

# An Adaptive Partitioning Scheme for Sleep Scheduling and Topology Control in Wireless Sensor Networks

Yong Ding, *Student Member, IEEE*, Chen Wang, *Senior Member, IEEE*, and Li Xiao, *Member, IEEE*

**Abstract**—This paper presents an adaptive partitioning scheme of sensor networks for node scheduling and topology control with the aim of reducing energy consumption. Our scheme partitions sensors into groups such that a connected backbone network can be maintained by keeping only one arbitrary node from each group in active status while putting others to sleep. Unlike previous approaches that partition nodes geographically, our scheme is based on the measured connectivity between pairwise nodes and does not depend on nodes' locations. In this paper, we formulate node scheduling with topology control as a constrained optimal graph partition problem, which is NP-hard, and propose a Connectivity-based Partition Approach (CPA), which is a distributed heuristic algorithm, to approximate a good solution. We also propose a probability-based CPA algorithm to further save energy. CPA can ensure  $K$ -vertex connectivity of the backbone network, which achieves the trade-off between saving energy and preserving network quality. Moreover, simulation results show that CPA outperforms other approaches in complex environments where the ideal radio propagation model does not hold.

**Index Terms**—Sensor network, sleep scheduling, graph partition, radio irregularity.

## 1 INTRODUCTION

WIRELESS sensor networks consist of a large number of small battery-powered nodes that need to operate in unattended status for months. In order to sustain sensors to run for a long period of time with limited energy capacity, it is critical to save energy in sensor operations. Since wireless communication consumes the majority of energy among all the sensors' activities, reducing power consumption in communication is the most effective approach to prolong sensors' lifetime. Two strategies are usually used to minimize energy dissipation in sensor communication: 1) adjust the radio transmission power of each node or 2) schedule the wireless interfaces of sensor nodes to rotate between active and sleeping status.

Several approaches have been proposed to reduce the energy consumption of a sensor network by minimizing sensors' transmission power while maintaining the network connectivity [1], [2], [3]. However, the major energy of a sensor network is often consumed by idle listening instead of packet transmission and reception under light traffic or in a dense network. It has been broadly observed that the energy consumption of a wireless interface cannot be ignored even when it is in the listening mode. The experiment in [4] shows that the energy consumption ratio

of listening, receiving, and transmitting is 1:1.2:1.7. (The ratio is shown to be 1:1.05:1.4 in [5] and 1:2:2.5 in [6].) Therefore, energy can be further saved by reducing the time spent in idle listening of sensor nodes. In duty cycling approaches [7], [8], the wireless interface of each sensor node follows a periodic cycle of active/sleep states. However, this approach incurs additional end-to-end communication delay in the sensor network, because the intermediate node has to wait for the node at the next hop to wake up for receiving the packet.

In this paper, we adopt another sleep scheduling approach to reduce the energy consumption without causing dramatic data delivery delay in a dense sensor network. Since only a small portion of the sensors are involved in packet transmission and reception in a dense sensor network where broadcast is not frequently initiated, it will be most effective to save energy by turning off the wireless interfaces of those redundant sensors that only operate in listening status. Therefore, we can divide the sensor nodes into groups such that nodes in each group are equivalent with regard to data delivery. At each time, one node is selected from each group to operate in active radio mode (listening, transmitting, and receiving), while other nodes put themselves into sleeping mode by turning off their wireless interfaces. No matter which node is selected from each group, all the active nodes need to form a connected backbone network. If a sleeping node wants to send data, it can turn on its wireless interface temporarily to transmit the packets through the backbone network. In addition, the roles of active nodes and sleeping nodes need to be swapped once in a while to balance the power consumption among all the nodes, which prolongs the network's lifetime.

- Y. Ding and L. Xiao are with Michigan State University, 3115 Engineering Building, East Lansing, MI 48824.  
E-mail: {dingyong, lxiao}@cse.msu.edu.

- C. Wang is with the Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043. E-mail: chenw@google.com.

Manuscript received 5 June 2007; revised 26 Nov. 2007; accepted 1 Oct. 2008; published online 10 Oct. 2008.

Recommended for acceptance by X. Jia.

For information on obtaining reprints of this article, please send e-mail to: tpsds@computer.org, and reference IEEECS Log Number TPDS-2007-06-0175. Digital Object Identifier no. 10.1109/TPDS.2008.230.

The algorithm that saves energy by utilizing this node scheduling method has been proposed in ad hoc networks. GAF [9], [10] partitions the nodes based on their geographic locations. It divides the deployed area into multiple equal-size squared cells so that nodes in the same cell form a group. By assuming an ideal radio propagation model and choosing the appropriate side length of cells, it ensures that a well-connected backbone network can be formed as long as at least one node in each cell remains in active mode. However, this geographic partition suffers from several drawbacks besides its dependence on the localization infrastructure. First, as GAF uses fixed cells in its partition, it lacks the flexibility to provide different partitions that can ensure different connectivity levels of the backbone networks. Second, geographic partition methods depend on the assumption of an ideal radio propagation model, which does not always hold due to a radio's irregular transmission pattern and multipath effect in a real environment. As a result, the connectivity of the backbone network cannot be guaranteed, which causes degradation in network performance.

Motivated by these limitations, we propose a Connectivity-based Partition Approach (CPA), which divides nodes based on their measured connectivity instead of guessing connectivity by their positions. In comparison, our approach has more flexibility because it can generate partitions while ensuring  $K$ -connectivity of the backbone network. In addition, as CPA is based on measured connectivity, it can guarantee the connectivity of the backbone network even under unideal radio propagation models. Thus, CPA is more adaptive to complex environments.

The rest of the paper is organized as follows: To illustrate the basic idea of CPA, we introduce the motivation and give a formal problem description in Section 2. The detailed description of the algorithm is discussed in Section 3. In Section 4, we evaluate our proposed approach by comparing it with the GAF approach. Previous studies are summarized in Section 5, and we conclude our work in Section 6.

## 2 OVERVIEW

In this section, we first analyze several limitations of GAF that motivate us to partition the nodes based on their measured connectivity rather than their locations. After that, a formal description is given for the problem of partitioning nodes based on their connectivity graph in a large sensor network. Our solution is given in the next section.

### 2.1 Motivation

GAF identifies redundant nodes based on location information and virtual grids. It assumes an ideal radio propagation model of circular transmission range and the same radio transmission radius for all the nodes. Based on this assumption, it divides the deployed area into virtual grids with a side length of  $R/\sqrt{5}$ , where  $R$  is the radio transmission radius, so that each node associates itself with a corresponding grid according to its location. As any two nodes in neighboring grids are guaranteed to be connected because their distance is within  $R$ , a connected communication backbone can be formed by selecting only one active node

from each grid. GAF provides a partition solution for node scheduling in a large sensor network; nevertheless, it still suffers from several limitations.

Although turning off nodes can reduce the energy consumption dramatically, the change of network graph property may affect the communication performance and therefore incur more power dissipation. Vertex connectivity is a useful metric to evaluate the communication quality of the backbone network with regard to node failure and congestion. In a  $K$ -connected network, the failure of any  $K - 1$  nodes will not disrupt it into a disconnected graph. In addition, a network graph with higher vertex connectivity has a lower possibility to have bottleneck nodes of congestion because there are at least  $K$  vertex disjoint paths between any two vertices in a  $K$ -connected graph. In the extreme case, a 1-connected graph may have cut vertices that are very likely to become congestion nodes.

GAF actually provides a 4-connected backbone network for a large sensor network. As any two nodes in neighboring grids are guaranteed to be connected, each node is connected with at least four nodes in its four neighboring grids, respectively, in the backbone network. However, GAF lacks the flexibility to provide backbone networks with different vertex connectivity under different requirements. If the nodes are relatively more robust and the traffic rate is not high, a backbone network with lower connectivity is desired to achieve more energy saving by maintaining fewer active nodes. On the other hand, a backbone network with higher connectivity can cope with a higher node failure and traffic rate. Therefore, a more flexible algorithm is desired to partition the nodes into groups of appropriate size.

Another problem with GAF is that it may not work well under irregular radio propagation models. To illustrate this, we use the Degree of Irregularity (DOI) [11] as a radio propagation model. This model assumes an upper and lower bound on signal propagation range. The parameter DOI is defined as the maximum radio range variation per unit degree change in the direction of radio propagation. The DOI model used in our example is shown in Fig. 13b. The upper bound is the maximum radio transmission radius  $R$ , the lower bound is half of the upper bound, and DOI is set to 0.1.

We deploy 12 sensor nodes with a maximum radio transmission radius of  $\sqrt{5}$  uniformly into a  $2 \times 2$  area, as shown in Fig. 1a. Each edge between pairwise sensors represents a symmetric link between them based on the DOI model. In Fig. 1b, the deployed area is divided into  $2 \times 2$  grids, each of which owns three nodes according to GAF. At each period, one node in each grid turns into active status to form a communication backbone. However, there is a possibility that the backbone graph is disconnected. As the case shown in Fig. 1b, nodes 1, 4, 9, and 10 are selected from each grid to become active nodes, but they form a disconnected backbone network. The partition of GAF is invalid because the connectivity between nodes in neighboring grids is no longer ensured under the irregular radio model. They may not be able to communicate with each other even though they are within a distance of  $R$ . In addition, although the connectivity can be guaranteed by halving the grid's side length so that each two nodes in

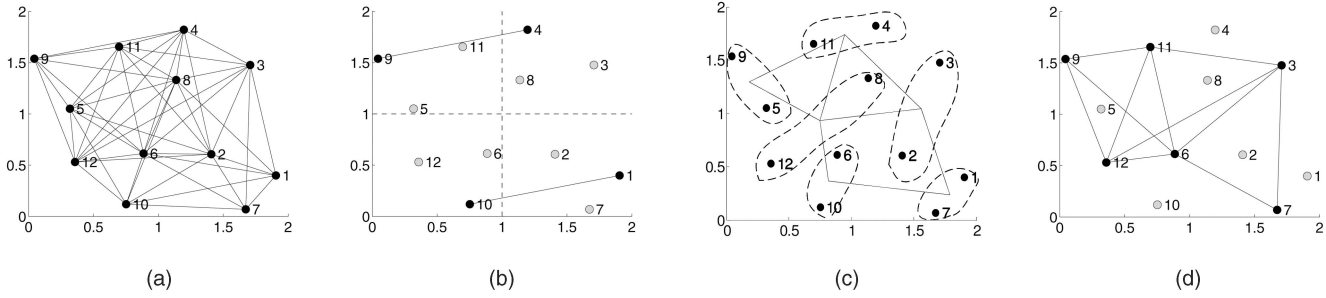


Fig. 1. GAF under irregular radio propagation models. (a) Connectivity graph. (b) GAF partition. (c) A valid partition. (d) Backbone network.

neighboring grids are within a distance of  $R/2$ , the lower bound of the radio propagation range, it is still not a good partition because it makes  $4 \times 4$  grids, which is even more than the number of nodes. On the other hand, Fig. 1c shows a valid partition. It consists of six groups where the nodes in each group are mutually connected. Each edge between two groups means that any node from one group is connected with any node in the other group. Fig. 1d shows an example of the backbone network constructed from the partition. It is obvious that the backbone formed by selecting an arbitrary node from each group is always connected, because the graph in Fig. 1c is a connected graph. This partition is valid because it is based on the measured connectivity between nodes instead of guessing connectivity from nodes' locations. Therefore, it can ensure the connectivity between neighboring groups in the partition.

In this paper, we propose CPA, which is based on the measured connectivity between nodes. CPA aims at partitioning the nodes into groups of appropriate size to ensure  $K$ -connectivity of the backbone network and at the adaptability in complex environments with irregular radio transmission models. Before presenting our design of CPA, a formal problem description is given in the following.

## 2.2 Problem Formulation

To reduce the energy consumption of communication in sensor networks, we can divide sensor nodes into groups such that only one node in each group keeps active at each snapshot while others are put into sleeping mode. The partition must satisfy the following constraints:

- Any node is within one hop away from all the other nodes in the same group. Under such a constraint, each node can be covered by the communication backbone, that is, each node is either in the backbone network if it is an active node or directly connected to the backbone network if it is a sleeping node. This also enables efficient communication among nodes in each group to switch between active and sleeping modes for load-balance purposes, because each node can communicate directly with any other node in the same group.
- The backbone network formed by active nodes at each snapshot must satisfy some connectivity properties such that it does not suffer a significant loss of communication quality as compared with the original network.

- The analysis in [12] shows that for those sensor applications where data are collected by a sink, the sensors closer to the sink always deplete their energy faster under uniform distribution of nodes, no matter what sleep scheduling is used. However, some mobility-assisted approaches such as [13] and [14] can help achieve uniform energy consumption in sensor networks. Therefore, in order to better evaluate the sleep scheduling algorithm, we assume the uniform energy consumption for sensor nodes in this paper. In order for all the groups to remain alive together as long as possible, the energy needs to be evenly distributed among groups. This is because if there is a considerable number of groups with dramatically less total energy than the others, the connectivity of the backbone network will deteriorate with the early death of these lower energy groups.
- A smaller number of groups is preferred without degrading the communication quality of the original network, because more energy conservation can be achieved by decreasing the number of active nodes at each time.

By referring to some terms in graph partition problems [15], we can formalize the problem as below.

Let  $G(V, E)$  be an undirected graph for the original sensor network, where each vertex in  $V$  corresponds to a sensor node, and each edge in  $E$  represents a symmetric communication link between the two nodes.

**Definition 1.** Given a graph  $G(V, E)$ , we can partition  $V$  into  $N$  disjoint sets  $A_1, A_2, \dots, A_N$  such that the induced graph of each vertex set, denoted by  $G[A_i]$  ( $i \in \{1, 2, \dots, N\}$ ), is a clique. We can encode this partition by a symmetric  $N \times N$  matrix  $M$ , where

- $M_{i,i} = 1$  ( $i \in \{1, 2, \dots, N\}$ ), representing that each  $G[A_i]$  is a clique, and
- for  $M_{i,j}$  ( $i, j \in \{1, 2, \dots, N\}, i \neq j$ )
  - $M_{i,j} = 2$  if  $A_i$  and  $A_j$  are completely adjacent, that is, any vertex in  $A_i$  is connected with any vertex in  $A_j$ ,
  - $M_{i,j} = 1$  if  $A_i$  and  $A_j$  have arbitrary connections, that is, there exists some vertex in  $A_i$  connected with some vertex in  $A_j$ , but  $A_i$  and  $A_j$  are not completely adjacent, and
  - $M_{i,j} = 0$  if  $A_i$  and  $A_j$  are completely nonadjacent, that is, no vertex in  $A_i$  is connected with any vertex in  $A_j$ .

This partition is called an *M-partition* of  $G$ , and  $N$  is the size of the partition.

**Definition 2.** Given an  $M$ -partition  $P = \{A_1, A_2, \dots, A_N\}$  of  $G(V, E)$ , we select an arbitrary vertex  $x_i$  from  $A_i$  ( $i = 1, 2, \dots, N$ ). Let  $X = \{x_1, x_2, \dots, x_N\}$ . The induced graph  $G[X]$  is called a **backbone graph** of  $G$  under  $M$ -partition  $P$ . In the rest of the paper, we also call it the backbone network without confusion.

Note that the  $M$ -partition implies some connectivity property of the backbone graph. Let  $B$  denote a backbone graph of  $G$  under an  $M$ -partition  $P = \{A_1, A_2, \dots, A_N\}$ . If  $M_{i,j}$  is 2, then  $x_i$  and  $x_j$  (the two vertices selected from  $A_i$  and  $A_j$ , respectively) are guaranteed to be connected; if  $M_{i,j}$  is 0, they must be disconnected; otherwise, they may or may not be connected.

**Definition 3.** Given an  $M$ -partition  $P = \{A_1, A_2, \dots, A_N\}$  of  $G$ , we define a graph  $H(S, T)$ , where each vertex  $s_i \in S$  corresponds to  $A_i \in P$  ( $i = 1, 2, \dots, N$ ), and  $(s_i, s_j) \in T$  if and only if  $M_{i,j} = 2$ . We call  $H$  the **2-induced graph** of  $P$ .

We are interested in  $H$  because it reflects the minimum connectivity property of any backbone graph  $B$  of  $G$ . If  $s_i$  and  $s_j$  are connected in  $H$ , then  $A_i$  is completely adjacent with  $A_j$ . Thus, in any backbone graph  $B$  of  $G$ ,  $x_i$  and  $x_j$  (two arbitrary vertices selected from  $A_i$  and  $A_j$ , respectively) are guaranteed to be connected. In other words,  $E(H)$  is a subset of  $E(B)$ . Therefore, suppose  $\kappa(H) = K$ ; then, we have  $\kappa(B) \geq K$ , where  $\kappa$  denotes the vertex connectivity of the corresponding graph.

Let  $l$  be a label on  $V$  of the original network  $G(V, E)$ , where  $l(v)$  ( $v \in V$ ) is the amount of energy in the sensor node  $v$ ; then, the total energy of the sensor network is  $E_{total} = \sum_{v \in V} l(v)$ . We can also derive another label  $g$  on the  $M$ -partition  $P = \{A_1, A_2, \dots, A_N\}$  of  $G$ ,  $g(A_i) = \sum_{v \in A_i} l(v)$ , for each  $A_i \in P$ , which represents the total energy in each group of the partition.

**Problem formulation (constrained minimum-size  $M$ -partition problem).** Given a graph  $G(V, E)$ , which represents the original sensor network, and a label  $l$  on  $V$ , which represents the amount of energy in each sensor node, find a minimum-size  $M$ -partition  $P^*$  of  $G$  such that 1)  $\kappa(H) \geq K$ , where  $H$  is the 2-induced graph of  $P^*$ , and  $K$  is the minimum vertex connectivity required by the backbone network, and 2)  $(1 - \delta) \frac{E_{total}}{N} \leq g(A_i) \leq (1 + \delta) \frac{E_{total}}{N}$  for each  $A_i \in P^*$ , where  $N$  is the size of  $P^*$ , and  $0 \leq \delta < 1$  is the unbalanced factor.

As  $G[A_i]$  ( $A_i \in P^*, i = 1, 2, \dots, N$ ) is a clique, each node is within one hop away from all the other nodes in the same group of the partition. The connectivity property of the backbone network can be guaranteed by the first constraint of the problem, and the balanced energy distribution can be satisfied by the second constraint. Moreover, the optimization nature of this problem requires the most efficient partition with regard to energy saving.

**Theorem 1.** The constrained minimum-size  $M$ -partition problem is NP-hard.

**Proof.** Consider a problem that is less hard than the formulated problem. We remove the two constraints

from the formulated problem, and the resulting problem becomes finding a minimum-size  $M$ -partition of  $G$ , which we denote by the *smallest  $M$ -partition problem*. We then show that the *clique problem*, which is NP-complete, can be reduced to the *smallest  $M$ -partition problem* in polynomial time. (The clique problem is the problem of determining whether a graph contains a clique of at least a given size  $k$ .) As  $M$ -partition requires that  $G[A_i]$  is a clique for each  $A_i \in P$ , the *smallest  $M$ -partition problem* is equivalent to finding all maximal cliques in  $G$ . Suppose the set of all maximal cliques of  $G$  has been found. Then,  $G$  contains a clique of size at least  $k$  if and only if there exists a maximal clique of size at least  $k$ . This reduction only needs polynomial time. Thus, the *smallest  $M$ -partition problem* is NP-hard. Therefore, the constrained minimum-size  $M$ -partition problem is also NP-hard.  $\square$

### 3 CPA DESIGN

In this section, we propose CPA to approximate a good partition for this problem. We want to find an  $M$ -partition of which the size is as small as possible, while satisfying that any backbone graph under the  $M$ -partition is at least  $K$ -connected and the energy distribution in different groups is as even as possible. The proposed algorithm is a distributed heuristic algorithm, where only local computation is involved. Some of our preliminary results have already been published in [16]. CPA is a distributed iterative process. It starts from the initial partition where each node forms a unique group. CPA continuously merges two groups into a larger one until further merging will break the constraints of the problem.

In CPA, there are two kinds of nodes in each group: ordinary nodes and a head node. Each kind of node maintains its node ID and associated group information, including its group ID, the IDs of other group members, and the ID of the head node in its group. One head node is selected from each group to maintain some additional information on the connectivity between its group and the neighboring groups in the current  $M$ -partition. Let  $N_l(A_i)$  be the set of neighboring groups that are connected with group  $A_i$  through  $l$ -value edges in the current  $M$ -partition, i.e.,  $N_l(A_i) = \{A_j | M_{i,j} = l\}$ . Thus, each head node of group  $i$  will store  $N_1(A_i)$  and  $N_2(A_i)$ , which are the set of neighboring groups having arbitrary connection with group  $i$  and completely adjacent with group  $i$ , respectively.

CPA starts from the initial partition of one node in each group. Let  $A_i$  denote the group formed by node  $v_i$ . Consequently,  $v_i$  acts as the head node and stores group connectivity information  $N_1(A_i)$  and  $N_2(A_i)$ , where  $N_2(A_i)$  is the set of groups  $A_j$  whose node  $v_j$  is connected with node  $v_i$ , and  $N_1(A_i)$  is empty because any two groups are either completely adjacent or completely nonadjacent in the initial partition. CPA goes through a group merging process iteratively before it gets to the final partition.

#### 3.1 Group Merging

In the group merging process, the head nodes of each two completely adjacent groups exchange group connectivity information to decide whether their groups should merge. Only completely adjacent groups can merge so that the new

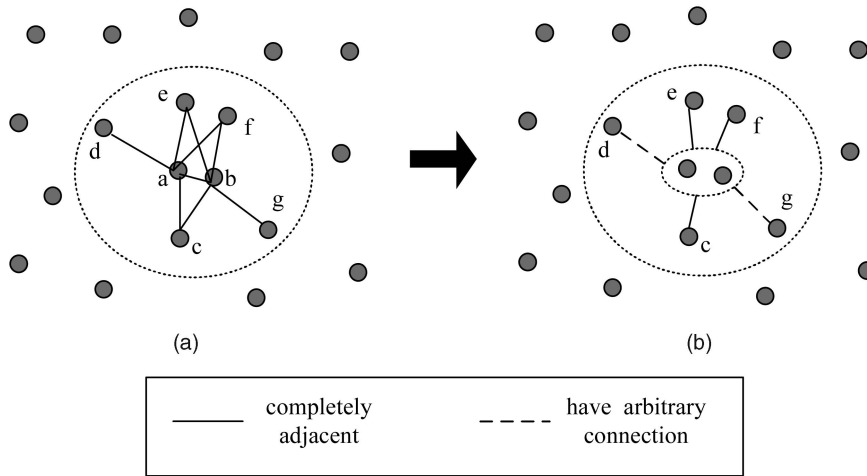


Fig. 2. CPA group merging process. (a) Two groups  $a$  and  $b$  to merge. (b) The two groups have been merged.

group is also a clique. Suppose  $A_i$  and  $A_j$  are two completely adjacent groups in the current  $M$ -partition. Let  $A_{ij}$  be the new group obtained by merging  $A_i$  and  $A_j$ . The group merging process first updates the group information in each node of  $A_{ij}$  and keeps only one head node to maintain the group connectivity information in  $A_{ij}$ . Then, the new group and the neighboring groups of  $A_i$  and  $A_j$  update their group connectivity information based on the following rules: 1) for each group  $A' \in N_2(A_i) \cap N_2(A_j)$ ,  $A_{ij}$  and  $A'$  are completely adjacent (edge value of 2), 2) for each group  $A' \in N_0(A_i) \cap N_0(A_j)$ ,  $A_{ij}$  and  $A'$  are completely nonadjacent (edge value of 0), and 3) otherwise,  $A_{ij}$  and  $A'$  have arbitrary connections (edge value of 1). Therefore, an updated  $M$ -partition is formed. Fig. 2 illustrates the process of merging two groups into a larger group. In Fig. 2a,  $a$  and  $b$  are two completely adjacent groups, that is, any node in  $a$  is connected with any node in  $b$ . When the two groups merge, as shown in Fig. 2b, as groups  $c$ ,  $e$ , and  $f$  are completely adjacent to both  $a$  and  $b$ , each of them is completely adjacent to the new merged group. On the other hand, groups  $d$  and  $g$  only have arbitrary connection with the new group, which are illustrated by the dashed lines in the figure, because  $d$  is not completely adjacent with  $b$  and  $g$  is not completely adjacent with  $a$ .

Contentions may occur when multiple neighboring groups want to merge simultaneously. We resolve this by imposing a randomized backoff delay on the time when the two groups announce their willingness to merge. If no contention is observed at the end of the delay, the two groups about to merge will announce their decision to all of their neighboring groups. Otherwise, they will reevaluate the backoff delay based on the updates from other group merges.

The goodness of the final partition depends on the sequence of group merge. We consider several factors for deciding which two groups are preferred to be merged first in the current partition in order to arrive at a good partition eventually. These factors can be reflected as a utility function in the randomized backoff delay so that higher priority groups will announce their intentions to merge with a shorter time of delay:

- For any two groups  $A_i$  and  $A_j$  in the current partition, let  $P = |N_2(A_i) \cap N_2(A_j)|$  and  $Q = |N_2(A_i) \cup N_2(A_j)|$ ;

then,  $C = P/Q$  indicates the level of equivalence between  $A_i$  and  $A_j$ . The two groups with a higher  $C$  value will be given higher priority in the group merging process. Specifically, at the starting phase of the algorithm, where each node constitutes a single group, nodes with exactly the same set of neighbors will be merged first, because these nodes are exactly equivalent with regard to relaying data.

- Let  $g(A_i)$  denote the energy in group  $A_i$ . We want the total energy to be evenly distributed in each group so as to maximize the network's lifetime. For any two groups  $A_i$  and  $A_j$ , let  $D = [g(A_i) + g(A_j)]/E_{total}$ , where  $E_{total}$  is the total energy of all the sensor nodes in the network; then, we will give pairwise groups with a lower  $D$  value higher priority in the group merging process.

Therefore, each two completely adjacent groups can be assigned with a utility value  $U = k_1(1 - C) + k_2D$ , where  $k_1$  and  $k_2$  are coefficients. The backoff delay for each pair of groups is set to be proportional to  $U + R$ , where  $R$  is a random value uniformly distributed in  $[0, 1]$ , which is used to resolve contentions among pairwise groups with the same utility value. As a result, the appropriate assignment of backoff delay enables pairwise groups with a lower utility value to merge first, as well as resolving contentions in the group merging process.

### 3.2 Guarantee of Connectivity

As we have discussed in the previous section, the  $K$ -connectivity of the backbone network can be guaranteed by the  $K$ -connectivity of the 2-induced graph of  $M$ -partition. However, the group merging process may disrupt this connectivity. When  $A_i$  and  $A_j$  merge, the number of completely adjacent groups for  $A_{ij}$  may decrease, and this number for each  $A' \in N_2(A_i) \cup N_2(A_j)$  will decrease by one. For example, when the groups  $a$  and  $b$  merge in Fig. 2, the number of completely adjacent groups for the new group decreases by two, while the number decreases by one for each group of  $c$ ,  $d$ ,  $e$ ,  $f$ , and  $g$ . In order to ensure the  $K$ -connectivity of the 2-induced graph of  $M$ -partition, we apply a result on the property of random geometric graphs that was published in [17].

Random geometric graphs (with parameters  $n$  and  $r$ ) are constructed by dropping  $n$  points randomly uniformly into the unit square (or more generally on a  $d$ -dimensional euclidean space) and adding edges to connect any two points distant at most  $r$  from each other. Therefore, a large-scale sensor network with an ideal radio propagation model can be modeled as a random geometric graph. Penrose [17] proved that “For a random geometric graph  $G(n, r)$ , let  $r_n$ , (respectively  $s_n$ ) denote the minimum  $r$  at which the graph, obtained by adding an edge between each pair of points distant at most  $r$  apart, is  $K$ -connected (respectively, has minimum degree  $K$ ). Then  $r_n = s_n$  with probability approaching 1 as  $n$  tends to infinity.” In other words, for a random geometric graph with a large number of nodes, the network reaches  $K$ -connectivity at the same time when its minimum degree reaches  $K$ .

We assume that the backbone graph can be approximated as a random geometric graph, then its  $K$ -connectivity can be guaranteed by ensuring its minimum degree of  $K$ . As the 2-induced graph of  $M$ -partition is a subgraph of each backbone graph, its minimum degree is no greater than the backbone graph. Therefore, we can guarantee the  $K$ -connectivity of the backbone graph by ensuring the minimum degree of  $K$  in the 2-induced graph of  $M$ -partition.

In  $M$ -partition, we refer “2-degree of  $A_i$ ” to the number of groups that are completely adjacent to  $A_i$ . Each group keeps track of its 2-degree value during the group merging process, which is used to decide whether pairwise groups should be merged. If a group merge may cause the 2-degree of some group to drop below  $K$ , then these two groups will give up their intention to merge. The group merging process will be terminated when no groups can be merged.

The theoretical proof above demands the ideal radio propagation model to form a sensor network into a random geometric graph. However, our simulations in Section 4 show that under the irregular radio propagation model, we can still form a  $K$ -connected backbone network with high probability by ensuring the minimum degree of  $K$  in the 2-induced graph of the  $M$ -partition. Compared with geographic partition methods, CPA can preserve the network’s communication quality much better in irregular radio environments.

### 3.3 Centralized and Distributed Implementation

To better illustrate the idea of CPA, we first show the centralized version in Algorithm 1. During the group merging process, for each pair of completely adjacent groups,  $f_{ij}$  denotes the priority of merging these two groups, and  $t_{ij}$  indicates whether the minimum 2-degree of the partition will drop below  $K$  (the connectivity requirement of the backbone network) if we merge them. In each step of the iteration, we will merge the two groups with the highest priority (smallest  $f$  value) that will not break the connectivity constraint ( $t$  value is *false*). The merging process continues until the merge of any pair of groups will break the constraint.

#### Algorithm 1. Centralized Partitioning Algorithm

- 1: **for all**  $v_i \in V$  **do**
- 2:  $A_i = \{v_i\}$

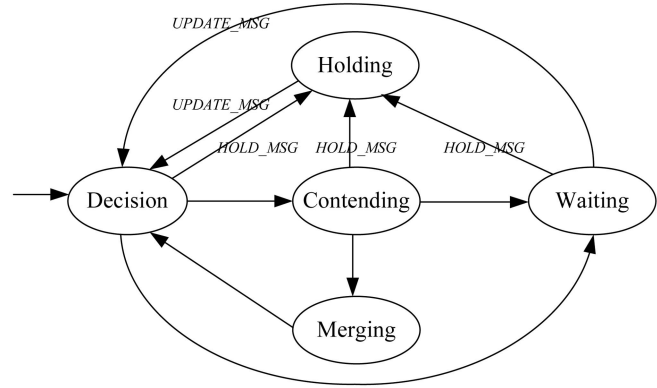


Fig. 3. Distributed partitioning algorithm.

- 3:  $deg_2(A_i)$  = degree of  $v_i$  in  $G$
- 4: **end for**
- 5: **repeat**
- 6: **for all**  $A_i, A_j \in A$ , where  $A_i$  and  $A_j$  are completely adjacent **do**
- 7:  $C = |N_2(A_i) \cap N_2(A_j)| / |N_2(A_i) \cup N_2(A_j)|$
- 8:  $D = [g(A_i) + g(A_j)] / E_{total}$
- 9:  $f_{ij} = k_1(1 - C) + k_2D$
- 10:  $d^* = |N_2(A_i) \cap N_2(A_j)|$
- 11:  $mindeg = \min\{d^* \cup \{deg_2(s) - 1 \mid s \in N_2(A_i) \cup N_2(A_j)\}\}$
- 12:  $t_{ij} = (mindeg < K) ? \text{true} : \text{false}$
- 13: **end for**
- 14: Choose  $(A_x, A_y)$  with the smallest  $f$  value among all  $(A_i, A_j)$  where  $t_{ij} = \text{false}$
- 15: **for all**  $s \in N_2(A_x) \cup N_2(A_y)$  **do**
- 16:  $deg_2(s) = deg_2(s) - 1$
- 17: **end for**
- 18:  $A_x = A_x \cup A_y$ ,  $deg_2(A_x) = |N_2(A_x) \cap N_2(A_y)|$
- 19:  $A = A - \{A_y\}$
- 20: **until**  $t_{ij} = \text{true}$  for all  $(A_i, A_j)$

With some modifications, the algorithm can be implemented in a distributed fashion. Fig. 3 illustrates the state transition diagram of each head node. Initially, each node constitutes a group (or is a head node). At the start, each node broadcasts an *UPDATE\_MSG* to its neighbors containing its group information. In the *Decision* state, the head node calculates the  $f$  value and  $t$  value for each of its completely adjacent groups and selects the group with the smallest  $f$  value (highest priority to merge) where the  $t$  value is *false* (the merge will not break the connectivity constraint). Let  $A_x$  be the current group and assume that the best group to merge with is  $A_y$ . If any completely adjacent group of  $A_x$  or  $A_y$  is in the *Merging* or *Holding* state, then the head node will go into the *Waiting* state, because the merge of  $A_x$  and  $A_y$  will collide with the merge of some other groups. Otherwise, the head node will go into the *Contending* state. In this state, the head node will send a *MERGE\_REQ* message to  $A_y$ , expressing its willingness to merge with  $A_y$ , with a backoff delay as a function of  $f$ . If the contention succeeds ( $A_x$  receives a *MERGE\_ACK* message from  $A_y$ ), it goes into the *Merging* state, and otherwise ( $A_x$  receives a *MERGE\_NAK* message from  $A_y$ ), it enters the *Waiting* state. In the *Merging* state, it first broadcasts a

*HOLD\_MSG* to the neighboring groups of  $A_x$  and  $A_y$  to put them into *Holding* state and then updates its group information. After the merge is finished, it broadcasts an *UPDATE\_MSG*, which contains the information of the newly merged group, to its neighboring groups. If a head node has not yet entered the *Merging* state, it will enter the *Holding* state when it receives a *HOLD\_MSG* from neighboring groups. When a head node is in the *Holding* state or *Waiting* state, it will enter the *Decision* state again once it has received an *UPDATE\_MSG*.

### 3.4 Probability-Based Partition

In the previous partition algorithm, we only focused on the completely adjacent groups, that is, we ensure that each group has at least  $K$  completely adjacent groups in the partition. Thus, in the backbone network formed by activating one node from each group, each node has at least  $K$  neighbors, which maintains the network's connectivity. However, we did not make use of the groups having arbitrary connections, especially when one group has most nodes connected with most nodes in the other group. In this section, we try to utilize this information in the optimization problem to further reduce the partition size.

Let  $A_i$  and  $A_j$  be two groups with size  $n_i$  and  $n_j$ . We can evaluate the connectivity between these two groups by the connectivity probability  $P_{ij} = \frac{m}{n_i \times n_j}$ , where

$$m = |\{(x, y) | x \in A_i \cap y \in A_j \cap (x, y) \in E\}|.$$

Thus, if  $A_i$  and  $A_j$  are completely adjacent, then  $P_{ij} = 1$  because  $m = n_i \times n_j$ . If  $A_i$  and  $A_j$  are completely nonadjacent, then  $P_{ij} = 0$  because  $m = 0$ . Otherwise,  $0 < P_{ij} < 1$ .

With slight modifications, we can utilize the connectivity probability in the previous algorithm. In Algorithm 1, we keep track of the number of completely adjacent groups for each group. Instead, we can count the number of groups with connectivity probability more than  $P$  for each group. Thus, we are able to guarantee that each group has at least  $K$  groups with connectivity probability more than  $P$  in the final partition.

One drawback of the probability-based approach is that the  $K$ -connectivity of the backbone network cannot be guaranteed for sure if  $P < 1$ . However, by loosening the requirement on connectivity, we save more energy because of the reduced final partition size, while ensuring the  $K$ -connected backbone with high probability at the same time. We will show this in the simulation.

### 3.5 Load Balancing Energy Usage in Groups

As all the nodes in the network are equally important, running a node in active status until its energy is depleted is not an appropriate energy usage strategy. In order to prolong the lifetime of each node, the nodes in each group need to switch between active and sleeping status periodically so that all nodes remain alive together for as long as possible.

Assume that all the nodes in each group can be synchronized. In order to elect an active node, each group member broadcasts a message to the whole group stating its willingness to become active. Each node waits for a certain time delay before its announcement. The earliest announcement will suppress the others so that the

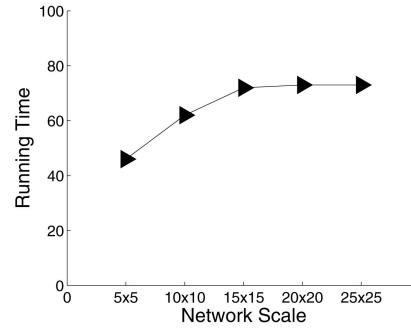


Fig. 4. Runtime of CPA (node density = 5).

corresponding node will become the active node in the group. The time delay for each node is set to be inversely proportional to its residual energy. Therefore, the node with the maximum residual energy will be selected. Then, the selected active node informs other nodes of the time it will remain active, after which all the nodes need to reselect an active node again within the group.

## 4 PERFORMANCE EVALUATION

We evaluate our schemes under uniform deployment of sensors. The simulation is based on the energy consumption model observed in [4], that is, the ratio of energy consumed in listening, receiving, and transmitting status is 1:1.2:1.7. The initial energy level of each node is set to 500, which means that the node will remain alive for 500 units of time in the listening status. According to the assumption in Section 2.2 that the energy consumption is uniform over all the sensor nodes with the mobility-assisted approaches helping to collect data, we simulate the energy consumption in the sensor network by an equivalent scenario. In each time slice, we randomly select 20 traffic nodes, which send and receive packets between each other. In addition, we use load-balanced energy-aware routing [18] in the backbone network. One slight modification we make is that we use the total residual energy of the group to denote the residual energy of the corresponding active node in the load-balanced route decision.

We perform the simulation in Matlab. The energy consumption is calculated based on the changing status of each node and the energy consumption ratio for each status. In our evaluation of the sensor network's lifetime, we do not take the energy consumed in the partition process into consideration, because it runs only once at the deployment phase of the sensor network so that it only consumes a trivial portion of the network's total energy.

In this section, we will first analyze the scalability of our partition algorithm and then evaluate the partition of CPA in comparison with GAF under both the ideal radio transmission model and the irregular radio transmission models.

### 4.1 Scalability of CPA

As CPA is a distributed algorithm where only local computation is involved, it is scalable with the network size. This can be seen in Fig. 4. We run the distributed partition algorithm on networks with the same density of sensors but different ranges. In all configurations, the density is set to five sensors per square unit, and the radio

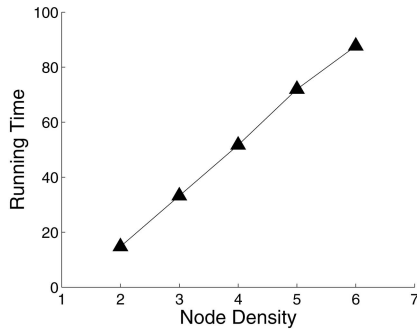


Fig. 5. Runtime of CPA under different densities.

transmission range is set to  $\sqrt{5}$ . We try networks with an area of  $5 \times 5, \dots, 25 \times 25$ , which are shown in the  $x$ -axis of the figure. Assume that each group merge needs one unit of time. We use a relative scale to measure the runtime. The  $y$  value of each point means that the runtime of the distributed algorithm is comparable to the time used for sequentially merging groups  $y$  times. We can observe from the figure that the runtime tends to converge with the increase of network scale.

In Fig. 5, we run the partition algorithm on a network with the range of  $10 \times 10$  but with different node densities (two,  $\dots$ , six nodes per square unit). We can observe that the runtime increases linearly with the node density approximately. This is because the radio transmission range remains unchanged though the node density increases. When two groups are going to merge, all the groups within their communication range cannot merge due to collision. Therefore, increasing the node density does not increase the pairs of groups that can merge simultaneously.

Fig. 6 illustrates the overhead of CPA under different network scales while the node density remains unchanged at five. We evaluate the overhead by counting the average number of messages transmitted per node. In the figure, we can observe that the average number of messages transmitted per node converges with the increasing network size. This is because the CPA algorithm only requires local information at each node. The overhead is reasonable, because CPA is not running frequently. Only when there is dramatic environmental change or there are a considerable number of groups dead will CPA be reexecuted.

#### 4.2 CPA under the Ideal Radio Propagation Model

In this section, we evaluate the partitions of CPA and GAF under the ideal radio propagation model in a  $10 \times 10$  area. We first set the node density to five, that is, we deploy 500 sensors uniformly in the area. We compare CPA and

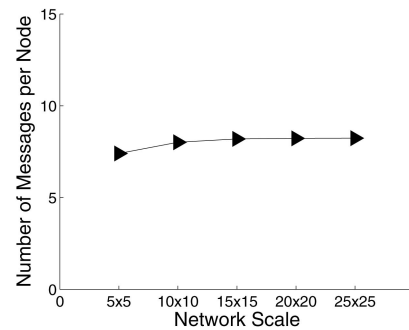


Fig. 6. Overhead of CPA (node density = 5).

GAF based on the network lifetime and the connectivity of backbone networks. After that, we repeat our simulation for different node densities and network scales.

In the ideal radio propagation model, the radio transmission range is the same in different directions. In our simulation, the radio transmission radius  $R$  is set to  $\sqrt{5}$ , which is the same with GAF. GAF uses squared cells with a length of  $R/\sqrt{5} = 1$  to partition the deployed area; thus, all the nodes are divided into 100 groups. We also run CPA, which is based on the connectivity between nodes instead of their locations under the same experiment setting. CPA is executed with different values for parameter *mindeg*, which controls the minimum degree of the 2-induced graph of the final partition. CPA guarantees that the backbone network generated based on this partition will be *mindeg*-connected.

The partition results are shown in Table 1. For CPA, the number of groups increases with the parameter *mindeg*, because more active nodes are needed each time in order to ensure higher connectivity of the backbone network. GAF ensures that each node is connected with at least four nodes in its four neighboring cells in the backbone network. Therefore, we can regard GAF as comparable to CPA(*mindeg* = 4). As shown in the table, CPA(*mindeg* = 4) partitions the nodes into 91 groups, which is fewer than that of GAF. This indicates that CPA can identify redundant nodes more sufficiently than GAF. As discussed in previous sections, it is preferable to have the total energy evenly distributed in the groups so as to prevent the early death of some groups, which may disrupt the connectivity of the backbone network. We assume that each node has the same initial energy when deployed, so the standard deviation of group size can be an indication of how evenly the total energy is distributed in the groups. Table 1 shows the standard deviation of group size for each partition result. It shows that our approach can divide energy as evenly in groups as GAF under random uniform distribution of nodes.

TABLE 1  
Partitions of GAF and CPA

Partition Approach	CPA (mindeg=2)	CPA (mindeg=3)	CPA (mindeg=4)	GAF	CPA (mindeg=5)	CPA (mindeg=6)
Number of Groups	71	84	91	100	106	116
Average Group Size	7.0	6.0	5.5	5.0	4.7	4.3
Standard Deviation on Group Size	0.92	0.77	0.79	0.73	0.67	0.63

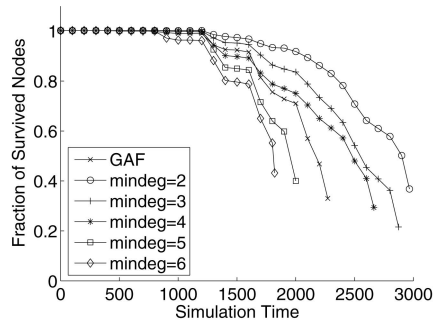


Fig. 7. Comparison of network lifetime.

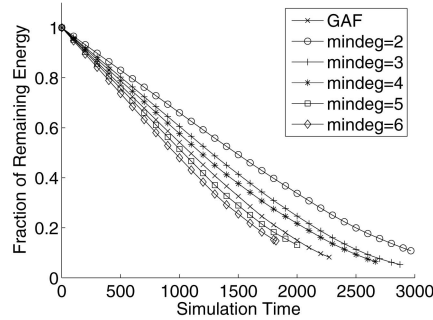


Fig. 8. Comparison of energy consumption.

We then evaluate the lifetime of the network when the partition results are applied. Each group keeps only one active node each time to form a backbone network. The nodes in each group balance the energy usage by reselecting the active node periodically, which may change the topology of the backbone network. We refer to lifetime as the time when the backbone network formed by active nodes turns out to be disconnected. If no sleeping schedule scheme is used, that is, all nodes keep active until death, the network lifetime will be less than 500. Fig. 7 illustrates the network lifetime where different partitions are adopted. As shown in the figure, the fraction of surviving nodes decreases with time, and the simulation stops when the backbone network becomes disconnected. The partition of CPA( $\text{mindeg}=2$ ) achieves the longest network lifetime (around 3,000), because it keeps the fewest number of active nodes each time. The network lifetime decreases for partitions of CPA with higher  $\text{mindeg}$  values, which can, however, ensure better connectivity of the backbone network. We can also observe from the figure that the partition of CPA( $\text{mindeg}=4$ ) has a longer network lifetime than the partition of GAF, even though they ensure the same level of connectivity. Fig. 8 illustrates the energy consumption of the whole network with regard to time. The energy consumption rate is relatively constant because the traffic

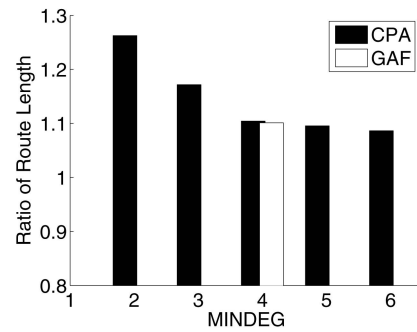


Fig. 9. Comparison of the impact on route length under GAF and CPA.

nodes generate traffic at a constant speed and the number of active nodes remains constant each time as well.

Although lowering the node density reduces the energy consumption of the network, the topology change may affect the network's communication quality. For example, if a packet goes through a much longer path in the backbone network than it does in the original network, a longer data transfer delay will be experienced. Fig. 9 shows the ratio of the average routing path length in the backbone network and the original network for different partitions. The ratio decreases for CPA with higher  $\text{mindeg}$  values because its partition ensures higher connectivity. As can be seen, the ratio is quite low (below 1.3) for all the partitions listed in the figure, which means the node scheduling based on these partitions does not dramatically increase packet delivery delay.

We repeat our experiments under different node densities without changing the range of the network. Table 2 shows the partition sizes when the parameter  $\text{mindeg}$  is set to different values. We can observe that when the node density remains unchanged, the partition size increases with  $\text{mindeg}$ , which is consistent with previous experiment results. In addition, when  $\text{mindeg}$  is fixed, the partition size remains approximately the same under different node densities, which indicates that our algorithm can effectively identify redundant nodes under different node densities.

We also repeat the simulations to see the network lifetime when these partitions are applied. The simulation results are shown in Fig. 10. We can observe from the figure that when  $\text{mindeg}$  is fixed, the network lifetime increases with the increase of node density. This is because more redundant nodes can be utilized to prolong the network lifetime under a higher node density. Moreover, when the node density remains unchanged, the network lifetime decreases with the increase of  $\text{mindeg}$ , because more nodes need to be active in each time slice to maintain better network connectivity.

TABLE 2  
Partition Size of CPA under Different Node Densities

	$\text{mindeg}=2$	$\text{mindeg}=3$	$\text{mindeg}=4$	$\text{mindeg}=5$	$\text{mindeg}=6$
Density=2	66	76	88	99	110
Density=3	70	81	90	103	115
Density=4	69	81	92	104	115
Density=6	73	84	94	105	118

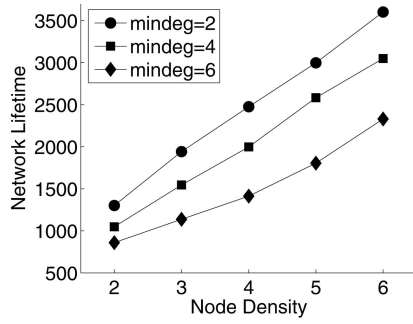


Fig. 10. Network lifetime under different node densities.

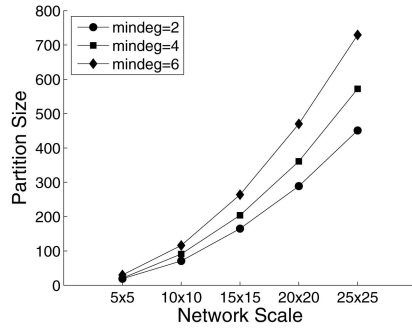


Fig. 11. Partition sizes under different network scales (node density = 5).

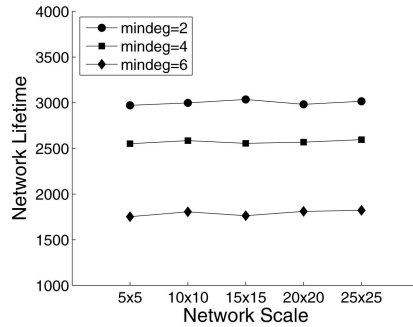


Fig. 12. Network lifetime under different network scales (node density = 5).

In Figs. 11 and 12, we fix the node density to five nodes per square unit and run CPA with  $\text{mindeg}$  equal to two, four, and six, respectively, on networks of different scales. As shown in Fig. 11, the partition size increases with the network scale. However, the network lifetime remains approximately the same regardless of the network scale, which is shown in Fig. 12.

### 4.3 CPA under an Irregular Radio Propagation Model

In order to evaluate the performance of CPA in comparison with GAF under complex environments, we choose the DOI model [11]. This model assumes an upper and lower bound on the signal propagation range. The parameter DOI is defined as the maximum radio range variation per unit degree change in the direction of radio propagation. In our simulation of radio irregularity, we set the upper bound to  $\sqrt{5}$  and the lower bound to half of the upper bound. Fig. 13 shows two examples of radio propagation range in different directions where the DOI value is set to 0.05 and 0.10,

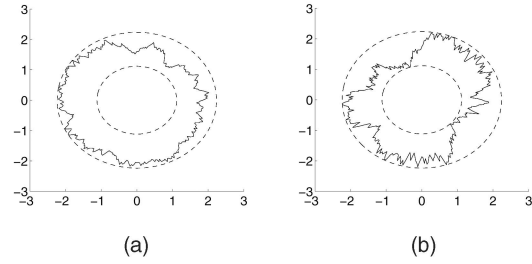


Fig. 13. Irregular radio propagation model. (a) DOI = 0.05. (b) DOI = 0.10.

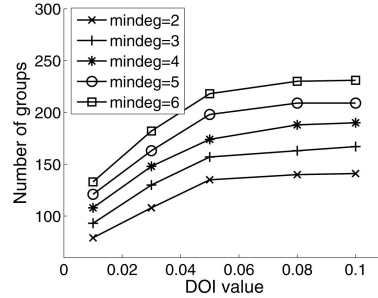


Fig. 14. Partition sizes under different DOI values.

respectively. The higher the DOI value, the more irregularity in the radio propagation range.

Like the previous section, we first perform the simulation in a  $10 \times 10$  square area with the node density of five. GAF cannot adapt to different levels of irregularity in the radio propagation model, because it is based on the sensors' locations and consequently cannot detect the irregularity level. Unlike GAF, CPA partitions sensors based on their measured connectivity, which enables it to obtain appropriate partition sizes under different levels of radio irregularity. Fig. 14 shows the partition sizes of CPA with different  $\text{mindeg}$  under an irregular radio with different DOI values. We can observe in the figure that the partition size increases with the DOI value. As the communication between sensors is more seriously influenced by higher irregularity in the radio propagation model, more active nodes are needed to maintain the same level of connectivity in the backbone network, leading to a larger partition size.

We perform simulations to study the network lifetime under GAF and CPA ( $\text{mindeg} = 2$  or 4). For GAF, we use the same partition for different DOI values, that is, 100 groups with a cell length of one, because GAF is unaware of the radio irregularity. We run the simulation multiple times for each partition, and the comparison of average lifetime is shown in Fig. 15. Our simulation finds that the lifetime for GAF is not stable through repeated simulations. As the connectivity between neighboring cells is no longer guaranteed by GAF under the irregular radio propagation model, there is a possibility that the backbone network formed by randomly selecting an active node from each group is disconnected. The higher the radio irregularity, the higher the probability of a disconnected backbone network. As illustrated in the figure, when DOI is close to 0.1, the GAF partition cannot even work. In contrast, CPA works well under different conditions. The lifetime for CPA decreases with the radio irregularity level, because more active nodes are needed to maintain the same connectivity of the backbone network, and thus, more energy

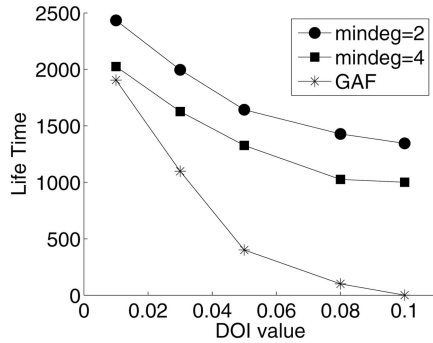


Fig. 15. Network lifetime under different DOI values.

is consumed per time unit. On the other hand, GAF does work if it divides the deployed area by cells with a side length of 0.5 so that any nodes in two neighboring cells are within  $\sqrt{5}/2$ , the lower bound of the radio transmission range in the DOI model. However, this results in 400 groups with an average group size of 1.25. Apparently, this partition can only prolong the network's lifetime for a very small portion.

As we cannot guarantee the  $K$ -connectivity of the backbone network for 100 percent by ensuring the minimum degree of  $K$  in the 2-induced graph of the  $M$ -partition under irregular radio propagation models, we try to evaluate the connectivity of the backbone networks generated by CPA partitions through simulation. We select the CPA partition ( $\text{mindeg} = 4$ ) and compare its connectivity with the GAF partition under different values of radio irregularity. For each partition, we generate different backbone networks by randomly selecting an active node from each group. We then compute the connectivity of each backbone network, and the statistical results are shown in Fig. 16. The percentages of graphs that are disconnected, connected, 2-connected, 3-connected, and 4-connected under the irregular radio with different DOI values are shown as bars with different colors in the figure. Note that if a graph is  $K$ -connected, it is also  $(K - 1)$ -connected,  $(K - 2)$ -connected, and so on. In Fig. 16a, we can observe that the connectivity of the backbone graphs deteriorates dramatically with the increase of radio irregularity for the GAF method. For example, the percentage of 4-connected backbone graphs drops to around zero when the DOI value increases to 0.03; the percentage of 3-connected graphs approaches zero when DOI is 0.08. Furthermore, when DOI

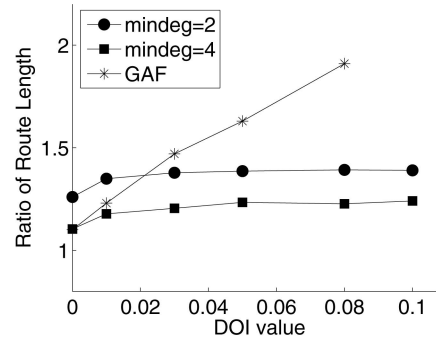
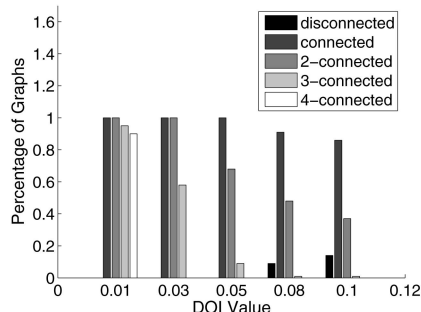


Fig. 17. Impact on route length under irregular radio.

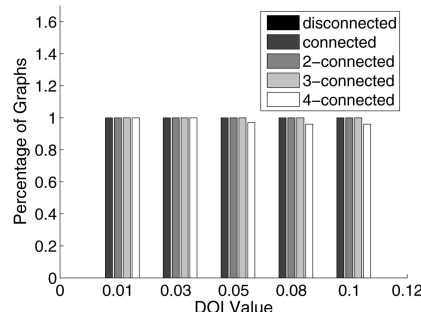
increases to 0.08, there is possibility that the backbone formed by GAF is disconnected. In contrast, as shown in Fig. 16b, CPA maintains the connectivity of backbone graphs much better. It keeps higher than 95 percent of 4-connected graphs even when DOI increases to 0.10, and all backbones generated by CPA are at least 3-connected under different DOI values.

In addition to the connectivity metric, we also evaluate the communication quality of different backbone networks by the ratio of the average routing path length in the backbone network and the original network, because it reflects the packet delivery delay. As illustrated in Fig. 17, the ratio of route length under the GAF method increases dramatically with the DOI value. When DOI reaches 0.1, the ratio is infinite because the backbone network is frequently disconnected. In comparison, the ratio of route length under the CPA method remains approximately constant with the change of the DOI value. The results show that CPA can form well-connected backbone networks, which preserve the communication quality under different levels of radio irregularity. Therefore, it is more adaptive to the real environment than GAF.

We change the number of nodes while keeping the same scale of the deployed area. As shown in Fig. 18, we simulate CPA ( $\text{mindeg} = 4$ ) in the same deployed area with 200, 300, ..., 600 nodes separately, that is, the corresponding densities are two, three, ..., six nodes per square unit. For all the cases, nodes are distributed uniformly. We evaluate the CPA partitions and the corresponding network lifetime under both ideal ( $\text{DOI} = 0$ ) and irregular ( $\text{DOI} = 0.01, 0.03$ ) radio propagation models. Fig. 18 illustrates that the network lifetime under CPA partitions increases with node



(a)



(b)

Fig. 16. Connectivity of backbone networks under irregular radio. (a) Using GAF partition. (b) Using CPA( $\text{mindeg} = 4$ ) partition.

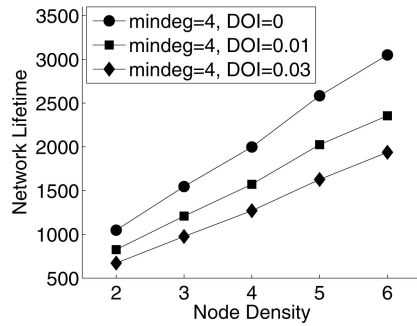


Fig. 18. Effect of node density.

density in both ideal and irregular radio environments. We can also observe that under a certain node density, the network lifetime decreases with the increase of radio irregularity, because CPA needs to keep more active nodes in order to maintain the same level of network connectivity.

#### 4.4 Probability-Based Partition

In this section, we will evaluate the probability-based partition algorithm discussed in Section 3.4. In our simulation, the deployed area is still a  $10 \times 10$  square with 500 nodes uniformly distributed, and the ideal radio propagation model is assumed. The threshold  $P$  is set to 0.90. Fig. 19 shows the partition sizes of both approaches under different configurations of  $mindeg$ . The probability-based approach generates smaller size partitions than CPA because it looses the requirement on the network connectivity. Fig. 20 illustrates the network lifetime corresponding to both approaches. The probability-based approach is able to increase the network lifetime by more than 10 percent at the cost of lowering the network connectivity a little bit. Through simulation of the probability-based approach, we find that the backbone network is  $mindeg$ -connected with a probability of more than 0.90 and  $(mindeg - 1)$ -connected with a probability of more than 0.99.

## 5 RELATED WORK

Energy in sensor networks can be saved by adjusting the radio transmit power of each node. Several topology control algorithms [1], [2], [3] have been proposed to reduce energy consumption by selecting adequate node transmit power while maintaining network connectivity. Ramanathan and Rosales-Hain [1] formulated it as a constrained optimization

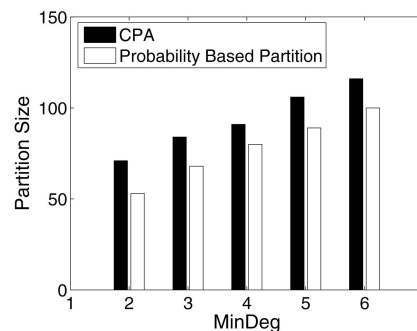


Fig. 19. Comparison of partition size between CPA and the probability-based approach.

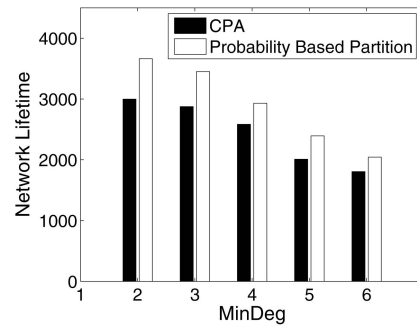


Fig. 20. Comparison of network lifetime between CPA and the probability-based approach.

problem and presented distributed heuristic algorithms to maintain a connected topology using minimum power. Wattenhofer et al. [2] suggested to decide the radio transmit power of each node based on the directional information, that is, a node grows its transmission power until it finds a neighbor node in every direction. Li and Hou [3] proposed  $FLSS_k$ , which minimizes the maximum transmission power used while preserving  $K$ -vertex connectivity of the wireless network. In addition, some other work studied energy saving specifically in routing packets, such as energy-conserving routing [19] and the efficient communication proposed in [20]. Energy-conserving routing [19] selects the appropriate routes and corresponding power levels in order to maximize the network lifetime. Li and Song [20] proposed a topology control algorithm for each node to adjust its transmission power such that the topology formed is efficient for both unicast and broadcast communications.

References [21], [22], and [23] select the set of active nodes for routing purposes based on the idea of approximating a minimum connected dominating set (MCDS). Reference [24] further discusses how to balance energy dissipation in the cluster heads of the connected dominating set (CDS). Deb and Nath [25] proposed a node scheduling approach that can adapt to the trade-off between energy conservation and data delivery quality. Although CDS approaches save energy by decreasing the number of active nodes, they are not efficient at balancing energy consumption among nodes so as to maximize the network lifetime.

To reduce the energy waste in idle listening, duty cycling has been proposed in [7] and [8], where the wireless interface of each node follows a periodic cycle of active/sleep states. Although duty cycling is energy efficient, it increases the delay of data delivery, because the intermediate node has to wait for the next-hop node to wake up to receive the packet. Reference [26] analyzes the bounds of data delivery delay by using completely decentralized duty cycling. In [27], the authors formulate the problem of assigning duty cycle to each node while minimizing the end-to-end communication delay and provide optimal solutions for networks with some special topologies and heuristic solutions for networks with arbitrary topologies.

Node scheduling algorithms that maintain a CDS and balance energy usage by switching node status have been studied in [4] and [9]. These approaches cope with the idle listening problem without causing a dramatic data delivery delay. Span [4] aims at reducing energy consumption of a wireless network without significantly diminishing its

capacity or connectivity. In Span, each node makes a local decision on whether to sleep or join the backbone as a coordinator by periodically checking the status of its neighbors. Unlike Span, GAF [9] divides nodes into groups such that a communication backbone is formed by selecting an arbitrary active node from each group while keeping others in sleeping mode. Compared with Span, GAF imposes less overhead on switching node status, because only nodes within each group need to communicate with each other for load-balance purposes.

Our proposed CPA schedules nodes based on partitioning. Different from GAF, CPA is based on the measured connectivity between nodes instead of their locations. Besides preserving GAF's advantage in efficient load balancing of energy dissipation, it aims at ensuring  $K$ -vertex connectivity of the backbone network and better adaptivity for unideal radio propagation. CEC [10] divides nodes into clusters based on measured connectivity similarly, but it cannot efficiently switch node status within each group like GAF and CPA. Instead, it needs to reform clusters to balance energy consumption among nodes. Friedman [28] proposed timed grid routing, which is based on virtual grids and synchronized clocks. It aims at avoiding message collision, as well as conserving energy through node scheduling. Reference [29] focuses on energy efficiency and preserving network coverage at the same time. In comparison, our work focuses on maintaining network connectivity while saving energy consumption of the network.

## 6 CONCLUSION

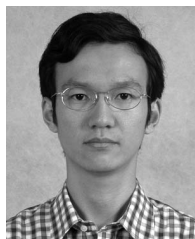
As the energy consumption of idle listening nodes is comparable to active nodes that send and receive packets in a wireless sensor network, node scheduling mechanisms can reduce energy dissipation dramatically. In this paper, we propose to partition the nodes based on their measured connectivity instead of geographic locations. We formulate it as a constrained optimal graph partition problem and present CPA to approximate a good partition. As a distributed algorithm, CPA has fast converging speed and is scalable with the network size. CPA partition outperforms other partition approaches in two aspects. First, CPA can guarantee  $K$ -vertex connectivity of the backbone network under ideal radio propagation models, which balances the trade-off between saving energy and preserving the network's communication quality. In addition, simulation results show that CPA can also ensure  $K$ -vertex connectivity of the backbone network with high probability under irregular radio propagation models. Therefore, CPA has better adaptivity to complex environments.

## ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported in part by the US National Science Foundation (NSF) under Grants CCF-0514078, CNS-0551464, and CNS-0721441.

## REFERENCES

- [1] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment," *Proc. IEEE INFOCOM*, 2000.
- [2] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed Topology Control for Wireless Multihop Ad-Hoc Networks," *Proc. IEEE INFOCOM*, 2001.
- [3] N. Li and J.C. Hou, "FLSS: A Fault-Tolerant Topology Control Algorithm for Wireless Networks," *Proc. ACM MobiCom*, 2004.
- [4] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *Proc. ACM MobiCom*, 2001.
- [5] M. Stemm and R.H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices," *IEICE Trans. Comm.*, 1997.
- [6] O. Kasten, *Energy Consumption*, [http://www.inf.ethz.ch/~kasten/research/bathtub/energy\\_consumption.html](http://www.inf.ethz.ch/~kasten/research/bathtub/energy_consumption.html), 2008.
- [7] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient Mac Protocol for Wireless Sensor Networks," *Proc. IEEE INFOCOM*, 2002.
- [8] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient Mac Protocol for Wireless Sensor Networks," *Proc. First ACM Int'l Conf. Embedded Networked Sensor Systems (SenSys)*, 2003.
- [9] Y. Xu, J. Heidemann, and D. Estrin, "Geography-Informed Energy Conservation for Ad Hoc Routing," *Proc. ACM MobiCom*, 2001.
- [10] Y. Xu et al., "Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks," technical report, 2003.
- [11] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T.F. Abdelzaher, "Range-Free Localization Schemes in Large Scale Sensor Networks," *Proc. ACM MobiCom*, 2003.
- [12] R. Subramanian and F. Fekri, "Sleeping Scheduling and Lifetime Maximization in Sensor Networks: Fundamental Limits and Optimal Solutions," *Proc. Fifth Int'l Conf. Information Processing in Sensor Networks (IPSN)*, 2006.
- [13] J. Luo and J.-P. Hubaux, "Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks," *Proc. IEEE INFOCOM*, 2005.
- [14] W. Wang, V. Srinivasan, and K.-C. Chua, "Using Mobile Relays to Prolong the Lifetime of Wireless Sensor Networks," *Proc. ACM MobiCom*, 2005.
- [15] T. Feder, P. Hell, S. Klein, and R. Motwani, "Complexity of Graph Partition Problems," *Proc. 31st Ann. ACM Symp. Theory of Computing (STOC)*, 1999.
- [16] Y. Ding, C. Wang, and L. Xiao, "A Connectivity Based Partition Approach for Node Scheduling in Sensor Networks," *Proc. Third Int'l Conf. Distributed Computing in Sensor Systems (DCOSS)*, 2007.
- [17] M.D. Penrose, "On  $K$ -Connectivity for a Geometric Random Graph," *Wiley Random Structures and Algorithms*, 1999.
- [18] A. Sankar and Z. Liu, "Maximum Lifetime Routing in Wireless Ad-Hoc Networks," *Proc. IEEE INFOCOM*, 2004.
- [19] J.-H. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-Hoc Networks," *Proc. IEEE INFOCOM*, 2000.
- [20] X.-Y. Li, W.-Z. Song, and W. Wang, "A Unified Energy-Efficient Topology for Unicast and Broadcast," *Proc. ACM MobiCom*, 2005.
- [21] B. Das and V. Bharghavan, "Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets," *Proc. IEEE Int'l Conf. Comm. (ICC)*, 1997.
- [22] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," *Proc. Fourth Ann. European Symposium Algorithms (ESA)*, 1996.
- [23] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-Hop Wireless Networks," *Proc. IEEE INFOCOM*, 2001.
- [24] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks," *J. Comm. Networks*, 2002.
- [25] B. Deb and B. Nath, "On the Node-Scheduling Approach to Topology Control in Ad Hoc Networks," *Proc. ACM MobiHoc*, 2005.
- [26] O. Dousse, P. Mannesalo, and P. Thiran, "Latency of Wireless Sensor Networks with Uncoordinated Power Saving Mechanisms," *Proc. ACM MobiHoc*, 2004.
- [27] B.K.A.G. Gang Lu and N. Sadagopan, "Delay Efficient Sleep Scheduling in Wireless Sensor Networks," *Proc. IEEE INFOCOM*, 2005.
- [28] R. Friedman and G. Korland, "Timed Grid Routing (TIGR) Bites Off Energy," *Proc. ACM MobiHoc*, 2005.
- [29] D. Tian and N. Georganas, "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks," *Proc. First ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA)*, 2003.



**Yong Ding** received the BS and MS degrees from Southeast University, China, in 2001 and 2004, respectively. He is currently a PhD student in computer science at Michigan State University, East Lansing. His research interests are in the areas of distributed systems and computer networking, including wireless sensor networks, vehicular ad hoc networks, and wireless mesh networks. He is a student member of the IEEE and the IEEE Computer Society.



**Li Xiao** received the BS and MS degrees in computer science from Northwestern Polytechnic University, China, and the PhD degree in computer science from the College of William and Mary in 2002. She is an associate professor of computer science and engineering at Michigan State University, East Lansing. Her research interests are in the areas of distributed and networking systems, overlay systems and applications, and wireless sensor and mesh networks. She is a member of the ACM, the IEEE, the IEEE Computer Society, and IEEE Women in Engineering.



**Chen Wang** received the BS and MS degrees from Northeastern University, China, and the PhD degree in computer science and engineering from Michigan State University in 2007. He is currently working at Google Inc., Mountain View, California. His research interests are in the areas of distributed systems and computer networking. He is a senior member of the IEEE and the IEEE Computer Society.

▷ **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**