

Energy Efficient Residual Energy Monitoring in Wireless Sensor Networks

EDWARD CHAN¹ and SONG HAN²

¹Department of Computer Science, City University of Hong Kong, Kowloon,
Hong Kong

²Department of Computer Sciences, University of Texas at Austin, Austin,
TX, USA

A crucial issue in the management of sensor networks is the continuous monitoring of residual energy level of the sensors in the network. With the large number of sensors in a typical network, the energy monitoring process can itself be very energy intensive. In this article, we propose a hierarchical approach to construct a continuous energy map of a sensor network. Our method consists of a topology discovery and clustering phase, followed by an aggregation phase when energy information collected is abstracted and merged into energy contours in which nodes with similar energy level are grouped into the same region. The topology of the monitoring tree is restructured periodically to distribute energy cost among all nodes fairly, which helps to reduce the impact of the monitoring scheme on the lifetime of the sensor network. Simulation results indicate that our method is able to generate accurate energy maps with much lower energy cost compared with traditional monitoring approaches.

Keywords Wireless Sensor Network; Energy Collection; Network Monitoring; In-Network Aggregation

1. Introduction

With recent advances in MEMS-based sensor technology, low-power analog and digital electronics, and low-power RF design, wireless sensor network has a received substantial research interest. A wide range of future applications including scientific data gathering, environmental monitoring (air, water and soil), surveillance, smart homes and offices, personal medical systems, and robotics have been envisaged.

Unlike traditional wireless ad-hoc network, wireless sensor network has several special characteristics. A sensor network typically consists of a large number of nodes and the scalability is of paramount importance. Unlike nodes in the ad-hoc network, the nodes in a sensor network are static once they have been deployed. Finally, sensor nodes have limited resources such as computing capability, memory, and battery power, and it is particularly difficult to replenish the battery of the sensors. Hence methods to preserve energy, as well as the monitoring of the residual energy level are crucial research topics.

In this article, we focus on the issue of efficient residual energy information collection. The process of monitoring energy levels of sensors consumes energy, and since it is done on a continuous basis, it is important to make sure that the process is as energy efficient as

Address correspondence to Edward Chan, Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong. E-mail: csedchan@cityu.edu.hk

possible. Furthermore, given the large number of nodes in a sensor network, typically only an approximation instead of an exact view is required. Just like a weather map, it is sufficient to draw an energy map for a sensor network, in which we separate the sensor nodes into several groups according to different residual energy ranges. If such a map can be generated efficiently and accurately, we can use this energy distribution map to deploy additional nodes to the regions where the energy of the sensors will be depleted soon. 40

In this work, we propose a hierarchical approach for collecting residual energy information in the sensor network continuously in order to construct an energy map at the base station. The entire sensor network is first separated into several static clusters using the TopDisc algorithm proposed by Deb et al. [1]. Each cluster is represented by a head node inside the cluster, and at the same time, a topology tree is constructed which consists of these head nodes and some bridging delivery nodes between two adjacent clusters. Based on the energy information collected, a set of polygons which represent the contours of different energy levels are produced independently for each cluster. The topology tree is then used to collect the energy graphs from leaf nodes to the base station, and in-network aggregation is used to unify the adjacent polygons with the same energy range in order to simplify the overall energy map and reduce the message cost. In addition to the construction of the initial topology tree for energy collection, reorganization of the tree to evenly distribute energy consumption in the monitoring process is performed periodically to extend the battery life of the sensors in the network. 45 50 55

The rest of the article is organized as follows. A summary of related work is presented in Section 2. Section 3 describes the design of the continuous hierarchical residual energy information collection in detail. Section 4 analyzes the energy dissipation of CREM and eScan. Section 5 presents the performance of the algorithm through extensive experimental results comparing with other monitoring approaches. It is demonstrated that our algorithm can reduce energy consumption significantly without introducing excessive error. We conclude the article in Section 6 and discuss future work. 60

2. Related Works 65

Although there has been a large number of recent work on sensor networks [2–5], only a fairly small number explicitly deals with the issue of managing sensor networks and even fewer deals with the monitoring of residual energy levels.

Topology control is the key issue in [7], where two topology control protocols called Geographic Adaptive Fidelity (GAF) and Cluster-based Energy Conservation (CEC) that extend the lifetime of dense ad hoc networks while preserving connectivity are presented. The protocols conserve energy by identifying redundant nodes and turning their radios off: GAF identifies redundant nodes by their physical location and a conservative estimate of radio range. CEC directly observes radio connectivity to determine redundancy, so it can be more aggressive at identifying duplication and is also more robust with respect to radio fading. A *multi-resolution* approach to topology extraction in sensor networks is proposed in [8], which is interesting in that it can be extended to a general-purpose multi-resolution information retrieval framework (where the information can include residual energy, for instance). 70 75

The notion of an energy map for sensor networks is first proposed by Zhao et al., called the residual energy scan (eScan) [9, 13]. This pioneering work applies the techniques of in-network aggregation and abstracted representation of energy graphs which are also used in our algorithm. To reduce the energy cost for collecting an eScan and to make monitoring information available to all nodes within a network, the authors propose another class of 80

monitoring tools called *digests* which are an aggregate of some network properties [13]. They also show how digests can be collected efficiently and accurately on a continuous basis and how they can be distributed throughout the entire network. However, different from our work, there is no notion of a hierarchical structure in eScan and the topology tree consists of all the sensor nodes in the network, which can lead to additional cost in message delivery. Moreover, since no topology maintenance scheme is proposed in eScan, the nodes close to the base station will consume energy at a very high rate for large-sized networks due to the large number of messages delivered, leading to quick depletion of the available energy resources for these nodes. 85 90

In [10, 11], a mechanism to predict the energy consumption by a sensor node in order to construct the energy map of a sensor network is proposed. With the proposed energy dissipation model, a sensor node does not need to transmit its energy information periodically. Instead it can just send one message with its energy information and the parameters of the model, with the major advantages of a greatly extended lifetime for the sensor. This prediction based approach works well when the sensor's energy dissipation rate is relatively stable. Its performance will decrease along with the increase of the events' randomness inside the network. As a consequence, it will not only suffer from the reduced accuracy of the generated energy map, but also has to re-transmit both the energy information and the parameters of the model to the base station. As there is no in-network aggregation mechanism in the model, the re-transmission cost will soon overcome the improvement we achieved from the energy prediction. 95 100 105

A distributed approach to monitoring sensor network is proposed in [6]. The notion of active neighbor monitoring is combined with low overhead passive monitoring by the networkwide control system to provide high responsiveness without incurring heavy energy consumption. However, our continuous hierarchical residual energy collection approach is an active monitoring scheme based on constant updates. Although a passive scheme has advantages over an active one in monitoring independent anomalies in the network, it does not work well in maintaining the continuous energy map because of the difficulty in merging the energy information of individual nodes into the abstracted representation of the energy map as well as the possibility of introducing additional errors in the energy map generation. More specifically, in passive monitoring, if an individual node's energy change is reported, it is difficult to represent this point into the current energy map. Even worse, for example, if three nodes far away from each other all report an energy drop to the same energy range, then it is not trivial to decide whether to take them as an individual point or take them as the contour of a new polygon. The latter one will introduce inaccuracy or even error into the current energy map. 110 115 120

LEACH [12] is a clustering-based protocol designed to minimize energy dissipation in sensor networks. LEACH includes distributed cluster formation, local processing to reduce global communications, as well as randomized rotation of the cluster heads so that the high energy consumption in communicating with the base station is shared by all sensor nodes in the sensor network. It is shown that LEACH is scalable and achieves high energy efficiency for all the sensor nodes. However, a key assumption is that all the nodes can communicate with the base station directly, which can be restrictive for some sensor network configurations. Compared with LEACH, our approach achieves higher scalability with the introduction of a hierarchical structure, thus allowing the sensors that are multiple hops away from the base station to successfully transmit the packets along the constructed topology tree. 125 130

3. Continuous Hierarchy Residual Energy Collection

In this article we propose an energy-efficient mechanism for Continuous Residual Energy Monitoring (CREM), including the provision of an informative and accurate presentation of the energy profile of the entire network to the network manager. The residual energy information is presented in the form of an energy map (Fig. 1), inspired by eScan. The energy map shows the residual energy of different regions in the sensor network. The regions are colored differently depending on the different energy ranges within that region. Using this energy map, the network manager can decide where new sensor nodes need be deployed to maintain the effectiveness of the monitoring activity of the sensor network.

3.1. System Model and Assumptions

In this section we briefly describe the system model used in our study and state the underlying assumptions used in the formulation of our model.

We assume that there is a base station at the network edge. The base station, which has a continuous energy supply, sends a request to all nodes in the sensor network, asking them to collect residual energy. This collection of residual energy information occurs at regular intervals called the *monitoring cycle*. Without loss of generality, we assume that the sensor network consists of N sensor nodes on an m by m square field. The positions of the sensor nodes are generated randomly. Each node is immobile and has symmetric communications to other nodes within a certain range R . The nodes also know their positions, and each node is powered by battery with a normalized capacity of 100. The sensor network consumes energy according to the Hotspot Dissipation model proposed in [9], which will be described in detail in a later section. The positions of the hotspots are also generated randomly according to uniform distribution.

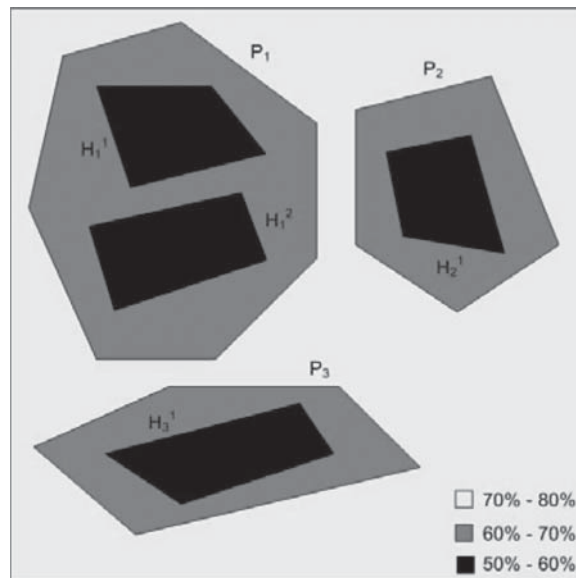


Figure 1. Example of an energy map.

3.2. Topology Discovery

The first task in the energy collection is to organize the network in a way that would reduce the required level of message transmission in the entire monitoring process. The basic approach is to divide the network into a number of clusters, each with a cluster head which acts as a representative for nodes in the neighborhood. The cluster head is responsible for most of the more expensive communication tasks such as replies to topology discovery and maintenance messages, and by doing all these tasks on behalf of other nodes, the overall communication cost is reduced. Of course this also means that the cluster head itself will deplete its energy more rapidly, and this is an issue that will be considered in Section 3.3.4.

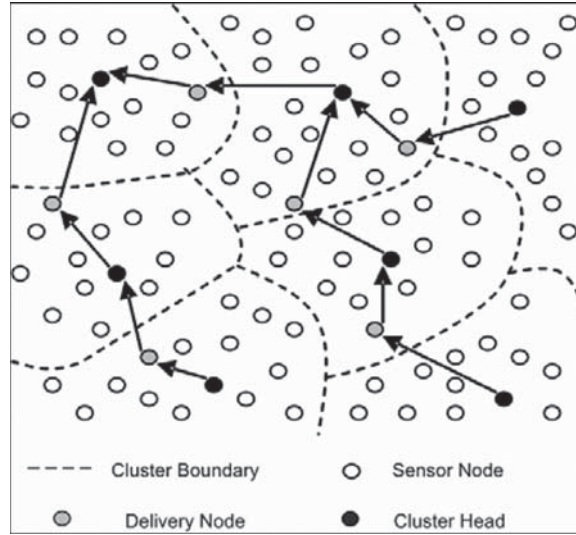
The first step in the topology discovery and network organization process is for the base station to initiate a “Topology Discovery Request.” The request is propagated through controlled flooding so that all nodes receive a request packet if they are connected. At the same time, the sensors which cannot receive any message from their neighbors will be excluded.

In the second step, the sensor network is divided into clusters based on the TopDisc algorithm [1]. This algorithm is based on the simple greedy log (n)-approximation algorithm for finding the set cover. The key steps in the process are summarized below:

1. The node which initiates the topology discovery request is colored black and broadcasts a topology discovery request packet. All other nodes are colored white initially.
2. All white nodes become grey nodes when they receive a packet from a black node. Each grey node broadcasts the request to all its neighbors with a random delay inversely proportional to its distance to the black node from which it received the packet.
3. When a white node receives a packet from a grey node, it becomes a black node with some random delay. In the meantime if it receives any packet from some other black node, it becomes a grey node. Again the random delay is inversely proportional to the distance to the grey node from which the request was received.
4. Once nodes are grey or black, they ignore other topology discovery request packets.

At the end of this process, the sensor network is divided into n clusters and each cluster is represented by a black node, which is called the head node. The head node obtains energy information from all the nodes in the cluster because they are all within its communication range. Each head node knows its parent head node which is the last black node from which the topology discovery was forwarded, but they can not communicate with each other directly. Instead, the head node knows the default grey node to which it should forward packets to reach the parent black node. This grey node is essentially the node from which it received the topology discovery request. Figure 2 is an example of the tree generated in the topology discovery phase.

The algorithm as described is essentially the same as TopDisc. However, our algorithm differs from the original one in one significant aspect. In constructing the topology, we reduced the radius of the communication from range R to $R/2$ to make sure that in each formed cluster, every node is reachable from any other node in the same cluster. This is important to our topology maintenance process, which will be described in detail in Section 3.3.4.



Q2

Figure 2. Topology discovery.

3.3. Periodical Residual Energy Collection

Residual energy collection is a core process that is performed on a continuous basis. It can be divided into three phases: first, determining the local energy graph; second, performing in-network aggregation of the energy information and propagating the result to the base station, and third, reconstructing the topology tree from the root to the leaves to distribute energy consumption in a more equitable manner. In this section, we will first present the abstracted representation of the energy graph used in this article and then describe the three main phases of the residual energy collection in detail.

205

210

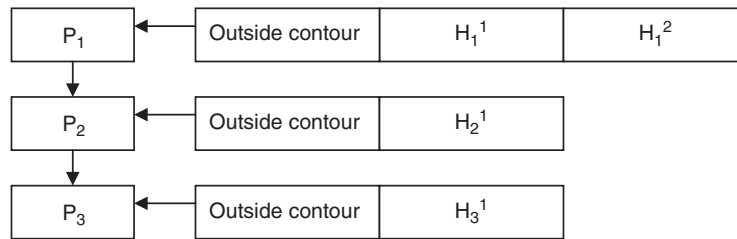
3.3.1. Abstracted Representation of Energy Graph. One of the main objectives of our algorithm is to reduce the message cost. Since there is no need to know the energy information for all nodes, our approach selects the polygon as an abstracted representation of the region inside which sensor nodes' residual energy is in the same range. The structure of the message which is used to deliver energy information is as follows:

215

Header	Sender ID	Receiver ID	Energy Range	Polygon Information
--------	-----------	-------------	--------------	---------------------

Energy Range is a vector which gives the min and max values of the region. Polygon information is a general list, which is a group of separated polygons. Each polygon can be concave and have holes which represent the regions inside them but with different energy ranges. The polygon is also represented by a general list: the first element is the outside contour and the other elements are the outside contours of the holes. The polygon information of Fig. 1 is shown below. It has three separate polygons, P_1 , P_2 , P_3 and the polygons have 2, 2, and 1 holes inside them, respectively.

220



3.3.2. *Determining the Local Energy Graph.* In this step, with the knowledge of each sensor's residual energy information in a certain cluster, we aim at constructing a set of polygons which can provide the contours of the different energy regions inside the cluster. There are two candidates. In the first method, the concave contour for each energy region is obtained directly. This method is more accurate but will generate more vertexes (Fig. 3); it will also complicate the in-network aggregation which merges different polygons with the same energy range together. 225

The second method is to get the convex contour first, and then perform Boolean computing [14] on the set of polygons. The convex contour is not as accurate as the concave contour, but applying Boolean computing will produce a contour that is similar to that of the concave contour with a greatly reduced number of vertexes. Figure 4 shows 230

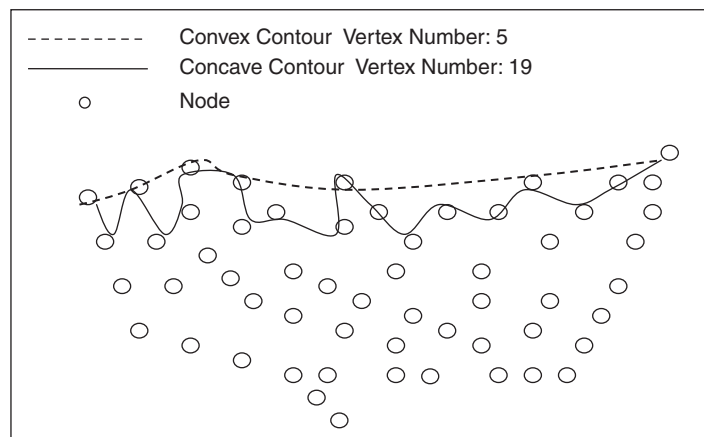


Figure 3. Comparison between concave and convex contours.

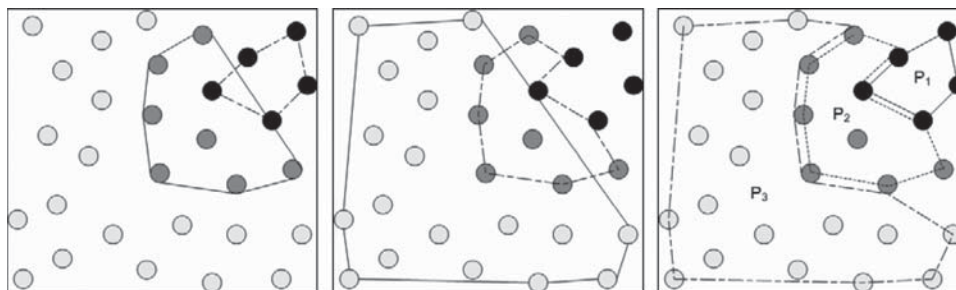


Figure 4. Applying Boolean computing to contours.

the resulting polygons P_1 , P_2 , and P_3 after applying Boolean computing to three convex contours.

During the energy monitoring cycle, every node in a given cluster sends its energy information to the head node of the cluster. The energy information includes the position of the node and the energy value. After the head node has received information from the nodes in its cluster, it divides all the nodes into several sets according to different energy ranges. A convex contour is generated for each node set, and only the vertexes and the sequence are stored. Boolean computing is applied between each contour. The result will be general contours; it can be concave, have holes, and be consisted of several parts. A general list is then used to represent the data structure of the polygon in this cluster, as explained in the previous section.

3.3.3. In-Network Aggregation of Energy Graphs. When a head node gets the local energy graph, the information will be sent to its parent head node through the default delivery node we have mentioned above. When a head node receives all the messages from its children, it will do the in-network aggregation of the energy graphs. If two polygons representing the same energy range are adjacent to each other physically, we can merge them into one polygon to reduce the number of vertexes and communication cost.

This process is demonstrated below. In Fig. 5a, two polygons P_1 and P_2 with the same energy range are adjacent to each other physically. The in-network aggregation merges them together and use the polygon P' to represent the merged region which is shown in Fig. 5b. As a consequence, the total number of vertex needed to represent the same region is reduced from 16 to 10 without loss of accuracy.

3.3.4. Topology Maintenance. Combining the schemes proposed in the previous sections, the energy graph of the sensor network can be obtained efficiently and accurately. However, since the monitoring process is a continuous one, the head node in each cluster will deplete its energy at a much higher rate than other nodes due to the need to constantly transmit and receive messages. Therefore the topology tree needs to be reconstructed periodically.

There are two methods to handle the issue of topology tree maintenance in our hierarchical clustering system: a dynamic scheme and a static scheme.

In the dynamic topology maintenance scheme, the whole sensor network is re-clustered periodically to distribute the energy cost among all nodes more evenly, similar to the

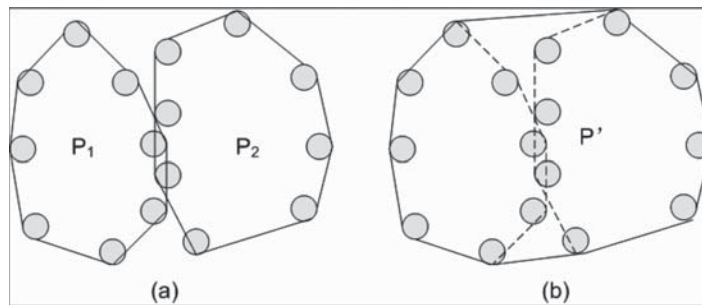


Figure 5. In-network aggregation of energy graphs.

approach taken in [12]. The disadvantage of this approach is the large number of extra messages required in re-clustering the network. 265

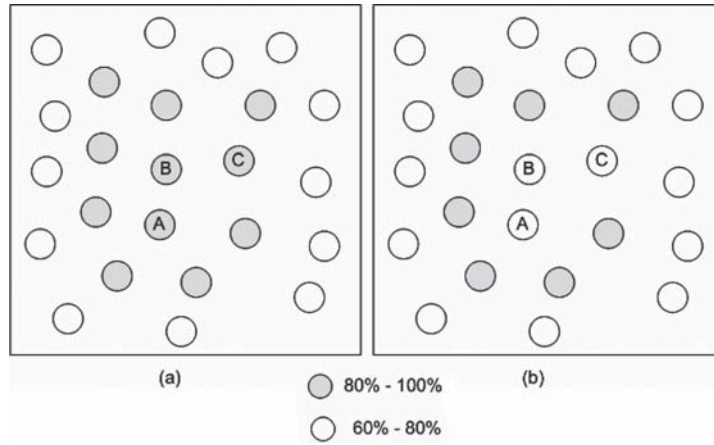
The alternative is to use a static topology maintenance scheme, which is the approach used in this article. In this approach, the hierarchical structure of the clusters is preserved, but a different head node and delivery node in the cluster are used based on the energy graph stored in the former head node. At the beginning of each monitoring cycle, this process is applied to the entire tree from root to leaves. In this process, each pair of parent and child cluster will perform the steps listed below: 270

1. In the parent cluster, the former head node (FP) selects the new head node (NP) according to the energy graph stored in its cache. 275
2. FP sends a RE-CONSTRUCT message which includes the position of NP to the former head node (FC) of the child cluster through the former delivery node (FD).
3. In the child cluster, according to the message from FD, FC selects the new head node (NC), using the same scheme as FP.
4. FC sends a CONFIRM message which includes the position of node NC to FP through FD. 280
5. As the delivery node has the same work load as the head node, it is necessary to rotate the delivery node periodically as well. Consequently, when FP receives the CONFIRM message, it selects another node as the new delivery node (ND) which is different from the new head node, and sends a CONFIRM-DNODE message which includes the position of ND to FC through FD. Then it will broadcast a TOPOLOGY-CHANGE message with the position of NC to all nodes in its cluster. 285
6. In the child cluster, after receiving the message CONFIRM_DNODE, FC sends a message to NC informing it of the new delivery node. Then it broadcasts the TOPOLOGY-CHANGE message (with the position of NC) to all nodes in its cluster. 290

After all communicating node pairs have finished the exchange of various messages, the new topology tree will be constructed. We will now discuss how the new head nodes and delivery nodes are selected within the cluster. Clearly, in order to reduce the size of the energy graph and also to facilitate in-network aggregation of the energy contours, we need to keep the energy graphs as regular as possible. A naïve approach is to always choose the node with the most remaining energy in the cluster as the new head node. Figure 6a shows the result of using this approach. Node *A*, *B*, and *C* have the most residual energy in descending order. By simply choosing nodes with the most residual energy without regard to existing energy contours, the energy levels of the nodes are depleted and they drop into a lower energy range, resulting in a highly irregular energy contour shown in Fig. 6b. 300

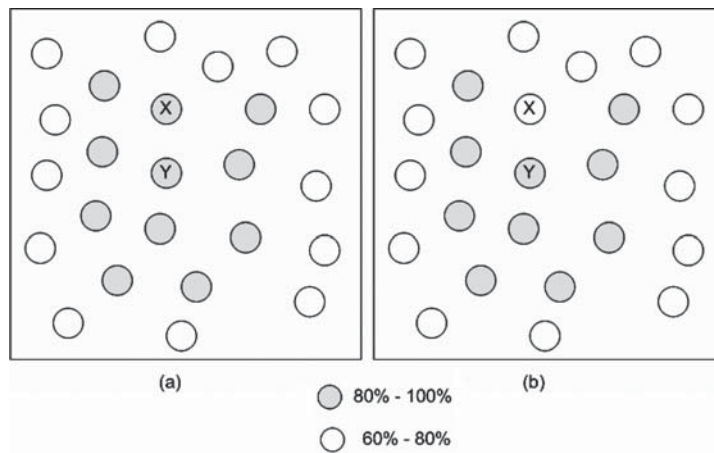
The approach used in this article is to select a node based not on its residual energy but on its proximity to the next lower energy range. This is shown in Fig. 7. The black nodes belong to a certain energy level and the white nodes belong to the next lower energy range. Suppose Node *X* has the minimum distance to any node in the next energy range, and *Y* has the most residual energy. If *X* (instead of *Y*) is chosen as the new head node or delivery node, after it has consumed substantial energy in its role and its residual energy drops to the next lower level, the energy graph will still be quite regular (Fig. 7b). 305

3.3.5. Energy Model. The nature of the underlying energy model of the sensor network has a significant impact on the effectiveness of any method for constructing energy maps. Clearly, if the energy consumption of the sensors is random, then the resulting energy map will be quite chaotic and the aggregation of energy contours will not yield meaningful reduction in the amount of information required in the energy graph. However, we believe 310



Q2

Figure 6. Selection of new head nodes based on residual energy. (a) is the initial state and (b) is the state after energy of A, B, and C have dropped to the lower range.



Q2

Figure 7. Selection of new head nodes based on distance to next energy level. (a) is the initial state and (b) is the state after energy of X have dropped to the lower range.

the Hotspot energy model [9] is a much more accurate model of energy consumption in the real world, where events tend to exhibit temporal as well as spatial locality, and we will use this model as the basis of our experiments.

315

In the Hotspot model, there are a number of hotspots uniformly and randomly distributed in the sensor field, but their locations are fixed. Each node n has a probability of $p = f(d)$ to initiate a local sensing activity and every node within a circle of r centered at n consumes fixed amount of energy e . The minimum distance to every hotspot node is denoted by d . We use the following density function based on the exponential distribution (with $\alpha = 0.5$):

320

$$f(x) = ae^{-\alpha x}$$

4. Analysis of Energy Consumption

In this section, we compare the energy dissipation speed of CREM and eScan based on a simplified model. Our analysis shows that it is CREM's abstracted representation of energy graph and its cluster head rotation that result in the extension of network lifetime. 325

4.1. Model Description

In our model, CREM topology maintenance cost is not considered as it is relatively small. For comparison purpose, we investigate eScan and CREM with the same sensor network in which sensors are uniformly deployed with a density of d . Also, for the simplicity of comparison, the sensor network is restricted in a circular area with radius as $3 * R$ (R is the sensor communication range). 330

Under this model, eScan organizes the sensors in a tree structure which is illustrated in Fig. 8. It involves all the nodes in the area under analysis except those not reachable from the sink node. eScan divides the nodes into 3 levels, L_0 , L_1 , and L_2 respectively. L_0 level nodes are those that can communicate directly with the sink node while nodes in L_1 level rely on a L_0 level node to relay their packets to the sink. Similarly, a L_2 level node needs a two-hop communication to reach the sink. 335

On the other hand, CREM is organized in a tree structure only involving the sink node, cluster headers, and delivery nodes. This structure is illustrated in Fig. 9. As we have mentioned in Section 3.2, CREM reduces the sensor's communication range (except the sink) from R to $R/2$ for the topology maintenance purpose. With this restriction, in our simplified sensor network, there are three levels of clusters, L_0 , L_1 , and L_2 . Level L_0 contains a unique cluster which consists of the sink node and those nodes it can directly communicate with. The radius of this cluster is R because each pair of the nodes in this cluster can communicate through the sink. The headers of L_1 level clusters are connected to the sink through a designated delivery node in the level L_0 cluster. Similarly, a header of L_2 level cluster will relay its packets to a L_1 level cluster header through a proper delivery node in the level L_1 cluster. All the other normal nodes only need to update their energy information to the cluster header for in-network aggregation. Notice that in Fig. 9, some areas do not appear to be covered by either cluster, but in practice, the shape of the cluster 340
345
350

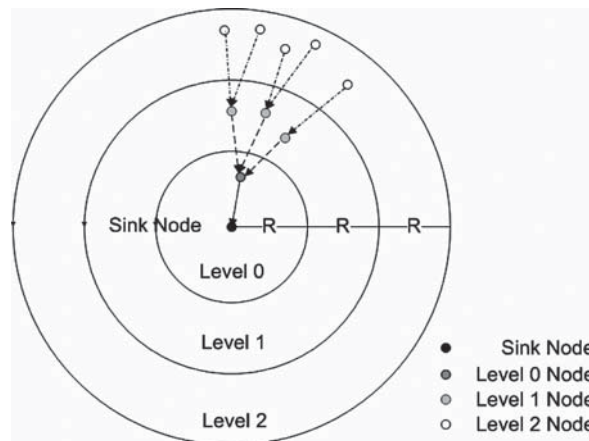


Figure 8. Tree structure in eScan.

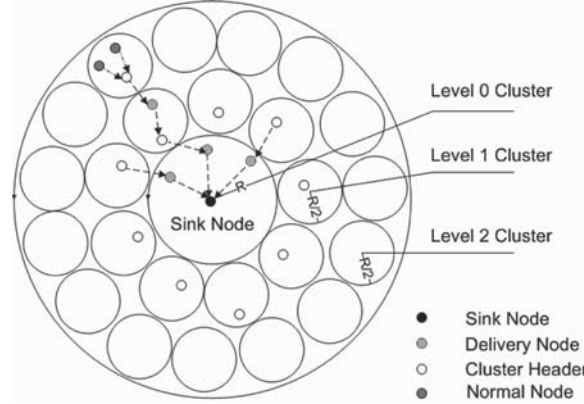


Figure 9. Tree structure in CREM.

formed by CREM should not be a circle but an irregular polygon. We only assume them to be circles here for simplicity.

4.2. Comparison of Energy Consumption in eScan and CREM

In this section, we focus our analysis on the energy consumption rate of L_0 level nodes in both eScan and CREM. L_0 level nodes are responsible for relaying the energy updates from the sensors all over the network. They are those most likely to use up their energy resulting in network disconnection and severe degradation in network performance.

We begin our analysis with CREM first. Denote the average number of clusters in level L_1 and L_2 as \overline{N}_1^c and \overline{N}_2^c respectively, and we get:

$$\overline{N}_1^c = \frac{\pi \cdot (2R)^2 - \pi \cdot R^2}{\pi \cdot (R/2)^2} = 12$$

$$\overline{N}_2^c = \frac{\pi \cdot (3R)^2 - \pi \cdot (2R)^2}{\pi \cdot (R/2)^2} = 20$$

Now denote the average number of normal nodes in each L_1 and L_2 cluster as \overline{N}_n , with a pre-determined sensor deployment density d mentioned earlier, we have:

$$\overline{N}_n = \pi \cdot (R/2)^2 \cdot d - 1$$

As we are using the abstracted representation for the energy graph in CREM, only the nodes on the contour of the cluster will be involved in constructing the representative polygon and the average number of nodes, \overline{N}_r on the contour of L_1 and L_2 level cluster is:

$$\overline{N}_r = \frac{2 \cdot \pi \cdot (R/2)}{1/\sqrt{d}} = \pi \cdot R \cdot \sqrt{d}$$

Here $1/\sqrt{d}$ is the averaged minimum distance between two adjacent sensor nodes in the area under test. The cluster header will report these representing nodes' information to the

Q2

355

360

365

sink through the designated delivery node. To receive and relay these reports to the sink, each delivery node will consume twice as much energy as the cluster head. Assume the energy to transmit the energy update information for one normal node to be J , then the energy consumed by each delivery node in level L_1 and L_0 are \overline{E}_d^1 and \overline{E}_d^0 respectively:

370

$$\begin{aligned}\overline{E}_d^1 &= \overline{N}_r \cdot J \cdot 2 = 2 \cdot \pi \cdot R \cdot J \cdot \sqrt{d} \\ \overline{E}_d^0 &= (\overline{N}_r \cdot \frac{\overline{N}_2^c}{\overline{N}_1^c} + \overline{N}_r) \cdot r \cdot 2 \cdot J = \frac{16}{3} \cdot r \cdot \pi \cdot R \cdot J \cdot \sqrt{d}\end{aligned}$$

Here we assume that for each cluster header, r is the average abstraction rate for aggregating the energy graphs received from its children. As there are \overline{N}_1^c clusters in the network under investigation, in CREM, \overline{N}_1^c delivery nodes are designated in level L_0 and the overall energy consumed in relaying the energy updates in level L_0 is \overline{E}_t :

$$\overline{E}_t = \overline{N}_1^c \cdot \overline{E}_d^0 = 64 \cdot r \cdot \pi \cdot R \cdot J \cdot \sqrt{d}$$

Taking the advantage of rotation in CREM, each node in level L_0 will take the role of the delivery node in turn thus evening out the energy consumption in the monitoring process. The average energy consumed for each Level L_0 node in one active monitoring period is \overline{E}_a :

375

$$\overline{E}_a = \frac{\overline{E}_t}{\pi \cdot R^2 \cdot d} = \frac{64 \cdot r \cdot \pi \cdot R \cdot J \cdot \sqrt{d}}{\pi \cdot R^2 \cdot d} = \frac{64 \cdot r \cdot J}{R \cdot \sqrt{d}}$$

From the above analysis, we observe that it is CREM's abstracted representation of an energy graph and its cluster head rotation that effectively reduces the energy consumption rate and eventually expands its lifetime. Without applying these two technologies, the average consumed energy for each designated delivery node \overline{E}_a' is:

380

$$\overline{E}_a' = ((\overline{N}_n + 1) \cdot \frac{\overline{N}_2^c}{\overline{N}_1^c} + (\overline{N}_n + 1)) \cdot 2 \cdot J = \frac{4}{3} \cdot \pi \cdot R^2 \cdot d \cdot J$$

Let the ratio between \overline{E}_a and \overline{E}_a' be r_{CREM} . We have the observation that r_{CREM} will decrease along with increase of the sensor communication range and deployment density.

$$r_{CREM} = \frac{\overline{E}_a}{\overline{E}_a'} = \frac{48 \cdot r}{\pi \cdot R^3 \cdot \sqrt{d^3}}$$

Now we will analyze the energy consumption in eScan. As in CREM, we focus on the energy consumption rate of level L_0 nodes because they play a critical role in deciding the lifetime of the sensor network. First, consider the level L_1 nodes. The average number of its children in level L_2 is:

385

$$\overline{N}_1 = \frac{\pi \cdot (3R)^2 - \pi \cdot (2R)^2}{\pi \cdot (2R)^2 - \pi \cdot (R)^2} = \frac{5}{3}$$

We observe that \overline{N}_1 is so small that most of the nodes would be on the contour of the energy graph and thus need their energy update report to be relayed to the upper layer and the effect

of applying the abstracted representation to generate the energy graph is poor. Similarly for each node in level L_0 , the average number of its children in Level L_1 nodes, \overline{N}_2 , is: 390

$$\overline{N}_2 = \frac{\pi \cdot (2R)^2 - \pi \cdot R^2}{\pi \cdot R^2} = 3$$

As we have assumed that the average abstraction rate for aggregating energy graphs is r , then the energy consumption for each node in Level L_0 to relay its children's energy update to the sink is $2 \cdot \overline{N}_1 \cdot \overline{N}_2 \cdot r \cdot J$. The ratio between the energy consumption rate of eScan and CREM is: 395

$$K = \frac{2 \cdot \overline{N}_1 \cdot \overline{N}_2 \cdot r \cdot J}{\overline{E}_a} = \frac{2 \cdot \overline{N}_1 \cdot \overline{N}_2 \cdot r \cdot J \cdot R \cdot \sqrt{d}}{64 \cdot J} = \frac{5 \cdot R \cdot \sqrt{d}}{32}$$

5. Simulation

In this section, two sets of simulation experiments are presented to evaluate the effectiveness of our method. In the first group of experiments, our algorithm is compared with a centralized monitoring scheme to demonstrate that CREM reduces overall message cost without compromising accuracy. The second group of experiments investigates on how CREM reduces the energy required for continuous energy monitoring by examining the impact of different phases on the lifetime of the sensors. 400

5.1. Performance Metrics

In this section we will define the following performance metrics used in the experiments:

1. Residual reachable nodes 405
2. Fidelity
3. Total Message Cost

Residual Reachable Nodes measures the number of nodes which can be reached by a path from the base station. If the number of residual reachable nodes is small, then clearly many of the sensors have run out of energy and the sensor network is unlikely to function well. 410

Fidelity is used to measure how accurately the energy map represents residual energy information in the actual network. We divide the energy information of a sensor node (from 0% to 100%) into N ranges, and each is called an *energy range*, where the i th energy range stands for the range $\langle 100\% \cdot (i-1)/N, 100\% \cdot i/N \rangle$. If a polygon consists entirely of nodes which fall into the energy range X , we call this polygon the *related polygon* to X . For any node n in a sensor network S , if n is in the energy range X , and n is inside the contour or on the outside contour of its related polygon, it is a *correct node*, otherwise it is a *wrong node*. Now we define the fidelity F of the energy map as follows: 415

Fidelity = total number of correct nodes/total number of the nodes in sensor network. 420

Since the message transmission is a major source of energy consumption in a sensor network, the overhead of our algorithm is reflected in the level of message transmission, 425

measured by the total number of bytes transmitted. The overall message cost consists of two components: the message cost involved in performing the periodic energy scan and that incurred in maintaining the topology tree:

$$Total\ Message\ Cost(bytes) = Cost_{local\ energy\ scan} + Cost_{topology\ tree\ reconstruction}$$

where $Cost_{topology\ tree\ reconstruction} = Cost_{handshake} + Cost_{broadcast}$. $Cost_{local\ energy\ scan}$ is the message cost used to get the energy graph for the sensor network; $Cost_{topology\ tree\ reconstruction}$ stands for the message cost used to reconstruct the topology tree at the beginning of each monitoring cycle. Topology tree reconstruction consists of two parts: $Cost_{handshake}$ stands for the message cost involved in the handshaking process between the parent and child cluster heads for choosing the new head nodes and delivery nodes, while $Cost_{broadcast}$ represents the message cost incurred by broadcasting information regarding the new topology to other ordinary nodes in the cluster. 430 435

5.2. Simulation Setup and Parameters

To study the performance of CREM, a simulation program is written using C++ and CSim-18 [15]. The network density is fixed as 1 node/1 unit². For example, if there are 100 nodes, then the simulation space is 10 unit * 10 unit. The position of the base station is at the origin <0, 0>. The size of the network ranges from 100 nodes to 800 nodes. All the simulation results are based on the averaged simulation outputs of 500 runs. For each run, the topology of the sensor network is generated randomly. We follow the Hotspot model proposed in [9]. A perfect MAC layer is assumed so that there is no need to consider the transmission errors. The energy monitoring cycle is set to 100 time units, i.e., each node monitor has its residual energy every 100 time units. The sensor network is divided into 5 energy ranges, from <0%, 20%> to <80%, 100%>. We assume the energy range to be represented by one byte, and two bytes are used to represent the position for each node. In the calculation of the message cost, the protocol overhead is ignored. 440 445

CREM distributes these nodes evenly in space, and uses the TopDisc algorithm to generate the clusters and form the topology tree. The in-network aggregation using polygon contours is applied to reduce the transmission overhead. 450

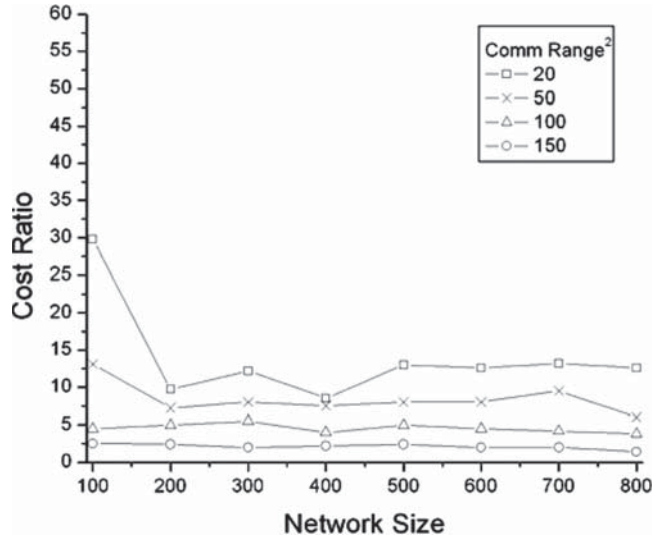
eScan uses flooding to generate the tree, and other parameters are the same as CREM for comparison.

A centralized schema collects data using the same frequency with CREM for comparison. The routing mechanism is like eScan, i.e., use a flooding algorithm to generate the topology tree. The difference is that the centralized schema does not take advantage of the in-network aggregation. 455

5.3. Simulation Results

In the first group of experiments, we compare CREM with a centralized collection, which collects the individual residual energy information directly from each node. In particular, we explore the effect of network size and communication range on the two algorithms, using the total message cost and fidelity as the performance metrics. 460

Figure 10 demonstrates the ratio of the total message cost for the centralized collection and CREM. It can be seen that CREM consistently outperforms the centralized collection by a wide margin regardless of network size, which means that the outperformance of the 465



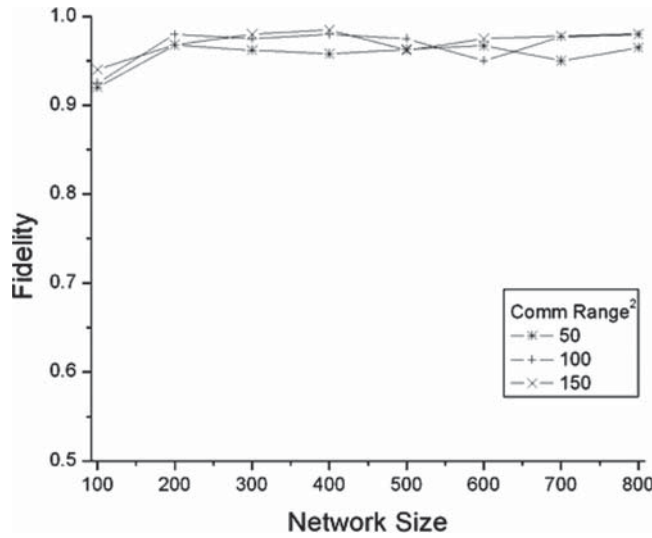
Q2

Figure 10. Cost ratio between CREM and centralized collection.

CREM algorithm is scalable. Moreover, Fig. 10 clearly indicates that for a given network size, the cost ratio between the centralized collection and CREM increases as the communication range increases. This is expected because as the communication range increases, the number of nodes in the cluster will increase, which also means that many more nodes fall into the contour of a certain energy range and hence can be ignored by CREM. An anomaly occurs when the communication range is small and the network size is 100, in which case the cost ratio is very high (almost 30). This is because in this situation most of the nodes actually lie in the cluster whose head node is the base station, hence the message cost to transmit the energy map information is greatly reduced. Figure 11 shows the fidelity

470

475



Q2

Figure 11. Fidelity vs network size for CREM.

achieved using CREM, which is typically above 95%. Thus, CREM is able to significantly reduce the total message cost at only a small degradation in fidelity when compared to the centralized collection (which has a fidelity of 100%).

In the second group of experiments, we examine the impact of several algorithms on the lifetime of the sensor network. The first method, the centralized collection, gathers energy information from every node in the network directly as described previously. The second method, called Static Clustering, is the same as our approach in that a hierarchical tree topology is used for residual energy monitoring but without topology restructuring. By comparing the performance of CREM to Static Clustering we can isolate the impact of topology restructuring on the overall performance. We also compare CREM with eScan in which no notion of a clustering structure is introduced and demonstrates the improvement in performance.

The results of the experiments, with a fixed communication range of 5 units for the sensor, are shown in Figs. 12–19. Figure 12 shows the effect of network size on fidelity. The fidelity of the energy map is over 90% initially, and stays in that range for a considerable number of monitoring cycles. However, as the number of nodes which have exhausted its energy climbs, it is more and more difficult to maintain an accurate representation of the energy levels. This effect is more pronounced when the network increases in size. As the number of nodes increases, the number of levels in the hierarchy increases and the total number of messages handled by the head nodes and delivery nodes increases as well, so that their energy is consumed at a higher rate. As a result of this decrease in the number of functional nodes, accurate representation becomes more difficult.

In Figs. 13–15, the various components contributing to the average message cost per node are shown. Figure 13 shows the normalized cost for the local energy scan. There are actually two components in this cost: the cost for the energy scan in a node cluster at the leaf of the hierarchy, and the cost for propagating the information up the tree. The second component is roughly proportional to the network size, while the first component is independent of it. The curves in Fig. 13 confirm the presence of these two components:

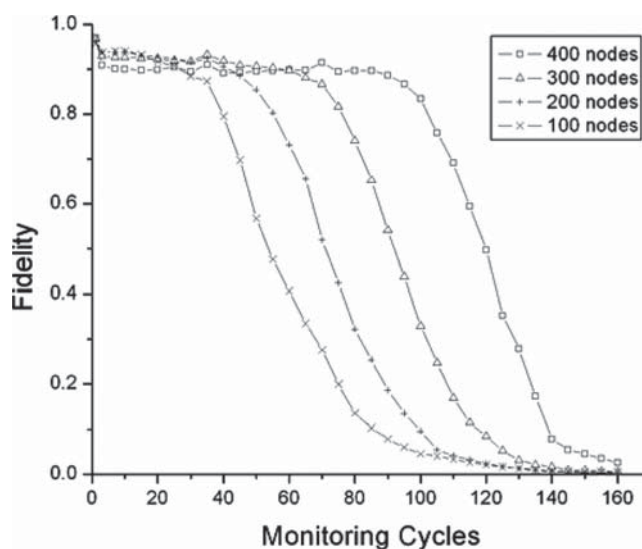
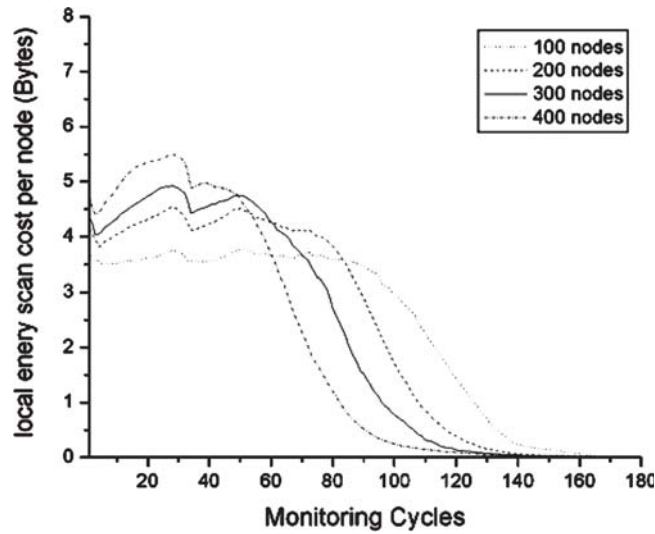


Figure 12. Fidelity vs monitoring cycles for CREM.



Q2

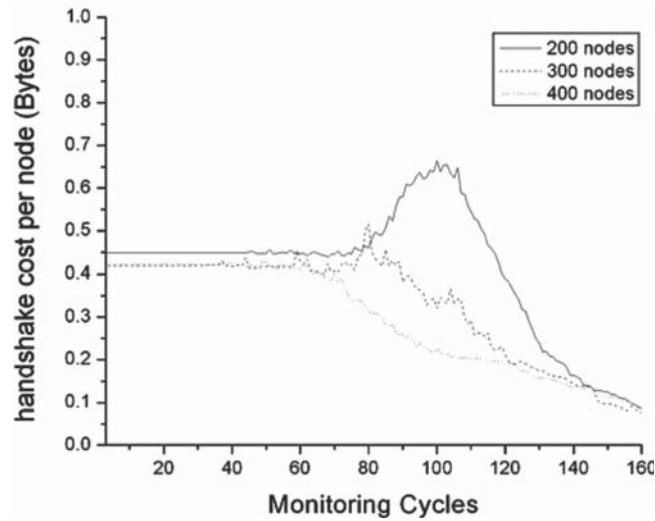
Figure 13. Cost for local energy scan vs. monitoring cycles for CREM.

hence the roughly constant offset of the different curves. Note that the cost per node drops towards the end of the life of the sensor network. The reasons is that as more and more sensors run out of energy, a cluster may not be able to connect to its parent cluster. Since it cannot even perform the local energy scan the message cost is reduced. Of course, we are not interested in this part of the lifecycle since network capability is likely to be seriously degraded at this period.

505

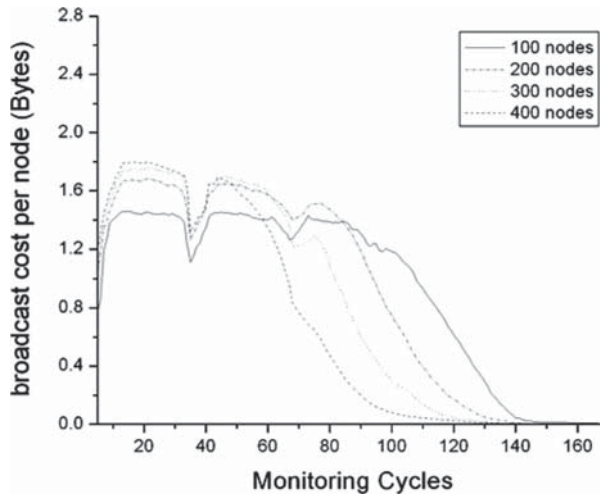
Figure 14 shows that the handshaking cost for the selection of new cluster heads and delivery nodes is roughly the same among all network sizes during the monitoring cycles when the network functions normally (i.e., up to about cycle 80). There is then an abrupt

510



Q2

Figure 14. Cost for handshake vs. monitoring cycles for CREM.



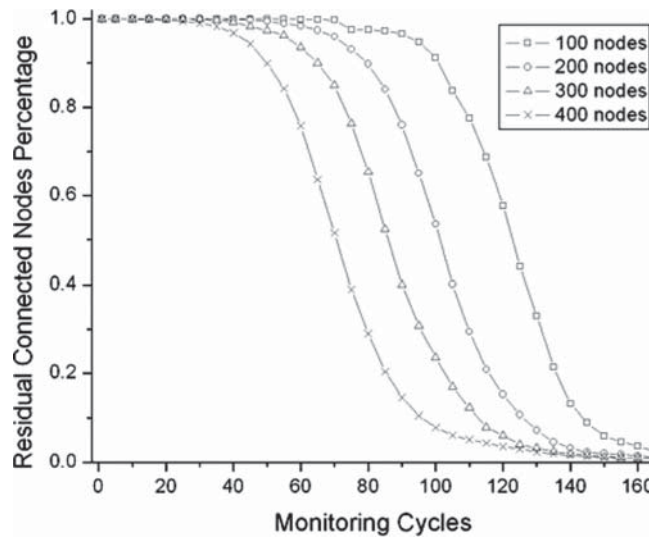
Q2

Figure 15. Cost for broadcast vs. monitoring cycles for CREM.

increase in the cost before the network monitoring process basically falls apart due to the lack of connected nodes. The reason for this abrupt increase is that the number of functional nodes will decrease over time as more and more nodes run out of energy, and two adjacent clusters may need to perform more than one handshake to select a new pair of a head and a delivery node that can maintain the topology tree. The effect is particularly drastic if the network has a small number of sensor nodes to start with, whereas the effect is noticeably more muted for the network with a large number of nodes (such as a 400 node network). Figure 15 shows that the broadcast cost for each node remains about the same for all network sizes during the normal operation of the network, which is not surprising given that

515

520



Q2

Figure 16. Residual reachable nodes vs. monitoring cycles for CREM.

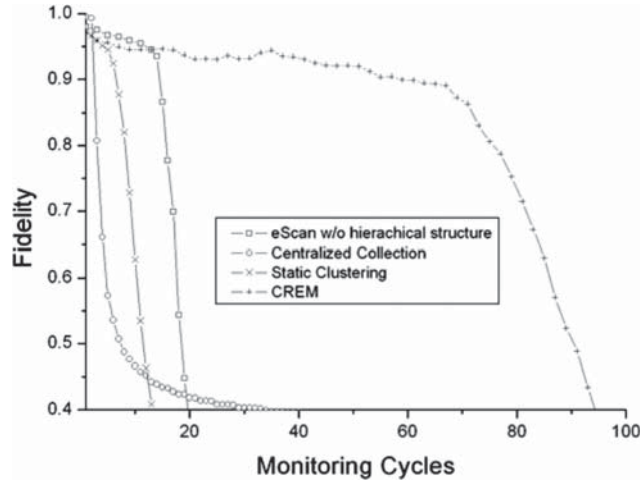


Figure 17. Fidelity vs. monitoring cycles for the four methods.

the average size and number of local clusters remains roughly the same for a different network size (recall that the density of nodes remained the same in our experiments even as the network size increases). This cost also stays fairly constant for a given network size during the monitoring cycles of normal operation.

These sharp periodic drops in cost can actually be seen in both Figs. 13 and 15, but is particularly obvious in Fig. 15 (for instance at around the 35th monitoring cycle). This anomaly is explained in Fig. 19, in which we examine the nodes in a particular cluster. The nodes start with two different residual energy ranges (Fig. 19a), but after a period of time sufficient nodes have their energy depleted so that all nodes in the cluster belong to the same energy band (Fig. 19b). What this means is that there will be fewer contours to send

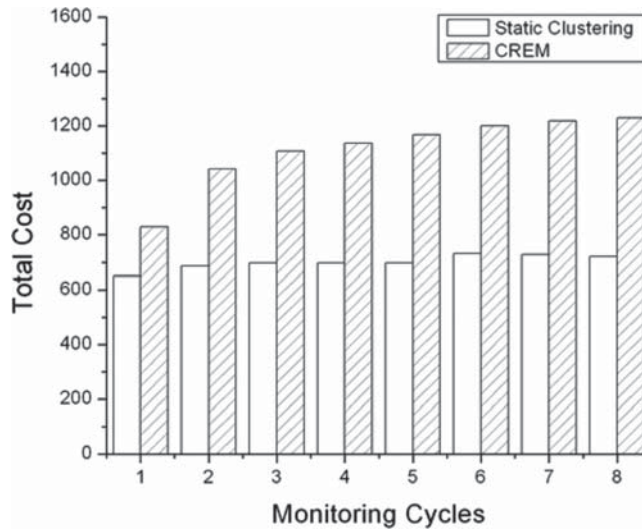


Figure 18. Total message cost: CREM vs. static clustering.

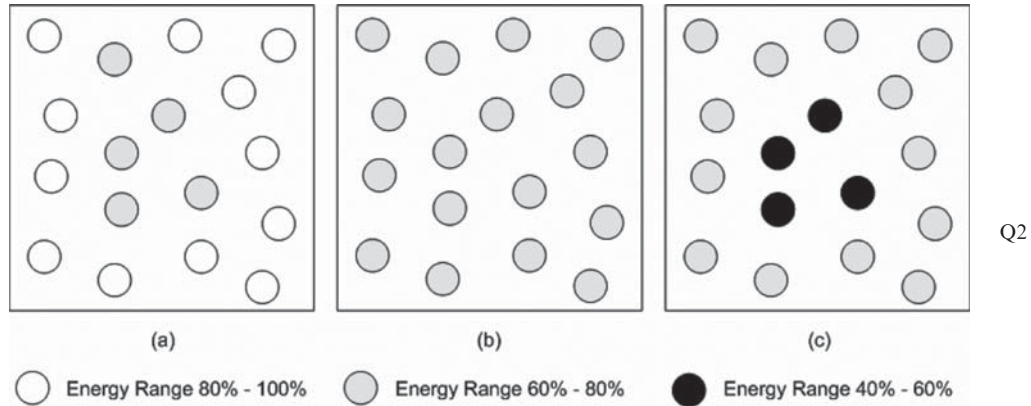


Figure 19. Reason for the periodical drop in cost.

and hence the broadcast cost decreases, which leads to a corresponding drop in overall cost. Eventually some nodes will move into the next lower energy band (Fig. 19c) and the message cost will go back up again due to an increase in the number of contours. The situation depicted in Fig. 19b will occur periodically as can be seen in Figs. 13 and 15.

Figure 16 shows clearly that the lifetime of the network decreases with an increasing network size. The reason is that for larger networks, there are more levels in the hierarchy and the head nodes and delivery nodes have more energy information to deliver, and hence depleting their energy more rapidly. This is consistent with the results seen in other experiments discussed in this section.

The fidelity of the four algorithms over the lifetime of the sensor network is shown in Fig. 17. In this experiment, the sensor communication range is 10 units and there are 400 nodes in the $20 * 20$ simulation space. If we assume that fidelity of about 90% is acceptable, then it can be seen that CREM results in a 20–25 fold increase in the lifetime of the network compared to a centralized collection, a 7–8 fold increase compared to Static Clustering, and a roughly four-fold increase compared to eScan without a hierarchical structure. We notice that according to the energy consumption analysis in Section 4.2, the estimation of the improved lifetime under CREM is around 1.6 but the experimental result is around 4. This is because in the analysis, we assume that the abstraction rate, r , is a constant for both CREM and eScan. However, as the energy graphs generated in eScan especially in the lower level (for example, Level L_2) are much smaller than CREM, the effect of the in-network aggregation of eScan will be comparatively poorer. As a tradeoff, the other three methods do have a marginally higher fidelity than CREM (by about 3%) during the first few monitoring cycles, but even then CREM still supports an average fidelity of over 95%, which should be adequate for most energy monitoring applications. It is clear that CREM is able to greatly extend the lifetime of the network at the expense of just a minor decrease in fidelity.

If we compare the total message cost between Static Clustering and CREM in Fig. 18, it can be seen that our algorithm introduces an additional overhead of around 25%, primarily due to the addition of $Cost_{handshake}$ and $Cost_{broadcast}$. Although this is a bit high, by suitably distributing the additional energy consumption among different nodes, the lifetime of the sensor network is extended by an even greater amount as shown in Fig. 17, making this additional cost very worthwhile.

Q2

535

540

545

550

555

560

6. Conclusion

Energy is one of the most critical resources in a sensor network, and it is important to monitor the availability of this resource with low overhead. In this article, we propose a hierarchical approach to continuously collect and aggregate residual energy information in the form of an energy map. By using techniques such as in-network aggregation, the construction of a hierarchical monitoring structure, as well as the continuous rotation of cluster heads, the energy cost for message transmission is reduced. Extensive simulation results show that our approach is energy efficient and able to extend the lifetime of the sensor network substantially compared with other methods, with only a small decrease in accuracy of the energy map.

Currently we are exploring the possibility of including a prediction based network state model similar to that in [11] to see if further savings in energy consumption are possible. Residual energy monitoring is only part of the overall monitoring process, and we are looking into ways to apply the techniques described in this article to other sensor network monitoring activities as well. Other interesting issues that are worth further investigation include exploring other approaches for energy-efficient monitoring (for instance, modifying multi-resolution data aggregation algorithms to report energy metrics), as well as investigating into the pros and cons of a periodic explicit energy map update mechanism.

About the Authors

Q1

References

1. B. Deb, S. Bhatangar, and B. Nath, "A topology discovery algorithm for sensor networks with applications to network management," *Proc. IEEE CAS Workshop on Wireless Communications and Networking*, Pasadena, CA, Sept. 2002. 585
2. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks", *Proc. ACM/IEEE International Conference on Mobile Computing and Network*, Aug. 1999. 590
3. A. Lim, "Distributed services for information dissemination in self-organizing sensor networks," Special Issue on Distributed Sensor Networks for Real-Time Systems with Adaptive Reconfiguration, *Journal of Franklin Institute*, Elsevier Science Publisher, vol. 338, 2001, pp. 707–727.
4. L. Subramanian, and R. H. Katz, "An architecture for building self-configurable systems," *Proc. IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing*, Aug. 2000. 595
5. I. A. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "A survey of sensor networks," *IEEE Communications*, vol. 40, no. 8, pp. 102–115, Aug. 2002.
6. C. F. Hsin and M. Y. Liu, "A distributed monitoring mechanism for wireless sensor networks," *Proc. Workshop on Wireless Security*, 2002. 600
7. Y. Xu, S. Bien, Y. Mori, J. Heidemann, and D. Estrin, "Topology control protocols to conserve energy in wireless ad hoc networks," submitted for review to *IEEE Transactions on Mobile Computing*.
8. B. Deb, S. Bhatnagar, and B. Nath, "Multi-resolution state retrieval in sensor networks," *Proc. First IEEE workshop on Sensor Network Protocols and Applications*, May. 2003. 605
9. Y. J. Zhao, R. Govindan and D. Estrin, "Residual energy scan for monitoring sensor networks," *Proc. IEEE Wireless Communications and Networking Conference*, Mar. 2002.

10. A. F. Mini, B. Nath, and A. A. F. Loureiro, "A probabilistic approach to predict the energy consumption in wireless sensor networks," *Proc. IV Workshop de Comunicação sem Fio e Computação Móvel*, São Paulo, Brazil, Oct. 23–25th 2002. 610
11. A. F. Mini, B. Nath, and A. A. F. Loureiro, "Prediction-based approaches to construct the energy map for wireless sensor networks," *Proc. 21st Brazilian Symposium on Computer Networks*, Natal, RN, Brazil, May 19–23, 2003.
12. W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *Proc. of the 33rd International Conference on System Sciences*, Jan. 2000. 615
13. J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," Technical Report 02–773, USC, Sep. 2003.
14. Michael V. Leonov, and Alexey G. Nikitin, "An efficient algorithm for a closed set of Boolean operations on polygonal regions in the plane," Preprint 46, Novosibirsk, A. P. Ershov Institute of Informatics Systems, 1997. 620
15. Herb Schwetman, "Model-based systems analysis using CSIM 18," *Proc. 1998 Winter Simulation Conference*.