

# Techniques for Maintaining Connectivity in Wireless Ad-hoc Networks Under Energy Constraints

FARINAZ KOUSHANFAR

Rice University

ABHIJIT DAVARE, DAVID T. NGUYEN, and

ALBERTO SANGIOVANNI-VINCENTELLI

University of California Berkeley

and

MIODRAG POTKONJAK

University of California Los Angeles

---

Distributed wireless systems (DWSs) are emerging as the enabler for next-generation wireless applications. There is a consensus that DWS-based applications, such as pervasive computing, sensor networks, wireless information networks, and speech and data communication networks, will form the backbone of the next technological revolution. Simultaneously, with great economic, industrial, consumer, and scientific potential, DWSs pose numerous technical challenges. Among them, two are widely considered as crucial: autonomous localized operation and minimization of energy consumption. We address the fundamental problem of how to maximize the lifetime of the network using only local information, while preserving network connectivity. We start by introducing the care-free sleep (CS) Theorem that provides provably optimal conditions for a node to go into sleep mode while ensuring that global connectivity is not affected. The CS theorem is the basis for an efficient localized algorithm that decides which nodes will go into sleep mode and for how long. We have also developed mechanisms for collecting neighborhood information and for the coordination of distributed energy minimization protocols. The effectiveness of the approach is demonstrated using a comprehensive study of the performance of the algorithm over a wide range of network parameters. Another important highlight is the first mathematical and Monte Carlo analysis that establishes the importance of considering nodes within a small number of hops in order to preserve energy.

Categories and Subject Descriptors: C.3 [**Special-Purpose and Application-Based Systems**]: Real-Time and Embedded Systems; C.2 [**Computer-Communication Networks**]; C.2.3

---

Authors' addresses: Farinaz Koushanfar, ECE Department, Rice University, Houston, Texas 77005; email: farinaz@rice.edu, Abhijit Davare, David T. Nguyen, and A. Sangiovanni-Vincentelli, EECS Department, University of California at Berkeley, Berkeley, California 94720; Miodrag Potkonjak, University of California at Los Angeles, Los Angeles, California 90024.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org). © 2007 ACM 1539-9087/2007/07-ART16 \$5.00 DOI 10.1145/1275986.1275988 <http://doi.acm.org/10.1145/1275986.1275988>

[**Network Operations**]: Network Management; D.4.4 [**Communications Management**]: Network communication; D.4.7 [**Organization and Design**]: Distributed Systems, Real-Time Systems, and Embedded Systems

General Terms: Performance, Reliability

Additional Key Words and Phrases: Low power, sleeping coordination, connectivity, power management, energy management, ad-hoc networks

**ACM Reference Format:**

Koushanfar, F., Davare, A, Nguyen, D. T., Sangiovanni-Vincentelli, A., and Potkonjak, M. 2007. Techniques for Maintaining Connectivity in Wireless Ad-hoc Networks Under Energy Constraints. *ACM Trans. Embedd. Comput. Syst.* 6, 3, Article 16 (July 2007), 22 pages. DOI = 10.1145/1275986.1275988 <http://doi.acm.org/10.1145/1275986.1275988>

## 1. INTRODUCTION

Wireless ad-hoc networks (WANs) are distributed embedded systems consisting of a large number of nodes each equipped with computational, storage, communication, and, in some cases, sensing and actuating, subsystems. WANs are widely perceived as the implementation platform of choice for numerous important distributed wireless systems (DWSs), such as pervasive computing, sensor networks, WWW-based information networks, and voice and multimedia communication networks. While WANs and DWSs open up numerous, high-impact research and economic opportunities, they simultaneously pose several new challenging technical problems. There is wide consensus that among these problems, two are of dominating importance: (1) low-energy design and operation, and (2) autonomous localized operation and decision making. The most effective method for energy minimization in DWSs is to put a large percentage of nodes into sleep mode, ensuring that the sleeping nodes are not required for addressing the current needs.

Although there have been a number of efforts to determine the conditions for a node to enter sleep mode using only locally available information, while preserving the overall connectivity of the network, only heuristic answers have been presented [Xu et al. 2001; Chen et al. 2002; Cerpa and Estrin 2002]. We propose necessary and sufficient conditions to maintain network connectivity while putting a number of nodes in sleep mode. It is interesting to note that some widely used localized heuristics do not correctly preserve network connectivity. Therefore, our main objective is to propose a fundamentally sound and effective localized technique for power minimization in WANs.<sup>1</sup>

To introduce the core problem of determining when a node can go to sleep while maintaining the connectivity requirements, we present a small and illustrative example. Figures 1a and b show a part of a network. It is assumed that nodes  $M$ ,  $L$ ,  $J$ ,  $I$ ,  $H$ ,  $F$ ,  $G$ ,  $D$ , and  $N$  are connected to other nodes in the network that are not shown, for the sake of simplicity. A square indicates an awake node and a triangle denotes a sleeping node. The edge between two nodes indicates that they can communicate directly if both nodes are awake. We

<sup>1</sup>This article extends the results presented in the International Symposium on Low Power Electronic Designs [Koushanfar et al. 2003].

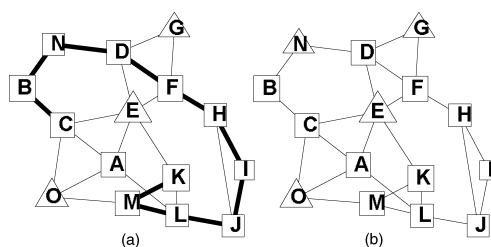


Fig. 1. Motivational figure.

assume that each node has information only about its neighbors' existence and the neighbors' awake/sleep status. In this example, our goal is to determine if node *A* can go into sleep mode. Assuming that communication has a dominating energy cost, the objective is to contact as few nodes as possible, which limits us to collect and use only local information. The key observation (as proved in Section 4) is that this task can be done locally: all that is needed is to check if there is a path through awake nodes that visits all awake neighbors of node *A* and that does not contain node *A*. In Figure 1a, one such path is highlighted. It is interesting to note that no previous approach for this task was able to correctly derive this condition. In Figure 1b, there are no such paths; thus, node *A* should not go into sleep mode. The reason is that if node *A* goes to sleep, nodes *C* and *B* will become disconnected from the rest of the network. Note that the only difference between the two networks is that in Figure 1a node *N* is awake and in Figure 1b it is in sleep mode. In the rest of the paper, we demonstrate how this observation can form the basis for an efficient method to optimize the lifetime of a network.

## 2. RELATED WORK

We survey only a small sample of important efforts, mainly those which started particular lines of research that have most influenced our work. Low-power research started its systematic rapid growth with two events: the first comprehensive set of techniques for energy modeling [Ghosh et al. 1992] and the establishment of principles for power minimization in CMOS design [Chandrakasan et al. 1992]. After that, optimization-intensive techniques have been first proposed for behavioral synthesis in statically scheduled systems [Chandrakasan et al. 1995]. Next, techniques have been proposed for power minimization in systems with dynamic behavior and for operating system-level power programmable systems [Macii et al. 1997]. In addition, variable voltage techniques started to gain in popularity [Yao et al. 1995; Hong et al. 1999]. More recently, centralized techniques for power minimization attracted a great deal of attention [Simunic et al. 2001], as have several approaches that use only local information for power management in WANs. Our goal is to exactly address these limitations.

A number of power-minimization methods using a sleeping strategy were proposed for wireless ad-hoc networks. GAF [Xu et al. 2001] is a conservative energy conservation scheme that superimposes a virtual grid proportional to

the communication radius of the nodes onto the network. Since nodes in one grid block are equal from the routing perspective, the redundant nodes within a grid block can be turned off. SPAN [Chen et al. 2002] is a distributed randomized algorithm that attempts to preserve connectivity and capacity in wireless networks, but does not provide a proof of necessary and sufficient conditions for guaranteeing this preservation of connectivity. ASCENT [Cerpa and Estrin 2002] proposes adaptive self-configuration algorithms for wireless networks that enable online tuning of the system parameters, such as sleep time of the nodes, to extend the lifetime of the overall system. STEM [Schurgers et al. 2002] suggests another power-saving strategy that does not try to preserve the capacity of the network. STEM works by putting an increasing number of nodes into sleep mode and then encountering the latency to set up a multihop path. Nodes in STEM must have an extra low-power radio called a *paging channel* that does not go into sleep mode and constantly monitors the network to wake the node up in cases of an interesting event.

The connected dominating set problem seeks to find a connected set with minimal cardinality with the property that each node in the network is either a member of the set or has a neighbor that is a member of the dominating set. The dominating connected set is often used in WANs for abstracting several important problems [Dai and Wu 2005; Han et al. 2004; Stojmenovic et al. 2002; Wu and Li 1999; Zheng and Kravets 2005]. The topology management under the energy constraints is different from the problem of finding the connected dominating set in two aspects. First, topology management is not a static problem and keeps changing the set of awake nodes. The set of awake nodes are not necessarily minimal. Second, at each point in time, the topology management under energy constraints seeks to identify the maximum number of  $k$  disjoint connected sets in an asynchronous and localized way. The disjoint sets are selected to be placed in the sleep state.

Hence, the relationship between the two problems (i.e., topology management and minimal connected dominating set) is analogous to the relationship between the maximal independent set and graph coloring. While both problems are NP-complete, it is well known that it is much easier to solve maximal independent set optimally using implicit enumeration or ILP. Also, it is well known that using maximal independent set iteratively to color a graph very often results in a very low-quality solution [Leighton 1979]. Our conjecture is that the stated observations also hold for the relationship between the connected dominating set problem and the problem of maximizing the lifetime of the multihop wireless networks. Therefore, although connected dominating set is a very important problem in its own right and has a wide application domain in wireless multihop networks, it is not applicable when the goal is to dynamically maximize the lifetime of wireless ad-hoc networks using topology management techniques.

We extend the results of the care-free sleeping theorem that we introduced before [Koushanfar et al. 2003]. We provide a more comprehensive set of evaluation results and use mathematical and Monte Carlo analysis to provide a deeper understanding and application range of the care-free sleeping theorem.

### 3. PRELIMINARIES

#### 3.1 Wireless Ad-hoc Networks

We assume an ad-hoc wireless network where there is no fixed infrastructure or centralized wireless control. Because of the limited energy sources, a node only communicates directly to other nodes within its short-range distance. If two nodes are not within the short-range distances of each other, they communicate using other nodes as intermediate relays. Although we have used the short-range communication model in our simulations, our algorithm is generic in that it can work with any communication model as long as the network is connected. The network is assumed to be fully connected initially, i.e., we assume that there is at least one path between every two nodes in the network. Once some nodes disappear from the network's topology because of power depletion, the network might become disconnected. The communication between the nodes is in the form of local broadcasting. Also, a node that is not in the sleep state periodically broadcasts a POLLING messages that contains information about that node, which includes its ID, current token group (see Subsection 6.3), location, and state (described in the energy model) along with a list of its current neighbors. Our algorithm is operating above the link layer and medium access (MAC) layers of the network, so it can take advantage of the power-saving strategies proposed for those levels.

#### 3.2 Energy Models

A node can be in one of the three following states. (1) A node is assumed to be in *sleep* state, if its radio is off and, hence, it is disconnected from the communication topology graph. (2) A listening node that has its radio on, but is not using it for communication, and is assumed to be in *idle* state. (3) A node that is communicating with other nodes using its radio is in *active* state. When a node is in active state, it can be either in transmitting or receiving state. A node can switch its state from time to time and the energy usage of a node depends on its current state. We assume an energy model where a node's energy consumption in *sleep* state is much lower than the energy consumed in other states and the *idle* state has energy consumption on the same order of magnitude as the *active* state. We further distinguish between the energy consumed for transmission and reception. We employ a random traffic model in our simulations for the communication between the nodes.

### 4. PROBLEM FORMULATION AND FOUNDATIONS

In this section, we present informal and formal definitions of the overall addressed problem—localized WANs life-time optimization. We refer to the problem as the max lifetime problem.

#### 4.1 Problem Formulation and Complexity

An ad-hoc network with  $n$  nodes that forms a connected graph is given. Initially, each node  $i$  has an amount of energy  $E_i$ . If a node is in sleep mode, it spends energy  $E_s$  per unit of time. If a node is in the active mode, it spends energy  $E_a$

per unit of time.  $E_a$  is significantly higher than  $E_s$ . Our goal is to maximize the lifetime of the network in such a way that at each point of time all awake nodes can communicate to each other using multihop communication and that each sleeping node has at least one awake neighbor. The rationale behind the first condition of this formulation is to allow all awake nodes to communicate. The rationale of the second condition is twofold. First, when a node wakes up, it can immediately access information that was sent to it through one of the awake neighbors. Second, if a node has to wake-up, the node can immediately communicate to any awake node.

Formally the max lifetime (MLT) problem can be stated in the following way.

INSTANCE: Graph  $G = (V, E)$ , function  $W(i)$ , for all  $i$  elements of  $V$ , and an integer  $T$ .

QUESTION: Is there a mapping  $F(i, t)$  to binary domain s.t. for each  $t$  each nodes  $i$  s.t.  $F(i, t) = 1$  have a path between them that consists only of nodes with property that  $F(i, t) = 1$ , and that for each node  $j$  that has  $F(j, t) = 0$ , there is an edge to a node  $k$  that has  $F(k, t) = 1$ . In addition, for each node in the graph  $\sum_t (F(i, t) \leq W(i))$ , and  $t \geq T$ .

Note that  $W(i)$  is the total energy at node  $i$  and the function  $F(i, t)$  indicates the state of node  $i$  at time  $t$ . If  $F(i, t) = 0$  the node  $i$  is sleep at time  $t$  whereas if  $F(i, t) = 1$ , node  $i$  is awake at time  $t$ . The problem can be generalized in a straightforward way to account for wake-up cost. We have proved that the MLT problem is an NP-complete problem by transforming the connected dominating set problem [Garey and Johnson 1979] into a special case of the MLT problem. For the sake of brevity and because of space limitations the proof is omitted. Therefore, it is extremely unlikely that an optimal polynomial time algorithm for the MLT problem can be developed. At the core of the MLT problem is a care-free sleep subproblem that provides the conditions that must be satisfied before a node can go into sleep mode, while the connectivity of the network is still maintained. We address this core subproblem in the next subsection.

#### 4.2 Care-Free Sleep Theorem

We term a node that is in sleep mode as a sleeping node and a node that is in transmitter/ receiving mode as an awake node. We denote a node being considered for sleep by  $A$ . We assume that when all nodes in the network are awake, they form a single connected graph. In addition, a specified number of tokens is initially generated. The number of generated tokens is smaller than the number of nodes in the network. The node that is the owner of a token is being denoted as a *token node*. Each token node initiates the procedure for evaluating the conditions for changing its operational mode from active into sleep state. Once the procedure is completed, the current token node passes its token to one of its randomly selected neighbors. Finally, we assume availability of a synchronization mechanism (one such protocol is presented in Section 6.3), which enforces that each node in the network is simultaneously considered by, at most, one procedure initiated by one of the concurrent tokens. Consequently, the considered neighborhood of the nodes that simultaneously have a token will be disjoint.

**Care-Free Sleep Theorem:** *An arbitrary node  $A$  can go into sleep if, and only if, there exists a path consisting only of awake nodes that does not contain node  $A$  and that visits all of its awake neighbors and each of its sleeping neighbors have at least one other awake neighbor.*

**PROOF.** The proof is presented using a reductio ad absurdum strategy.

If: Suppose that the conditions are satisfied. Note that the existence of a path that visits all the neighboring nodes is equivalent to and therefore implies a path between any pair of neighbors. Let us denote two arbitrary nodes in the network by  $S$  and  $D$ . Assume that there is a path  $P$  between  $S$  and  $D$  and that the path  $P$  is through node  $A$ . Obviously, path  $P$  is incoming to node  $A$  from one of its neighbors and leaving through another neighbor. We denote these neighbors as  $N_i$  and  $N_o$ . After node  $A$  goes to sleep, by the condition of the theorem, there is still a path  $P_{io}$  from  $N_i$  to  $N_o$  that does not contain  $A$ . If we replace the original subpath  $N_i \rightarrow A \rightarrow N_o$  by a subpath  $P_{io}$ , the later subpath will form a path from  $S$  to  $D$  with the remainder of the path  $P$  ( $P \setminus ((N_i \rightarrow A \rightarrow N_o))$ ). Furthermore, since each of the sleeping neighbors of node  $A$  has another neighbor awake, the second condition will be also satisfied after node  $A$  goes to sleep.

Only if: Suppose that the conditions are not satisfied. Therefore, there is a pair of neighbors  $N_i$  and  $N_o$  that do not have path between them after the node  $A$  goes to sleep or there is a neighbor node  $N_s$  in sleep mode that has  $A$  as its only awake neighbor. The first condition implies that  $N_i$  or  $N_o$  (or both) will be isolated (not connected to the rest of the network) after node  $A$  goes to sleep. The second condition implies that node  $N_s$  will not have an awake neighbor. Therefore, node  $A$  must stay awake.  $\square$

## 5. PROBABILISTIC PERFORMANCE EVALUATION

In this section, we present mathematical and Monte Carlo simulation-based analysis of small instances of networks that consist of a single node and a number of its one-hop or small number of hops neighbors. The goal of the analysis is to evaluate and justify the applications of the care-free sleep theorem as a practical tool for selecting the eligible node for entering the sleep mode by analyzing the node's small radius neighborhood. The analysis is conducted under the assumption that the awake nodes are uniformly randomly distributed.

Therefore, the purpose of the results of this section is *not* to establish the mathematical and simulation-based proofs that in uniformly distributed wireless ad-hoc networks it is sufficient with a very high probability to consider only the two-hop neighborhood to evaluate the eligibility of one node to enter the sleep state. The objective of the section is only to provide an intuition behind the expected size of the local search region. In many cases, one could expect that considering only a small immediate neighborhood is sufficient to make a correct decision for placing a particular node in the sleep state.

### 5.1 Mathematical Performance Analysis

In this subsection, we calculate the probability that a node can go to sleep when it has only two neighbors under the condition that there is a direct path

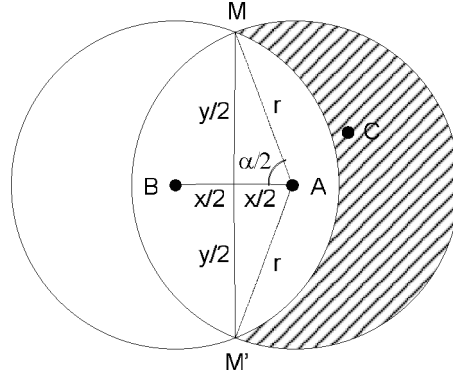


Fig. 2. Calculating the shared area between the ranges of  $A$  and  $B$ , while node  $B$  is at the distance  $x$  ( $x < r$ ) from  $A$ .

between its neighbors. Assume that we have three nodes  $A$ ,  $B$ , and  $C$ , where node  $A$  can directly communicate with both nodes  $B$  and  $C$ . Figure 2 illustrates this scenario. Under the stated assumptions, node  $A$  can only go into sleep mode if there is an edge between  $B$  and  $C$ , which implies that the distance between  $B$  and  $C$  is smaller than the communication range  $r$ . Therefore, our goal is to calculate the probability of having direct communication between nodes  $B$  and  $C$ .

We follow a uniform random distribution scheme for the placement of nodes in the area. Hence, the probability that a node exists in a given area is proportional to the size of the area. In order to calculate the probability that nodes  $B$  and  $C$  can directly communicate, we make two observations. The first is based on symmetry for all points that are the same distance  $x$  from node  $A$ . All of these points have the same probability of the targeted event. The second is that the probability that node  $B$  is distance  $x$  from node  $A$  is linearly proportional to  $x$ .

Therefore, we study the case where we move node  $B$  within the range of node  $A$ . For each placement of node  $B$ , we calculate the area shared between the communication ranges of  $A$  and  $B$ . If node  $C$  belongs to this area, there is direct communication between nodes  $B$  and  $C$ . This area corresponds to the intersection of two circles and is the nonshaded area of the circle  $A$  in Figure 2. If we denote the intersection points of the of circles  $A$  and  $B$  by  $M$  and  $M'$ , then the line  $\overline{MM'}$  has length  $y$  that can be calculated using the following equation.

$$y = 2\sqrt{r^2 + (x/2)^2} = \sqrt{4r^2 - x^2} \quad (1)$$

The angle  $\angle MAM'$  is denoted by  $\alpha$  and is calculated in the following equation.

$$\begin{aligned} \angle(\alpha/2) &= \arccos(x/2r) \\ \Rightarrow \angle\alpha &= 2 \arccos(x/2r) \end{aligned} \quad (2)$$

We consider the area that is shared between the circles  $A$  and  $B$  by  $Area(shared)$ . Using the calculated values for  $y$  and  $\alpha$ , we calculate  $Area(shared)$  for each placement of  $B$  at the distance  $x$  of the node  $A$  ( $x < r$ ) as follows.

$$Area(\widehat{MAM'}) = \alpha r^2 / 2 \quad (3)$$



$$\begin{aligned}
Area(\triangle MAM') &= xy/4 = x\sqrt{4r^2 - x^2}/4 \\
Area(shared) &= 2(Area(\widehat{MAM}') - Area(\triangle MAM')) \\
&= 2r^2 \arccos(x/2r) - x\sqrt{4r^2 - x^2}/2
\end{aligned} \tag{4}$$

The probability that an edge exists between  $B$  and  $C$ , given that node  $B$  is distance  $x$  from node  $A$  is the same as the probability of  $C$  being in the  $Area(shared)$ , given that  $\overline{AB} = x$ . Since we assumed a uniform distribution of nodes, where  $C$  can be placed anywhere within the circle  $A$  with radius  $r$ , the probability of  $C$  being in  $Area(shared)$ , given that  $\overline{AB} = x$  can be calculated in the following way.

$$\begin{aligned}
Prob(C \in Area(shared) | \overline{AB} = x) & \\
&= Area(shared)/Area(circle) \\
&= (2r^2 \arccos(x/2r) - x\sqrt{4r^2 - x^2}/2)/\pi r^2
\end{aligned} \tag{5}$$

In order to calculate the probability of an edge between  $B$  and  $C$ , we need to find the integral of the  $Prob(C \in Area(shared))$  as the node  $B$  moves within the range 0 to  $r$  inside the circle centered at  $A$ . This probability is calculated as follows.

$$\begin{aligned}
Prob(C \in Area(shared)) & \\
&= Prob(C \in Area(shared) | \overline{AB} = x).Prob(\overline{AB} = x)
\end{aligned} \tag{6}$$

Since we follow the uniform random distribution of nodes  $B$  and  $C$  within the circle, we need to take the integral with respect to  $x$ .  $dx$  to transform the uniformly distributed  $x$  and  $y$  from the Cartesian coordinates to the polar coordinates. Therefore, we calculate the targeted integral as follows.

$$\begin{aligned}
Prob(\overline{BC} \leq r) & \\
&= \int_0^r \left( \frac{2(2r^2 \arccos(x/2r) - x\sqrt{4r^2 - x^2}/2)}{\pi r^4} \right) x dx \\
&= \frac{2}{\pi r^4} \int_0^r (2r^2 \arccos(x/2r) - x\sqrt{4r^2 - x^2}/2) x dx
\end{aligned} \tag{7}$$

For calculation of the integral in the Eq. (7), we used the symbolic Integrator from the Mathematica 4.1 software package [WolframResearch 2001]. We then used the same package to calculate the value of the integral at each of the endpoints of the interval and, therefore, to obtain the probability of an edge between the nodes  $B$  and  $C$ . The result and the probabilities are shown in Eq. (8).

$$\begin{aligned}
Prob(\overline{BC} \leq r) & \\
&= \frac{2}{\pi r^4} \cdot \left[ \frac{-2r}{6\sqrt{4r^2 - x^2}} \cdot \left[ 3r^2 x \sqrt{4r^2 - x^2} \sqrt{1 - x^2/4r^2} \right. \right. \\
&\quad - 6r^2 x^2 \sqrt{1 - x^2/4r^2} \arccos(x/2r) \\
&\quad - 12r^4 \sqrt{1 - x^2/4r^2} \arcsin(x/2r) \\
&\quad \left. \left. + x^3 \sqrt{4r^2 - x^2} \text{Hypergeometric2F1} \left( 1.5, -0.5, 2.5, \frac{x^2}{4r^2} \right) \right] \right]_0^r \\
&= 0.5865
\end{aligned} \tag{8}$$

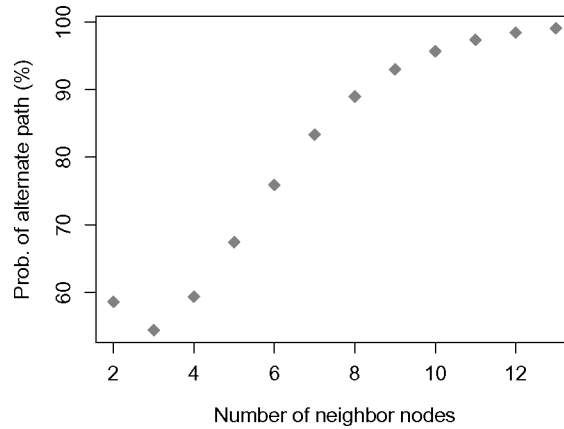


Fig. 3. The probability of an alternative path among the neighbors of the node  $A$  versus  $N$ , number of neighbors of  $A$ .

Thus, if a node has two neighbors, each placed according to the uniform random distribution, the probability of having an edge between the two neighbors is 0.5865.

## 5.2 Monte Carlo Performance Analysis

In order to study the cases where there are more than two nodes within the range of node  $A$ , we conduct Monte Carlo simulations. For each study, we generate random instances of node  $A$  with  $N$  neighbors. The neighbors are generated with a uniform random distribution within the circle of communication range of the node  $A$ . We vary the number  $N$  from 2 to 14. For each value of  $N$ , we perform 1,000,000 simulations and gather statistics on the existence of a path among all the neighbors of  $A$  that does not pass through node  $A$ . The results are shown in Figure 3. In order to gain additional insights, we extended the Monte Carlo simulation studies in 3-dimensional (3D) space. Now the communication domain of each wireless node has a spherical shape. The results for the probability of an alternative path in 3D space are shown in Figure 4. The key observation is that the changes for probabilities of paths are proportional to the probabilities obtained for the two dimensional scenarios. The only difference is that now we have smaller absolute probabilities. This can be explained by the larger average distances between the uniformly randomly distributed nodes within a sphere than within a circle of identical radius. Therefore, the key consequence is that in the 3D case it is even more important to consider the nodes outside of the communication range of a sleep candidate node to achieve the maximal lifetime of the network.

The first interesting and important observation is that for  $N = 2$ , the probability of a direct path between two neighbors is 0.5859. Since the Monte Carlo converges at rate  $\sqrt{n}$ , where  $n$  is the number of iterations, we expect that about 3.5 digits are correct after one million tries [Haber 1970; Rubinstein 1981]. Therefore, we have full agreement with the value that was derived in Eq. (8)

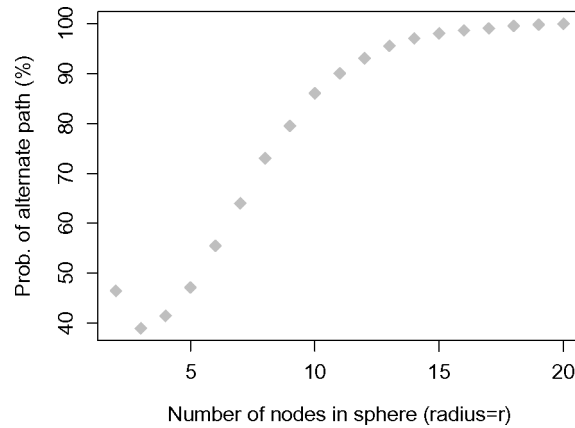


Fig. 4. The probability of an alternative path among the neighbors of the node  $A$  versus  $N$ , number of neighbor of  $A$  for 3D cases.

and full confirmation of the analysis in Subsection 5.1. The second interesting observation is that the graph in Figure 3 is nonmonotonic and the probability of having a path between three neighbors is 0.544, which is lower than the probability of having a path between two neighbors. This is unexpected, at first glance, since it seems that there is greater probability of having a path between three neighbors than between two neighbors. However, the study indicates that for three neighbors the probability is lower, because of the larger number of direct paths that are needed.

Another study that we have conducted using the Monte Carlo simulations attempts to find the required locality of the search to determine an alternative path among the neighbors. In this study, the deployment area is a circle with radius  $2r$ . Node  $A$ , which has the intention of entering sleep mode, is in the center of this circle. The communication range of node  $A$  remains  $r$ . We vary the density of the nodes in the circle with radius  $2r$ . For every number of nodes in the larger circle between 0 and 40, 1,000,000 experiments were run. For each density, we estimate the probability of finding a path that connects all the neighbors of node  $A$  and lies inside the communication range of node  $A$ . Also, for each density, we estimate the probability of finding a path among the neighbors of node  $A$  going to the larger circle. The results are shown in Figure 5. The dashed curve (Figure 5) indicates the probability of node  $A$  having no neighbors is very high for a low number of nodes in the  $2r$  circle. The curve with the square points shows the percentage of time the path among the neighbors can be found in the range of node  $A$ . The curve with the empty diamond points indicates the percentage of time we could find a path among the neighbors of node  $A$ , only by considering nodes in the outer circle (with the radius  $2r$ ). The last curve (the plain curve) is a function of the previous two curves and shows the percentage of time there are no paths among the neighbor nodes to node  $A$ , even after going to the second circle. We can observe that the probability of an alternative path in the smaller circle stays constant for the node densities between 5 and 15. The probability of having no path is a function that has a maximum of

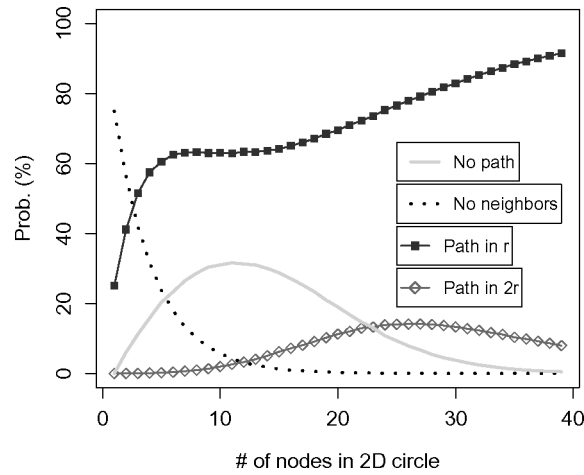


Fig. 5. The probabilities of finding alternative paths among the neighbors of the node  $A$  in the circles with radii  $r$  and  $2r$  for different node densities.

12 points in the area. This agrees with our previous results in Figure 3, since if we have 12 nodes within the larger circle, the expectation of the number of nodes in the smaller circle would be 3 (since it is one-quarter the area of the larger circle); we already found the largest probability of having no paths for three neighbors. Another interesting observation is that for numbers of nodes larger than 18 in the larger circle, there is more than a 99% probability of finding a path among the neighbors of node  $A$ , going as far as the circle with radius  $2r$ . The curves also show that the probability of finding a path in the larger circle for the number of nodes up to 27 total nodes and decreases after that point. These results clearly indicate the importance of considering additional nodes within a small hop distance to the node.

## 6. LOW-ENERGY CONNECTIVITY PROTOCOL

The network is connected as long as every awake node has at least one path to every other awake node in the network. In order to maintain the energy of the nodes and, hence, the long-term connectivity of the network, our goal is to put the maximal number of nodes in the sleep mode. The approach for assigning nodes to the sleep mode and rotating the awake/sleep nodes for a connected power-saving network are described in this section. Initially, nodes start in an *idle* state. The state changes as the nodes poll the network to discover their immediate neighborhood by exchanging routing and control messages. Also, if an unusual event occurs in the environment, the state of the corresponding nodes changes to *active* and they generate traffic to route the event to interested nodes that require the information. Node status is changed using a set of control signals.

### 6.1 Node Transition to Sleep State

In order to enable multiple threads to control the state transition algorithm, we introduce the notion of a *token*. A *token* defines the current computing node

```

Procedure Localized-Sleep-Strategy
1. At each CCN node with a token do
2. {
3.   Receive message from the non-locked nodes and record them
   in the routing table of (CCN); send message to available nodes;
4.   BEST-CANDIDATE  $\leftarrow$  (0,nil);
5.   for NODE(i) = CCN to (CCN-awake-neighbors) do
6.   {
7.     Calculate the shortest path between each pair of NODE(i)
     neighbors in k-hop neighborhood;
8.     If (care-free conditions are not satisfied for NODE(i));
9.       NODE(i) cannot go to sleep state;
10.    else
11.    {
12.      MAX-SHORTEST(i)  $\leftarrow$  max shortest path between pairs
      of neighbors to NODE(i);
13.      CRITICAL-E(i)  $\leftarrow$  node with the minimum energy on
      MAX-SHORTEST(i);
14.    }
15.    if (CRITICAL-E(i) > BEST-CANDIDATE)
16.      BEST-CANDIDATE  $\leftarrow$  (CRITICAL-E(i),i);
17.  }
18. Select the BEST-CANDIDATE node for sleep;
19. Select the next token recipient;
20. Broadcast the ID of the sleeping node; remove locks;
    pass the token to the token recipient node;
21.}

```

Fig. 6. Pseudocode for the localized strategy to select nodes for standby state.

(*CCN*) that has the control of the power-saving procedure. Since the procedure has a localized scope and a distributed nature, there can be multiple tokens operating in different places in the network. The tokens are assigned to nodes and handshaking between the tokens is done using the protocol presented in Subsection 6.3. Pseudocode of the procedure for putting a node to sleep is summarized in Figure 6. A *CCN* makes an evaluation of itself and its current neighbor nodes in order to decide which one of them is the best choice for transition to sleep mode. As shown in Figure 6, each *CCN* node makes that decision for the nodes within its local neighborhood (Step 1). Before making any decisions, the procedure has to update the information about the routes in the k-hop neighborhood of the *CCN* (Step 3). The updating is accomplished with local broadcasting to the neighbors and selective flooding to the *CCN* node. The procedure reserves a variable named *BEST-CANDIDATE* consisting of a node's ID and its critical energy at the *CCN* and assigns the value zero to critical energy and nil to ID (Step 4). While the computation is still conducted at *CCN*, the procedure evaluates *CCN* and each of its neighbors to determine the best candidate for the sleep state (Steps 5–17). Note that all the computation at the *CCN* requires only information from the k-hop routing table gathered in Step 3. The evaluation process for a *NODE*(*i*) starts by computing the shortest path between each pair of its neighbors using the standard Floyd–Warshall algorithm, without using *NODE*(*i*) on the path (Step 7). If there are pair of neighbor nodes

that do not have a path between them,  $NODE(i)$  cannot go to sleep state (Steps 8–9). Otherwise, the procedure selects the pair of neighbors to the  $NODE(i)$  with the longest path between them and saves the path information in  $MAX-SHORTEST(i)$  (Steps 10–12). Next, the node with the minimum energy on the shortest path is selected as the critical node and its information is saved in  $CRITICAL-E(i)$  (Step 13). This node is critical, since alteration of its mode can cause the alternative path to become disconnected. Since the goal is to maximize the network lifetime, if the alternative path to a sleeping node is critical with respect to energy, that node is not a good candidate for sleep state. Therefore,  $BEST-CANDIDATE$  always stores the node with the maximum  $CRITICAL-E(i)$  (Step 18). After that, the  $CCN$  has to decide the next token node from among its neighbors. The procedure selects an awake neighbor that has not received the token for the longest time (Step 19). The  $CCN$  node then broadcasts its decision about the sleep transition, frees the nodes in its  $k$ -hop neighborhood, and passes the control of the procedure to the next token node (Step 20).

## 6.2 Sleep to Idle Transition

A node that transitions into a sleep state, sets a self-timer and remains in this state for  $T_s$  time units. After the interval  $T_s$ , the node wakes up and sends POLLING messages to the nodes in its proximity. An important parameter in the effectiveness of the algorithm is the length of sleep time for at least three reasons. First, the longer the sleep time  $T_s$ , the higher the power savings. Note that the transition from sleep mode to active mode is usually very energy intensive. Second, there is an uncertainty in estimating the rate of energy consumption of the nodes on the alternative paths, because of the nonlinearity of battery discharging and also the random traffic patterns in the network, and, hence, the network might become disconnected if the length of  $T_s$  is long. Third, the shorter the length of  $T_s$ , the more flexible the network is to dynamic network topologies (e.g., handling node mobility). We determine  $T_s$  as a function of energy of the alternative paths to the node, the expected traffic energy consumption, and the length of the alternative paths to one node. For this task, we adopt the results of our experiments for tuning the parameter with respect to different metrics in the network. In our experiments, we tune  $T_s$  with respect to density, the overall lifetime of the network, and the percentage of tokens used by our sleep strategy.

## 6.3 Synchronization Protocols

Tokens are assigned to nodes during the network setup and are further updated via the POLLING mechanism. Initially, a number of nodes each randomly generate a token with a user-specified probability. In order to resolve any conflicts between the assignment of nodes to token groups, the generator node's ID (or its location) is used as the name of its group. Next, the token node invites the nodes in its neighborhood to join its group, unless they have already joined another token group. The token invitation continues to propagate further until there is no node adjacent to the nodes in the group without a token. If a token group ends up as a single node group, it randomly merges itself into one of its

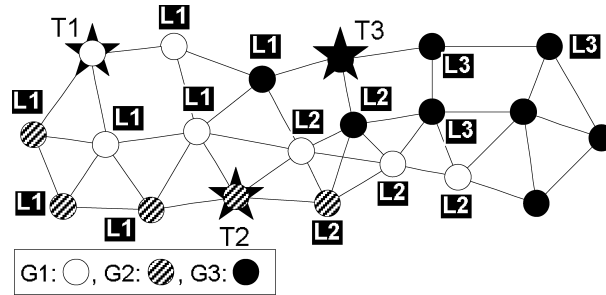


Fig. 7. An example of token synchronization between three token groups.  $T1$ ,  $T2$ , and  $T3$  denote the *CCN* nodes in the corresponding groups, while  $L1$ ,  $L2$ , and  $L3$  are the nodes locked by  $T1$ ,  $T2$  and  $T3$ , respectively.

adjacent token groups. A node stays in a token group as long as it has at least one neighbor that belongs to the same token group. Otherwise, the node sends a nil in its POLLING message and searches for a token group in its proximity to join.

At each point of time, each token makes a localized decision for the nodes in its group that may also use the information from other token groups that are within the  $k$ -hop neighborhood of the *CCN* node (Step 3 of Figure 6). This might cause a conflict between the tokens as there might be shared nodes that are within the  $k$ -hop neighborhood of two *CCN*'s from two different tokens. In order to solve this problem, our algorithm adapts a semaphore-based [Dijkstra 1968; Silberschatz et al. 2003] synchronization and resource allocation strategy, where the shared resources are only assigned to one token group at a time. During the  $k$ -hop neighborhood discovery in Step 3, the node only gathers information from unlocked nodes, which are the ones that are not in use by other tokens. Each token locks the nodes that are not locked in its  $k$ -hop neighborhood before making the decision as to which node to put to sleep in Steps 5–20. In order for the semaphore strategy to work, there should be a priority scheme among the tokens. We give priority to the token group with the smaller ID. A small example for token synchronization and locking mechanisms is shown in Figure 7. The members of the groups are shown using different colors. The nodes denoted by  $T1$ ,  $T2$ , and  $T3$  are *CCN*'s for  $G1$ ,  $G2$ , and  $G3$ , respectively. In order for the example to be small, the *CCN* nodes consider only their 2-hop neighborhood. The black labels  $L1$ ,  $L2$ , and  $L3$  show the nodes that are currently locked by  $T1$ ,  $T2$ , and  $T3$ , respectively. Hence, tokens can lock the nodes in other groups and use them in their alternative paths, although a token cannot assign state to a node that belongs to another token group.

## 6.4 Experimental Results

In this section, we present our experimental results for evaluation of our localized power saving sleeping strategy. The simulation environment consists of more than 3000 lines of simulation code in C++. Since there are a number of parameters involved in this study, we gathered a large database with more than 10,000 random instances to statistically study the performance of

Table I. Experimental Results, Showing the Relationship between Number of Nodes, Number of Tokens, Sleeping Time ( $T_s$ ), and the Lifetime of the Network

# of Nodes	% of Tokens	$T_s$ (%)	Lifetime Increase (%)
100	15	20	1.32
200	15	30	51.24
400	15	40	373.46
100	25	20	0.29
200	25	30	82.11
400	25	40	462.67
100	30	20	0.13
200	30	30	73.63
400	30	40	341.30

our algorithm. We study the lifetime of the network and measure how long the network is operational before it becomes partitioned into several disconnected parts. We show this measure as a percentage increase over the lifetime of the network without the sleeping strategy. Note that previous research in this area, such as SPAN [Chen et al. 2002] and GAF [Xu et al. 2001], have used a different metric for measuring the lifetime of the network, i.e., the fraction of nodes with nonzero energy. We did not adopt this metric, since we believe that the network becomes nonfunctional as soon as disconnections occur. We have considered a variety of k-hop neighborhood scopes for each token in Step 3 of Figure 6, although, in the studies shown here, we use only a 3-hop neighborhood. We also evaluated the new approach on both multihop networks with lossless and lossy links.

The placement of nodes in the network is generated by uniformly randomly placing the nodes in a  $1 \times 1$  unit square. The communication range of the nodes is specified by the user. Since changing the ranges and the number of points in a fixed area both generate scaled versions of the same density, we fixed the ranges to 0.13 units in these experiments and only altered the number of nodes to change the density. Initially, every node has an energy of 500J. We assume a traffic model that consists of a continuous bit rate (CBR) traffic generator that spreads the traffic randomly among 15% of the nodes. The sleeping time,  $T_s$ , is defined as a function of the percentage of the expected energy depletion from the nodes on the alternative paths to the node under consideration. The number of tokens is also calculated as a percentage of the number of nodes in the network. For the energy consumption model, we adopted the numbers from the study in Kasten [2001], from results obtained from the Digitan 2Mb/s 802.11 wireless LAN. The power consumption numbers according to this model are: transmission (1.9 W), reception (1.5 W), listening (0.75 W), and sleep (0.025 W).

Table I shows the average performance of our approach over a range of different node densities, various numbers of tokens, and different sleeping times. In Table I, the first column shows the number of nodes, the second column shows the number of tokens used by the algorithm, the third column shows the sleeping time, and the last column shows the lifetime of the network, averaged over at least 200 instances for each of the settings. As can be seen in the table,



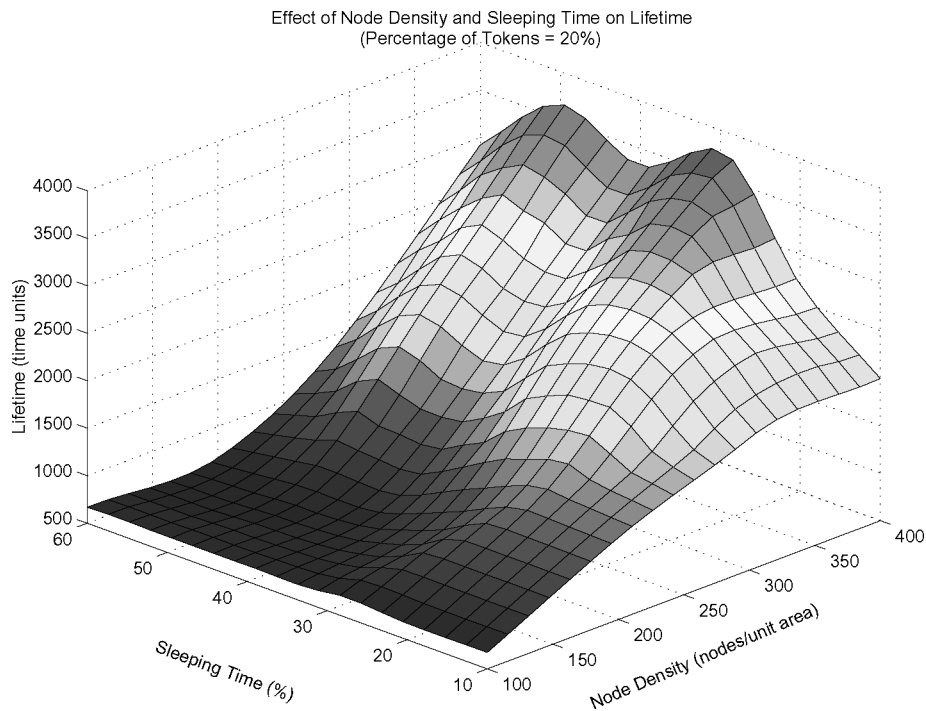


Fig. 8. Effect of the network density and sleeping time of the nodes on the lifetime of the network.

although the lifetime increases with increasing network density, there is a wide range of variation over the different parameters involved. The network lifetime also increase between 50–100% for medium to average densities (around 200 nodes in the area) and more than 200% for slightly larger densities (around 300 nodes or more).

In order to more carefully analyze these relationships and dependencies, we have used 3D plots of the lifetime of the network, based on the three parameters involved, i.e., number of tokens, sleeping time, and the density of the network. Figures 8 and 9 demonstrate these relationships. As we can see in the figures, almost all our experiments are confirming the result that our algorithm is capable of placing a large percentage of nodes to sleep as the density of the nodes increases and, as a result, an increased network lifetime. It is interesting to note that this trend is in contrast to the SPAN [Chen et al. 2002] algorithm that has lower, but comparable power savings to ours in the lower densities, but does not scale well with increasing density. Since our stopping criteria is when the network is partitioned in two or more disconnected components, existence of a large number of nodes in sleep mode can make the connectivity of the network very dependent on a few awake nodes. Once the power of those few nodes depletes, the network will disconnect even though sleep nodes can make a connection between the partitions when they wake up. For this reason, although sleeping times longer than 50% can potentially save more power, they are not practical, since they might cause temporary partitions in the network,

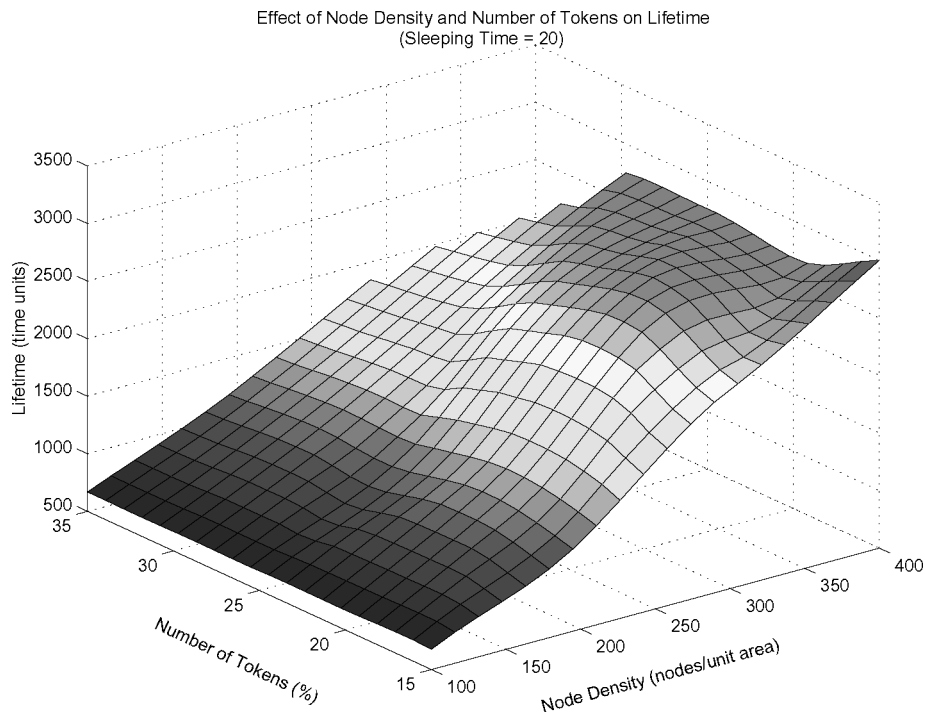


Fig. 9. Effect of network density and percentage of tokens on the lifetime of the network.

which can be seen in Figure 8. The best results for power saving are of sleeping times of around 25%.

Figure 9 shows the variation of the lifetime of the network when the density and the percentage of tokens changes. One important observation is the network saves more power for smaller percentage of tokens. This is because large number of tokens implies large number of awake nodes. For example, if the number of tokens is equal to 50% of nodes in the network, the expected number of nodes in each group is 2. At each point of time, at least one member of each token group must stay awake, which means around 50% of the nodes. However, there is a minimum number of tokens required for each density to ensure that nodes can schedule their sleeping schedule on time by receiving tokens sufficiently often. Thus, a good engineering practice is to tune the percentage of tokens to different node densities.

It is always interesting and important to compare a new approach for a particular task against already available approaches. Although there are a large number of approaches that address topology management in wireless ad-hoc networks under energy constraints, many of them have drastically different assumptions with respect to the architecture of the network, properties of the communication traffic, energy models, objectives, and constraints. For example, the SPAN system developed at MIT has as an objective of maintaining multihop connectivity between a set of source and destination nodes where all the source nodes are deployed on one side of the square, and all destination nodes are

positioned on the opposite side of the square. The nodes on the square are mobile and follow random waypoint mobility models [Broch et al. 1998].

There is at least one system that has the same assumptions and objectives as the approach presented in this paper. This approach is the geographic adaptive fidelity (GAF). It is interesting to note that it is not just possible to compare the two approaches, but one could also mathematically prove that our approach is superior to GAF. GAF imposes a virtual grid over the deployed wireless ad-hoc network. Virtual grids consists of a square that are  $r$  units long. The size of the edge of the square  $r$  is selected in such a way that any node in one square can talk to any node in a square that is placed left, right, above or below the square. The elementary geometry calculations indicates that if this condition is satisfied, the relationship between the size of the square  $r$  and the communication range  $R$  has to be  $r \leq \frac{R}{\sqrt{5}}$ .

Simple calculation indicates that under the conditions of our simulations, where the area of the node's deployment is a  $1 \times 1$  square and the communication range of 0.13, GAF would superimpose a total of 324 squares. Therefore, GAF would not even operate for the networks with less than 324 grid boxes. In reality, due to the estimates provided by the coupon-collector problem [Mosteller 1965], a significantly larger number of nodes are required just to make the GAF power-saving mechanisms applicable. The analysis indicates that unless the density of the nodes is high, there will be a significant discrepancy between the number of nodes placed in each square. On the other hand, as indicated by Table I in our experimental results section, our approach is already operational when only 100 nodes are placed in the  $1 \times 1$  square. Furthermore, our proposed technique improves the lifetime of the network by a factor of almost three times when 324 nodes are placed in the network. Therefore, we can conclude that there is often an order of magnitude or more improvement in lifetime of the wireless ad-hoc network provided by the new approach compared to GAF. Even when we restrict our attention to very dense networks, GAF induces significantly higher overhead. For example, it is easy to see that in very dense networks where our approach achieves perfect energy balancing, GAF is inferior by at least factor  $\sqrt{5}$ .

## 6.5 Experiments on Wireless Networks with Lossy Links

In this subsection, we present modifications and the analysis of the approach for maintaining connectivity of wireless ad-hoc networks under energy constraints. We start by introducing the experimental set-up that is used for collecting communication data traces. We also present a simulator that is used for the evaluation of the wireless connectivity on large networks and summarize the conceptual issues pertinent to addressing the sleeping coordination problem in wireless networks with lossy links. We introduce minor modifications to the presented care-free sleeping approach, to make the sleeping coordination strategy suitable for lossy links.

It is well known that the unit disc communication model is often not representative for the behavior of low-power wireless links. In order to evaluate the effectiveness of the proposed approach for maximizing the life time of wireless

Table II. Experimental Results of Networks with Lossy Links<sup>a</sup>

# of Nodes	% of Tokens	$T_s$	LI $\geq$ 90	LI $\geq$ 80	LI $\geq$ 70
54	5	20	5.52	5.89	6.18
54	10	25	5.76	6.02	6.50
54	15	30	5.70	6.00	6.44

<sup>a</sup>LI  $\geq$  70, LI  $\geq$  80, and LI  $\geq$  90 indicate instances where only links with reception rate above 70, 80, and 90% are considered. The numbers in columns 4–6 indicates a factor of improvement.

Table III. Experimental Results of Networks with Lossy Links<sup>a</sup>

# of Nodes	% of tokens	$T_s$ (%)	LI $\geq$ 90	LI $\geq$ 80	LI $\geq$ 70
100	15	20	27.72	33.54	49.27
200	15	30	108.81	142.31	167.52
400	15	40	517.42	543.83	556.87
100	25	20	17.88	25.65	39.90
200	25	30	110.92	138.72	162.87
400	25	40	567.02	589.81	594.46
100	30	20	10.24	18.82	19.98
200	30	30	118.92	130.89	135.76
400	30	40	522.82	546.72	560.76

<sup>a</sup>LI  $\geq$  70, LI  $\geq$  80, and LI  $\geq$  90 indicate instances where only links with reception rates above 70, 80, and 90% are considered.

networks, we utilize the traces of actually deployed wireless ad-hoc sensor networks. Specifically, we use the network test beds deployed at the LECS research labs that consist of 54 nodes embedded in the ceiling of the labs as described by Cerpa et al. [2005a]. We used TR-1000 radios set to the following parameters: Transmission (TX) power -5 dBm, packet size 200 bytes, and antenna height 0.25 ft. The network has been used for collecting data that was previously analyzed using statistical techniques to establish the properties of the ultralow-power multihop communication in wireless ad-hoc networks. The detailed presentation of the experimental set-up and the obtained and validated statistical model for generators of lossy links as a function of distance between nodes that also takes into account radio variability is presented in Cerpa et al. [2005a, 2005b].

Note that once the lossy links are considered, the full correspondence between geometric and communication neighborhoods is no longer valid. Therefore, there is need for a slight modification of the approach for maintaining connectivity in wireless ad-hoc networks under energy constraints: instead of considering all nodes in a given area for evaluation of the care-free sleep theorem, we only consider the nodes that are at most  $k$ -hop distance from the node with the token. In our experiments, we used  $k = 2$  for networks with 100 or less nodes and  $k = 3$  for networks with more than 100 nodes.

In our experimental evaluation, to form communication paths, we only considered the links that had a reception rate above a specified threshold. Specifically, we conducted three series of experiments when we considered only links that have reception rates above 70, above 80, and above 90% respectively.

For the small networks (with less than 100 nodes) the results are presented in Table II. For larger networks, the results are presented in Table III. In the

Table II, we use the actual communication ranges and actual node placements. Note that, since the communication ranges for the deployed nodes are relatively large, the improvements in the lifetime of the network are consistently high. In Table III, we place nodes in such a way that the average communication range was identical to the ones used in networks with unit disc communication model.

## 7. CONCLUSION

We have developed a new localized, distributed approach for power minimization in distributed, wireless systems under a set of communication functionality constraints. By leveraging the care-free sleep theorem that states provably optimal, necessary, and sufficient conditions for nodes to be placed in sleep mode, we have developed algorithms and synchronization protocols that efficiently exploit the tradeoff between the localized nature of operation and the effectiveness of optimization techniques. The effectiveness of the approach is demonstrated using comprehensive simulation and Monte Carlo simulation-based probabilistic analysis.

## REFERENCES

- BROCH, J., MALTZ, D., JOHNSON, D., HU, Y., AND JETCHEVA, J. 1998. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *ACM/IEEE International Conference On Mobile computing and networking (MobiCom)*. 85–97.
- CERPA, A. AND ESTRIN, D. 2002. Ascent: Adaptive self-configuring sensor networks topologies. In *IEEE Infocom*. vol. 3. 1278–1287.
- CERPA, A., WONG, J., KUANG, L., POTKONJAK, M., AND ESTRIN, D. 2005a. Statistical model of lossy links in wireless sensor networks. In *Information Processing in Sensor Networks (IPSN)*. 81–88.
- CERPA, A., WONG, J., POTKONJAK, M., AND ESTRIN, D. 2005b. Temporal properties of low-power wireless links: Modeling and implications on multi-hop routing. In *ACM International Symposium on Mobile Ad Hoc networking & Computing (MobiHoc)*. 414–425.
- CHANDRAKASAN, A., SHENG, S., AND BRODERSEN, R. 1992. Low-power cmos digital design. *IEEE Journal of Solid-State Circuits (JSSC)* 27, 4, 473–484.
- CHANDRAKASAN, A., POTKONJAK, M., MEHRA, R., RABAHEY, J., AND BRODERSEN, R. 1995. Optimizing power using transformations. *IEEE Transactions on CAD* 14, 1, 12–31.
- CHEN, B., JAMIESON, K., BALAKRISHNAN, H., AND MORRIS, R. 2002. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *Wireless Networks* 8, 5, 481–494.
- DAI, F. AND WU, J. 2005. On constructing k-connected k-dominating set in wireless networks. In *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*.
- DIJKSTRA, E. 1968. The structure of the ‘t.h.e.’ multiprogramming system. *Communications of the ACM* 18, 8, 453–457.
- GAREY, M. AND JOHNSON, D. 1979. *Computer And Intractability: A Guide To The Theory Of NP-Completeness*. W. H. Freeman, San Francisco, CA.
- GHOSH, A., DEVADAS, S., KEUTZER, K., AND WHITE, J. 1992. Estimation of average switching activity in combinational and sequential circuits. In *ACM/IEEE Design Automation Conference (DAC)*. 253–259.
- HABER, S. 1970. Numerical evaluation of multiple integrals. *SIAM Review* 12, 481–526.
- HAN, B., FU, H., LIN, L., AND JIA, W. 2004. Efficient construction of connected dominating set in wireless ad hoc networks. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*. 570–572.
- HONG, I., KIROVSKI, D., QU, G., POTKONJAK, M., AND SRIVASTAVA, M. 1999. Power optimization of variable voltage core-based systems. *IEEE Transaction on CAD* 18, 12, 1702–1714.

- KASTEN, O. 2001. Measurements of energy consumption for digital 2 mbps wireless lan module (ieee 802.11/ 2mbps). <http://www.inf.ethz.ch/kasten/research>.
- KOUSHANFAR, F., DAVARE, A., NGUYEN, D., POTKONJAK, M., AND SANGIOVANNI-VINCENTELLI, A. 2003. Low power coordination in wireless ad-hoc networks. In *International Symposium on Low Power Electronics and Design (ISLPED)*. 475–480.
- LEIGHTON, F. 1979. A graph coloring algorithm for large scheduling problems. *Journal of Research of the National Bureau of Standards* 84, 489–506.
- MACH, E., PEDRAM, M., AND SOMENZI, F. 1997. High-level power modeling, estimation, and optimization. In *ACM/IEEE Design Automation Conference (DAC)*. 504–511.
- MOSTELLER, F. 1965. *Fifty Challenging Problems in Probability With Solutions*. Addison-Wesley, Reading, MA.
- RUBINSTEIN, R. 1981. *Simulation and the Monte Carlo Method*. Wiley, New York.
- SCHURGERS, C., TSIATIS, V., GANERIWAL, S., AND SRIVASTAVA, M. 2002. Topology management for sensor networks: Exploiting latency and density. In *ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*. 135–145.
- SILBERSCHATZ, A., GALVIN, P., AND GAGNE, G. 2003. Operating system concepts: Windows xp update.
- SIMUNIC, T., L., B., ACQUAVIVA, A., GLYNN, P., AND DEMICHELI, G. 2001. Dynamic voltage scaling and power management for portable systems. In *ACM/IEEE Design Automation Conference (DAC)*. 524–529.
- STOJMENOVIC, I., SEDDIGH, M., AND ZUNIC, J. 2002. Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems* 13, 1, 14–25.
- WOLFRAMRESEARCH. 2001. Mathematica 4.1, symbolic programming. <http://www.wolfram.com/products/mathematica/index.html>.
- WU, J. AND LI, H. 1999. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Workshop on Discrete Algorithms and Methods for MOBILE Computing and Communications*. 7–14.
- XU, Y., HEIDEMANN, J., AND ESTRIN, D. 2001. Geography-informed energy conservation for ad hoc routing. In *ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*. 70–84.
- YAO, F., DEMERS, A., AND SHENKER, S. 1995. A scheduling model for reduced cpu energy. In *Symposium on Foundations of Computer Science (FOCS)*. 374–382.
- ZHENG, R. AND KRAVETS, R. 2005. On-demand power management for ad hoc networks. *Ad Hoc Networks* 3, 1, 51–68.

Received March 2005; revised September 2005; accepted April 2006