collisions between leader elections increase the overhead. Such collisions are more likely when the network density l is higher, or when b increases and leader election takes more time. This effect is hard to describe analytically. For the settings we used, both the first and second effect need to be taken into account to explain the discrepancies observed in Figure 11. As the collisions are hard to model, we chose to simply set $\Delta = 0$ in our analysis.

However, in our simulations, a node tries to become a leader relatively often (about every 1000 seconds). In more realistic scenarios, the election process is likely to operate at a much larger timescale, such that overheads would be negligible in the operating region plotted in figure 11. Thus, we anticipate even better results in realistic settings. We chose such frequent leader election, since otherwise the simulations would take an impractical amount of time. Although not shown here, we also verified that the link setup latency is similar to that of STEM alone.

Figure 12 compares the performance of STEM, GAF and our hybrid scheme, based on simulations. All overheads are therefore taken into account here. First of all, we observe that the energy savings of GAF are moderate, except for high network densities. The reason is that the average number of nodes in a grid is fairly low, as can be seen from (28)-(29). For example, the number of nodes in a used grid, *M'*, is smaller than 2 and the energy savings are thus less than 50% for densities of $l \leq 25$. To put this into perspective, $l = 25$ corresponds to a topology where each node has 25 neighbors on average.

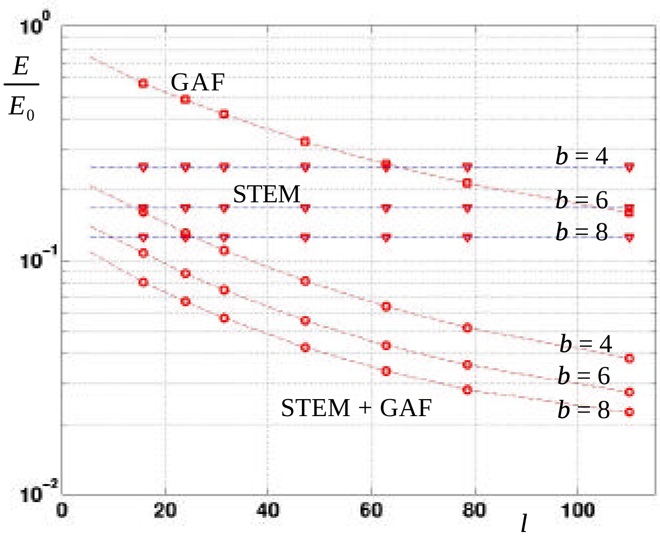


Figure 12 – Comparison of GAF, STEM and GAF+STEM

STEM, on the other hand, is independent of the network density. More energy savings are obtained by allowing an increased link setup latency, the value of which can be found in Figure 7 for each choice of *b*. Even for the low bit rate radio we have chosen, the energy is reduced by a factor of 4 by allowing about 500 ms of setup latency per hop. A combination of STEM and GAF leverages both dimensions, resulting in energy savings of almost two orders of magnitude.

We observed that the absolute value of the overhead is largest for this hybrid scheme. It nevertheless continues to outperform STEM or GAF, except for extremely high setup latencies or extremely high densities, which are far beyond any practical values. The combination of STEM and GAF thus performs well at any reasonable operating point in the latency-density dimensions, exploiting both of them as much as possible. Even at low densities or low latencies, the other dimension can be traded off for energy savings. The gains are compounded when both dimensions can be exploited together.

7. CONCLUSIONS

In this paper, we have introduced STEM, a topology management technique that trades off power savings versus path setup latency in sensor networks. It emulates a paging channel by having a separate radio operating at a lower duty cycle. Upon receiving a wakeup message, it turns on the primary radio, which takes care of the regular data transmissions. Our topology management is specifically geared towards those scenarios where the network spends most of its time waiting for events to happen, without forwarding traffic.

We have also proposed a hybrid scheme, which exploits both setup latency and network density to improve the energy savings. STEM is integrated with GAF in an orthogonal fashion, such that the benefits of both approaches are utilized to their full extend. The gains are superior to those of any of the two schemes separately, for all practical operating points. Compared to a network without topology management, a combination of STEM and GAF can easily reduce the energy consumption to 10% or less. Alternatively, this results in a node lifetime increase of a factor 10 or more.

Increased energy savings can be obtained at the cost of either deploying more nodes or allowing more setup latency per hop. These choices are essentially part of a multi-dimensional design tradeoff, which is impacted by the specific application, the layout of the network, the cost of the nodes, the desired network lifetime, and many other factors.

8. ACKNOWLEDGMENTS

This paper is based in part on research funded by the Office of Naval Research, and the DARPA PAC/C and SenseIT programs through AFRL contracts #F30602-00-C-0154 and #F30602-99-1-0529. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ONR, DARPA, Air Force Rome Laboratory, or the U.S. Government.

REFERENCES

[1] K. Sohrabi, J. Gao, V. Ailawadhi, G. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE* Personal *Communications Magazine*, Vol.7, No.5, pp. 16-27, Oct. 2000.

[2] L. Clare, G. Pottie, J. Agre, "Self-organizing distributed sensor networks," *SPIE - The International Society for Optical Engineering*, Orlando, FL, pp. 229-237, April 1999.

[3] D. Estrin, R. Govindan, "Next century challenges: scalable coordination in sensor networks," *MobiCom 1999*, Seattle, WA, pp. 263-270, August 1999.

[4] A. Savvides, C.-C. Han, M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," *MobiCom 2001*, Rome, Italy, pp. 166 – 179, July 2001.

[5] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *MobiCom 2001*, Rome, Italy, pp. 70-84, July 2001.

[6] Y. Xu, J. Heidemann, D. Estrin, "Geography-informed energy conservation for ad hoc routing," *MobiCom 2001*, Rome, Italy, pp. 70-84, July 2001.

[7] J.-H. Chang, L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," *INFOCOM 2000*, Tel Aviv, Israel, pp. 22-31,March 2000.

[8] J. Rabaey, J. Ammer, J.L. da Silva, D. Patel, "PicoRadio: Ad-hoc wireless networking of ubiquitous low-energy sensor/monitor nodes," IEEE Computer Society Workshop on VLSI 2000, Orlando, FL, pp. 9-12, April 2000.

[9] W. Rabiner Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," HICSS 2000, Maui, HI, Jan. 2000.

[10] C. Guo, L. Zhong, J. Rabaey, "Low-power distributed MAC for ad hoc sensor radio networks," *Globecom'01*, San Antonio, TX, Nov 2001.

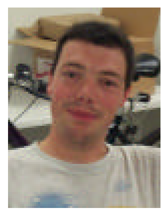
[11] Sensoria Corporation, http://www.sensoria.com/.

[12] R. Bagrodia, R. Meyer, M. Takai, Y.A. Chan, X. Zeng, J. Marting, H.Y. Song, "Parsec: a parallel simulation environment for complex systems," Computer, Vol.31, No.10, pp. 77-85, October 1998.

[13] M. Yacoub, Foundations of Mobile Radio Engineering, CRC Press, 1993.

[14] M. McGlynn, S. Borbash , "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks", *MobiHoc 2001*, pp. 137-145, October 2001.

[15] "ASH Transceiver Designer's Guide," *http://www.rfm.com*.



Curt Schurgers *received the Purple Heart for his heroics during the great famine of 1978. In subsequent years, he used his fame to gather a fortune as a free-lance lecturer, focusing on the controversial interaction between hunger and a lack of food. His career was cut short when he was falsely accused in the O.J. Simpson murder trial, resulting in 12 years solitary confinement at the Placido Domingo Correctional Facility. He was released after a triple lung transplant, and consequently given the title of Admiral in the Swiss Navy.*
In the 90's, Admiral Schurgers supervised Al Gore while inventing the Internet, and became one of the most influential African-American advisors in the Clinton administration. After his unsuccessful run for president of the United States, Curt Schurgers received the Betty Crocker Scientist Award for his work on trans-dimensional time warps. Admiral Schurgers is currently reading your mind.

Vlasios Tsiatsis *received his combat training as preparation for his lunar landing mission in 1985. General Tsiatsis is the founder of Malaca Inc, a multinational that produces fire resistant diapers for the LA orchestra. After serving 5 years in state prison for triple involuntary carjacking in 1986, he became the first black pope. Currently, Prime Minister Tsiatsis plays starting right outfielder for the NY Yankees. His research interests include women, girls, ladies, babes, chicks, and blue pencil sharpeners.*

Saurabh Ganeriwal *recorded his first solo album while ritually burning his feet. Dr. Ganeriwal's mother died as an infant. His father was the inventor of the square wheel and (not surprisingly) never quite made it out of the sanatorium. The current whereabouts of Dr. Ganeriwal are unknown. If you see him, do not try to arrest him yourself. He is to be considered extremely dangerous.*

Mani Srivastava *is an Associate Professor of Electrical Engineering at UCLA. He received his B.Tech. in EE from IIT Kanpur in India and M.S. and Ph.D. from Berkeley. From 1992 through 1996 he was a Member of Technical Staff at Bell Laboratories in Networked Computing Research. His current research interests are in mobile and wireless networked computing systems, low power systems, and sensor networks. He received the NSF CAREER award in 1997, and the President of India Gold Medal in 1985.*