

Maintaining Differentiated Coverage in Heterogeneous Sensor Networks

Xiaojiang Du

*Department of Computer Science, North Dakota State University, Fargo, ND 58105, USA
Email: xiaojiang.du@ndsu.edu*

Fengjing Lin

*Department of Computer Science, North Dakota State University, Fargo, ND 58105, USA
Email: fengjing.lin@ndsu.edu*

Received 27 November 2004; Revised 22 March 2005

Most existing research considers homogeneous sensor networks, which suffer from performance bottleneck and poor scalability. In this paper, we adopt a heterogeneous sensor network model to overcome these problems. Sensing coverage is a fundamental problem in sensor networks and has been well studied over the past years. However, most coverage algorithms only consider the uniform coverage problem, that is, all the areas have the same coverage degree requirement. In many scenarios, some key areas need high coverage degree while other areas only need low coverage degree. We propose a differentiated coverage algorithm which can provide different coverage degrees for different areas. The algorithm is energy efficient since it only keeps minimum number of sensors to work. The performance of the differentiated coverage algorithm is evaluated through extensive simulation experiments. Our results show that the algorithm performs much better than any other differentiated coverage algorithm.

Keywords and phrases: heterogeneous sensor networks, sensing coverage, differentiated coverage.

1. INTRODUCTION

Sensor networks hold the promise of facilitating large-scale, real-time data processing in complex environments. Existing research mainly considers homogeneous sensor networks, that is, all sensor nodes have identical capabilities in terms of communication, computation, sensing, reliability, and so forth. However, a homogeneous ad hoc network suffers from poor scalability. Recent research has demonstrated its performance bottleneck both theoretically (Gupta and Kumar [1] showed that the per-node throughput in a homogeneous ad hoc network is $\Theta(1/\sqrt{n})$, where n is the number of nodes), and through simulation experiments and testbed measurement [2]. In this paper, we adopt a heterogeneous sensor network model to achieve good performance and scalability. Scalability is particularly important to large-scale sensor networks with hundreds and thousands sensor nodes.

One of the fundamental problems in sensor networks is the sensing coverage problem. Sensing coverage characterizes the monitoring quality provided by a sensor network in a designated region. Energy is a paramount concern in wire-

less sensor network applications that need to operate for a long time on battery power. For example, habitat monitoring may require continuous operation for months, and monitoring civil structures (e.g., bridges) requires an operational lifetime of several years. Most sensor networks are deployed with high density (up to 20 nodes/m³ [3]) in order to prolong the network lifetime. Recent research has found that significant energy savings can be achieved by dynamic management of node duty cycles in sensor networks with high node density. In this approach, some nodes are scheduled to sleep (or enter a power saving mode) while the remaining active nodes provide continuous service. A fundamental problem is to minimize the number of nodes that remain active, while still achieving acceptable quality of service for applications.

Most existing researches consider the uniform sensing coverage problem in sensor networks, for example, PEAS [4] and OGDC [5]. In these algorithms, nodes switch to sleeping state as long as their neighbors can provide sensing coverage for them. These algorithms provide the same coverage degree for the entire network area. However, in many scenarios such as battlefields, there are certain geographic sections such as the command headquarters that need higher coverage degree than other areas. Since typical sensor nodes are unreliable devices and can fail or run out of power, and single sensing readings can be easily distorted by background

noise to cause false alarms, it is desirable to provide higher degree of coverage for critical areas. However, it is not efficient to support the same high degree of coverage for some less important areas. To handle this issue, in this paper we propose a differentiated coverage algorithm for sensor networks. Differentiated coverage means providing different degrees of sensing coverage for different areas in a sensor network according to the requirement.

The main contributions of this paper are the following. (1) We adopt a heterogeneous sensor network model to achieve good performance and scalability. (2) We propose a novel differentiated coverage algorithm for sensor networks. The rest of this paper is organized as follows. Section 2 reviews the related work in the literature. In Section 3, we introduce the differentiated coverage algorithm. Section 4 presents the simulation results. And Section 5 concludes the paper.

2. RELATED WORKS

Sensing coverage in sensor networks has been well studied. Several algorithms aim to find close-to-optimal solution based on global information. In [6], a linear programming technique is applied to select the minimal set of active nodes for maintaining coverage. In [7], sensor deployment strategies were investigated to provide sufficient coverage for distributed detection. In [4], Ye et al. presented PEAS—a probing-based sensing coverage algorithm. Tian and Georganas [8] proposed an algorithm that provides complete coverage using the concept of “sponsored area.” Both [4, 8] only consider the metric in terms of the total amount of energy consumed regardless of the distribution of the energy among the nodes. The unbalanced energy dissipation causes some nodes to die much faster than others; therefore, the half-life of the network is dramatically reduced in the unbalanced approach. In [5], Zhang and Hou showed that coverage with minimal overlap is achieved when three sensor nodes form an equilateral triangle, and they proposed a localized density control algorithm OGDC based on the result.

In [9], Yan et al. proposed a differentiated surveillance algorithm for sensor networks. In the algorithm, the sensor network is covered by uniformly distributed grid points, and the coverage of the network is converted to the coverage of all the grid points. Each sensor node chooses a random time reference point Ref within $[0, T]$ (T is the operation round), and broadcasts its location and Ref to the neighbors. Then each node locally decides its schedule of sleep and work, based on the Ref and location information of the neighbors that cover the same grid point. Since each sensor node usually covers several grid points, a scheme is needed to combine the schedules for covering multiple grid points. In [9], the final schedule of a sensor node is the union of its schedules for all the grid points that it can cover. However, since the Ref point is randomly selected, the probability of several Ref points close to each other is very small. In other words, the multiple Ref points are usually scattered across the $[0, T]$ time period, and thus the union of schedules usually leads

to a very long working duration, which means that a sensor node will work for most of time. For example, if a sensor node needs to cover three grid points, and the schedule for each grid point is $[0, T/3]$, $[T/2, 2T/3]$, and $[2T/3, T]$, respectively, then the union of the above schedules has a duration of $5T/6$, which means the sensor node needs to work for $5/6$ of the time. Thus, the differentiated surveillance algorithm in [9] is not efficient.

Recently deployed sensor network systems are increasingly following heterogeneous designs, incorporating a mixture of sensors with widely varying capabilities [10]. For example, in a smart home environment, sensors may be powered by AA batteries, AAA batteries, or even button batteries. Researchers have studied various issues in heterogeneous sensor networks. In [11], Mhatre et al. studied the optimum node density and node energies to guarantee a lifetime in heterogeneous sensor networks. Duarte-Melo and Liu analyzed energy consumption and lifetime of heterogeneous sensor networks in [12].

In this paper, we adopt a heterogeneous sensor network model to overcome the poor scalability and performance bottleneck of homogeneous sensor networks. We propose a novel differentiated coverage algorithm for wireless sensor networks.

3. THE ENERGY-EFFICIENT DIFFERENTIATED COVERAGE ALGORITHM

In this section, we present our differentiated coverage (DC) algorithm for heterogeneous sensor networks. We consider a heterogeneous sensor network (HSN) consisting of two types of nodes: a small number of powerful high-end sensors (H-sensors) and a large number of low-end sensors (L-sensors). One can build a heterogeneous sensor network by distributing H-sensors and L-sensors at the same time, or by adding a small number of H-sensors into an existing homogeneous sensor network. H-sensors and L-sensors are assumed to be uniformly and randomly distributed in the field. Both H-sensors and L-sensors are assumed to know their location information. Sensor nodes can use location services such as those in [13, 14] to estimate their locations, and no GPS receiver is required at each node. The operation of a sensor network is divided into several rounds, with each round being the same duration T . We assume that the L-sensor’s transmission range r_t is at least twice of its sensing range r_s , that is, $r_t \geq 2r_s$. This is true for many sensor nodes, including Mica II sensor [15], and so forth.

In Section 3.1, we describe the cluster formation scheme in HSN. In Section 3.2, we present the scheme that provides uniform coverage in a sensor network. The uniform coverage problem is a special case of the differentiated coverage problem. In Section 3.3, we present the differentiated coverage (DC) algorithm.

3.1. Cluster formation in HSN

During the initialization phase, all H-sensors broadcast Hello messages to nearby L-sensors with a random delay. The random delay is to avoid the collision of Hello messages

from two neighbor H-sensors. The Hello message includes the ID of the H-sensor and its location. Since the locations of H-sensors are random, H-sensors use the maximum transmission power to broadcast the Hello messages. With enough number of H-sensors uniformly and randomly distributed in the network, most L-sensors can receive Hello messages from multiple H-sensors, and most H-sensors can hear Hello messages from neighbor H-sensors. Then each L-sensor chooses the H-sensor whose Hello message has the best signal strength as the cluster head. Each L-sensor also records other H-sensors from which it receives the Hello messages, and these H-sensors are listed as backup cluster heads in case the primary cluster head fails.

If an L-sensor does not hear any Hello message during the initialization phase (e.g., T seconds after deployment), the node will broadcast an Explore message. When the neighbor L-sensors receive the Explore message, they will response with an Ack message after a random delay. The Ack message includes the location and ID of the sender's cluster head. An L-sensor will not send Ack message again if it overhears an Ack response from another neighbor. This mechanism reduces the number of response messages and thus the consumed energy. Then the L-sensor can select a cluster head based on the Ack message. This ensures that each L-sensor finds a cluster head.

The sensor network is divided into multiple clusters, where H-sensors serve as the cluster heads. For simplicity, assume the network is a two-dimensional plane, then each L-sensor will select the closest H-sensor as the cluster head (except when there is an obstacle in between), and this leads to the formation of Voronoi diagram where the cluster heads are the nuclei of the Voronoi cells. An example of the cluster formation is shown in Figure 1. The large rectangle nodes in Figure 1 are H-sensors and the small square nodes are L-sensors. During initialization, each H-sensor also records the locations of the neighbor H-sensors (based on the Hello messages), and H-sensors can calculate the boundary of the Voronoi cells based on the locations of neighbor H-sensors.

3.2. The uniform sensing coverage scheme

We first present the scheme that provides uniform coverage in a sensor network. A grid is installed in the sensor network, and the grid points are uniformly distributed in the network. An example is shown in Figure 2, where the crosses are the grid points. Assume all H-sensors know the location of a reference grid point and the grid size (e.g., storing such information before deployment), then H-sensors know the locations of all the grid points. An H-sensor can determine which grid points are covered by an L-sensor based on its location and sensing range. We will first study the problem of covering all the grid points while minimizing sensor energy consumption. When a reduced sensing range is used for node scheduling, it can be shown that covering all grid points is equivalent to covering the whole field. The reduced sensing range should satisfy $r_c < r_a - d/\sqrt{2}$, where r_c , r_a , and d are the reduced sensing range, the actual sensing range, and the grid side length, respectively. We will not present the details here. In [9], Yan et al. also showed the above equivalence.

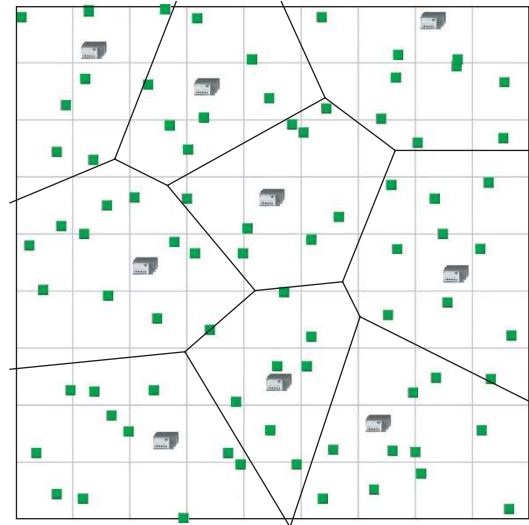


FIGURE 1: Voronoi cells in an HSN.

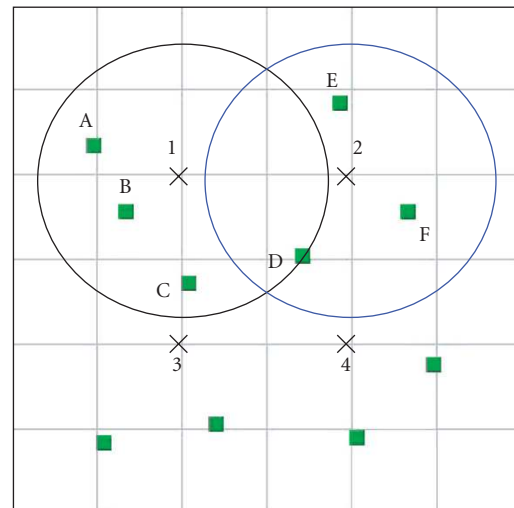


FIGURE 2: Coverage for grid points.

The goal is to design a node-scheduling scheme that ensures all the grid points have the required coverage, while at the same time minimize the total energy consumption in the network and balance node energy consumption.

The node scheduling is processed in each cluster independently. In a sensor network, all the grid points are numbered in a certain way, for example, from top to down and from left to right. In each cluster, the node scheduling is processed according to the increasing order of grid point number. That is, the schedule of sensors covering grid point 1 is determined first, then the schedule of sensors covering grid point 2 is determined, and so on.

In the sensing coverage scheme, a cluster head determines the node scheduling for all the L-sensors in its cluster. After initialization, each L-sensor sends its location information to the cluster head. Since the location of the cluster head is known from the Hello message, a greedy geographic routing protocol GPSR [10] is used for intra-cluster routing.

An L-sensor sends the packet to the active neighbor that has the shortest distance to the cluster head, and the next node performs the similar thing, until the packet reaches the cluster head. Since nodes within a cluster are not far away from the cluster head, the greedy geographic routing should be able to route packets to cluster head with high probability. The chance of having a void during greedy geographic routing (i.e., all the neighbors have longer distance to the cluster head than the node itself) is small. In case such a thing happens, several recover schemes can be used to solve the problem, for example, GPSR [10] and GOAFR [16] route a packet around the faces of a planar subgraph extracted from the original network.

After a certain time, a cluster head should receive the location information from all the L-sensors in its cluster, then the cluster head starts determining node schedule for each grid point in the cluster, according to the increasing order of the grid point number. In the following, we will use the example in Figure 2 to illustrate the scheme that determines node schedule for a grid point. Based on the locations of L-sensors, the cluster head (say H) knows which L-sensors cover a grid point, that is, L-sensors within the circle centered at the grid point with radius r_s (sensing range). In Figure 2, three L-sensors (D, E, F) cover grid point 2.

H counts the total number (say k) of L-sensors that cover grid point 2. An ideal schedule for the k sensors should be that each L-sensor works for T/k time and sleeps for $T - T/k$ time in a round T . This will ensure that the total energy consumption is minimized and each node has similar remaining energy. However, a sensor node may also need to cover other grid points, and some of them may already have one or more assigned working slots. H considers the assigned working slots of each L-sensor and tries to assign a time slot that has the maximal overlap with the existing working slots. For example, if node D already has a working slot of $[0, T/4]$ (for covering grid point 1), then H can assign the working slot of $[0, T/3]$ to D. Thus D only needs to be active during $[0, T/3]$ and covers both grid points 1 and 2. If there is conflict, then a node may have an additional (or overlapped) working slot besides its existing working slots.

After determining the node schedule for all grid points in the cluster, the cluster head H includes the working slots for all the L-sensors in one packet, and broadcast the packet to all L-sensors in its cluster. Each L-sensor records its working slots as well as the working slots of its neighbors. The neighbor working slots information is used by the greedy geographic routing—GPSR [10]. When an L-sensor wants to send a packet, it sends the packet to an active neighbor that has the shortest distance to the cluster head.

Periodically, all L-sensors wake up and enter a listen state, and cluster heads reschedule working slots for the L-sensors. This is to ensure that the coverage algorithm is robust to sensor failures. For example, at the end of each round, all L-sensors wake up and enter a listen state, and each cluster head broadcasts a rescheduling message to the L-sensors in its cluster. Then each alive L-sensor sends a packet to the cluster head, including its location and node ID. Cluster heads determine node schedule based on the coverage algorithm.

To ensure the sensing covering scheme works well, L-sensors in a cluster need to be synchronized. However, L-sensors from different clusters need not be synchronized, since the node scheduling is determined in each cluster independently. For our heterogeneous sensor network model, a simple scheme can be used to synchronize the L-sensors within a cluster. Each time before a cluster head H broadcasts the node scheduling, H broadcasts a short synchronization message including its local time, and all the L-sensors can synchronize their time with cluster head H.

3.3. The differentiated coverage algorithm

The above sensing coverage scheme can be easily extended to provide differentiated coverage for sensor networks. If we want to adjust the sensing coverage degree of a certain area to an arbitrary degree c , the cluster head will correspondingly increase or decrease the work time for each L-sensor in the area. For a grid point covered by k sensor nodes, the work time for each sensor node is T/k (in each round T) to provide degree-1 coverage. For degree- c coverage, the work time for each sensor node is cT/k . Thus, it is easy to provide differentiated coverage for a sensor network by using our scheme. The differentiated coverage algorithm is presented in Algorithm 1. In the following, we use the example in Figure 2 to describe the details of the Differentiated Coverage algorithm.

A cluster head H determines the schedules of all L-sensors in its cluster. For a grid point (say point 2 in Figure 2) in its cluster, H first counts the total number (say k) of L-sensors that cover this grid point. If $k \leq c$, then all the L-sensors that cover point 2 need to be active for all time. If $k > c$, H will determine the working slots for each L-sensor. An ideal schedule for the k sensors should be that each L-sensor works for cT/k time and sleeps for $T - cT/k$ time in a round T . This ensures that the total energy consumption is minimized and each node has similar remaining energy. However, a sensor node may also need to cover other grid points, and some of them may already have one or more assigned working slots. H considers the assigned working slots of each L-sensor and tries to assign a time slot that has the maximal overlap with the existing working slots.

In the scheduling algorithm, each round T is divided into k equal time slots, that is, $[0, T/k], [T/k, 2T/k], \dots, [(k-1)T/k, T]$, and these time slots are indexed by $1, 2, \dots, k$. A set I is used to include the indexes of the available time slots. Initially set I includes all the time slots, that is, $I = \{1, 2, \dots, k\}$.

Each L-sensor is assigned with c time slots for a required coverage degree c , and this is done by the second FOR loop in Algorithm 1. In each iteration of the second FOR loop, one time slot is selected for each of the k L-sensors. To avoid assigning the same time slot to an L-sensor twice, a set B_l is used to store the selected time slots for an L-sensor l . In the third FOR loop, for each L-sensor l , the algorithm finds a time slot j that belongs to set I but not set B_l while maximizing the overlap with node l 's existing working slots (which are used to cover other grid points). If $I \subseteq B_l$, that is, all the time slots left in set I are also in set B_l , then a time slot not in set B_l is randomly selected. After selecting all the c time slots

Notations:

H is the cluster head.

U is the set of grid points in H's cluster.

u is a grid point in H's cluster, that is, $u \in U$.

$L(u)$ is the set of L-sensors that cover grid point u .

$k = |L(u)|$ is the total number of L-sensors that cover grid point u .

c is the required coverage degree.

Each round T is divided into k equal time slots, that is, $[0, T/k], [T/k, 2T/k], \dots, [(k-1)T/k, T]$, and these time slots are indexed by $1, 2, \dots, k$.

I is the set of indexes of the available time slots.

Initially $I = \{1, 2, \dots, k\}$.

B_l is the set of selected time slots for a L-sensor l .

Initially $B_l = \emptyset$.

The following scheduling algorithm runs in each cluster head.

```

FOR each grid point  $u \in U$ 
// Iterating  $c$  times for a required coverage degree  $c$ .
FOR  $i = 1$  to  $c$ 
Resetting the available time slot set  $I = \{1, 2, \dots, k\}$ .
// For each L-sensor  $l \in L(u), 1 \leq l \leq k$ .
FOR  $l = 1$  to  $k$ 
IF  $I \supset B_l$ , find a time slot  $j$  that satisfies the following 3
conditions:
(1)  $j \in I$ ; // Selecting  $j$  from available slots.
(2)  $j \notin B_l$ ; //  $j$  should not be the same as any
// previously selected slot.
(3)  $j$  has the maximal overlap with  $l$ 's existing working slots.
ELSE // That is,  $I \not\supset B_l$ 
A time slot not in  $B_l$  is randomly selected.
ENDIF
Adding time slot  $j$  to set  $B_l$ .
Removing  $j$  from the available time slot set  $I$ , that is,
 $I = I - \{j\}$ .
END // End of the third FOR loop.
END // End of the second FOR loop.
// Adding the selected slots to the working slots.
FOR each L-sensor  $l \in L(u)$ 
Adding set  $B_l$  to  $l$ 's working slots.
END
END // End of the first FOR loop.

```

ALGORITHM 1: The differentiated coverage algorithm.

for each L-sensor, the selected time slots are added into the working slots of each L-sensor.

In [5], Zhang and Hou prove that the radio range being at least twice of the sensing range is both necessary and sufficient to ensure that coverage implies connectivity. In [17], Wang et al. also show the similar result. Our sensing coverage algorithm ensures the coverage in a sensor network, thus guarantees connectivity in the network when $r_t \geq 2r_s$.

4. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the differentiated coverage (DC) algorithm, and compare DC with another differentiated coverage algorithm in [9], which we refer to as differentiated surveillance (DS) algorithm. The following metrics are used to show the energy saving

and efficient coverage provided by DC algorithm: (1) total amount of energy consumption, (2) energy variation among nodes, (3) sensing coverage over time, (4) energy consumption for differentiated coverage, and (5) the number of working nodes.

We implemented DC algorithm in QualNet. For comparison, DS algorithm was also implemented in QualNet. The underlying medium access control protocol is IEEE 802.11 DCF. We adopt the same energy model as in TTDD [18]. A sensor node's transmitting, receiving, and idling power consumption rates are 0.66 W, 0.395 W, and 0.035 W, respectively, [18]. In DC, GPSR [10] is used as the routing protocol for transmissions from L-sensors to cluster heads. The default simulation testbed has 1 sink and 300 L-sensors uniformly, randomly distributed in a $200 \text{ m} \times 200 \text{ m}$ area. The sensing range and communication range of an L-sensor is 10 m and 25 m, respectively. The grid size d is 4 m. For DC, there are additional 10 H-sensors in the network. Although H-sensors also provide sensing coverage, for fair comparison we do not count the coverage from H-sensors in the following experiments.

Each simulation runs for 2000 seconds, and each experiment runs for 10 times with different node deployments and different random seeds. Each round T is set as 500 seconds, so there are 4 rounds in each simulation. In DC algorithm, all L-sensors enter listen state after every 500 seconds (one round) and the L-sensors are rescheduled by the cluster heads. In the following tests, the communication cost for transferring data packet is not included in the energy consumption, since it is highly application dependent. In Sections 4.1, 4.2, and 4.3, the uniform coverage case is considered, and the differentiated coverage is considered in Sections 4.4 and 4.5.

4.1. Total energy consumption

In Figure 3, we compare the total energy consumed for different node densities using DC algorithm and DS algorithm. The total number of L-sensors varies from 200 to 500 with an increasing of 50. The number of H-sensors in DC does not change. The total energy consumption when all sensor nodes are working is also plotted in Figure 3. The total energy consumption in DC also includes energy consumption of H-sensors.

From Figure 3, we can see that DC consumes much less energy than both DS and the "all working" case. DS consumes less energy than "all working" when sensor density is high. For the "all working" case, the total energy consumed is close to a linear function of the sensor number, and it increases very fast as the number of sensors increases, while the energy consumptions under DS and DC increase slowly when the number of sensors becomes large. In DS and DC algorithms, only a portion of sensors (that are enough to cover the area) are active at any time. When sensor density increases, the required coverage degree does not change, thus their energy consumptions do not increase much. The small increase of the energy consumptions in DS and DC mainly comes from the communication overhead to determine the node schedule.

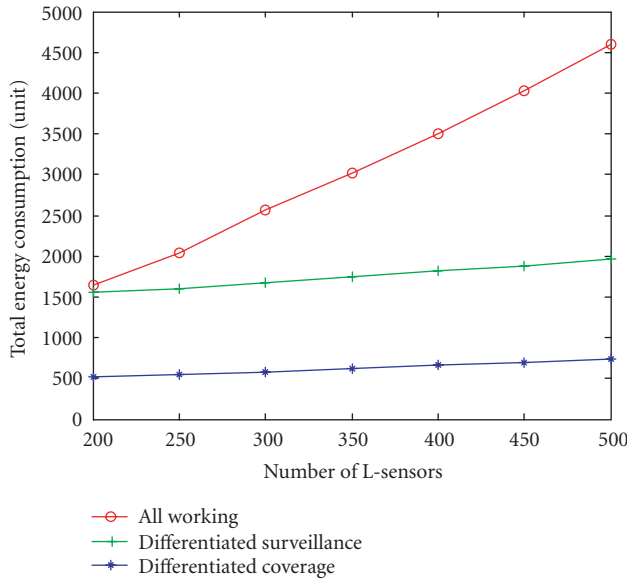


FIGURE 3: The total energy consumption.

Figure 3 also shows that the total energy consumed in DC is only about 1/3 of that in DS. In DS, each node decides its own schedule, and the integrated schedule is the union of the schedules for all the grid points that it can cover. For a sensor node using DS algorithm, the working slot for covering a grid point is randomly selected. Because of the randomness, the working slots for different grid points are usually different, thus the union of the schedules for different grid points leads to a long working duration. For example, consider a node C that covers three grid points. If the work-time for the three grid points is $[0, T/4]$, $[T/3, 2T/3]$, and $[3T/4, T]$, respectively, then node C will work for $5T/6$ in each round (T seconds). On the other hand, in DC algorithm, the cluster head considers the existing working slot when it makes schedule for covering the current grid point and tries to maximize the overlap between the existing and new working slots, and this dramatically reduces the total work-time for each node. For example, in the above example, the sensor node C could be scheduled to work only during $[T/3, 2T/3]$ to cover the three grid points, then the work duration is only $T/3$, much less than $5T/6$ in DS algorithm.

4.2. Balancing node energy consumption

In this study, we investigate the energy consumption of individual sensor nodes. Specifically, we want to check if the energy consumption is balanced among different sensor nodes. We measure the average value (Ave) and standard deviation (Std) of energy consumed by each node under different node densities, and the results are reported in Figure 4.

Figure 4 shows that in both DS and DC algorithms, the average energy consumption for an individual node decreases as the network node density increases. This is reasonable since more nodes means less work time for each node, and less energy consumed. The average energy consumption of each node in DC is always lower than that in

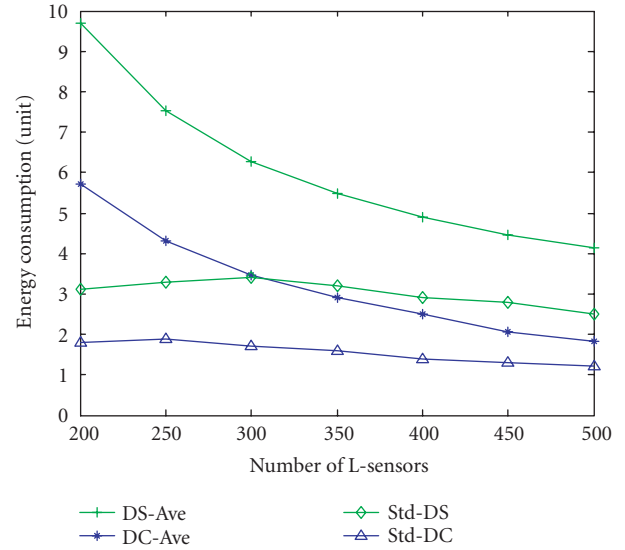


FIGURE 4: Average and standard deviation of node energy consumption.

DS, and this shows that DC is more energy efficient than DS. The reason is already stated in Section 4.1. In addition, from Figure 4 we can see that the standard deviation in DC is also smaller than DS, which means the node energy consumption is better balanced in DC than in DS.

4.3. Coverage over time

The coverage of a sensor network at different time instances after network deployment is an important performance. We measure the sensing coverage at different time by running the simulation for a longer time period—6000 seconds. Each sensor node has a fixed energy supply and it dies when the energy supply runs out. We test the sensing coverage for two different node densities: 300 nodes and 450 nodes. The test results are reported in Figure 5.

Figure 5 shows that before 2000 seconds, the sensing coverages under DC and DS are close to each other. When the simulation time is larger than 2000 seconds, the coverage under DS algorithm drops rapidly as time increases, and the sensing coverage is less than 30% at 6000 seconds. On the other hand, the sensing coverage under DC algorithm drops slowly as time increases. At 6000 seconds, the coverage under DC is still above 80% for the 450-node network, and close to 70% for the 300-node network. Sensor nodes using DS algorithm have much longer work (active) time and die out earlier than nodes in DC algorithm. That is why the sensing coverage under DS drops very fast, and the sensor network using DS can only provide low coverage after a long period of time.

4.4. Energy consumed for differentiated coverage

In this subsection, we measure the performance of DC algorithm for differentiated coverage and compare the total energy consumed in DC with DS for different desired coverage degrees. In this experiment, different areas in the network

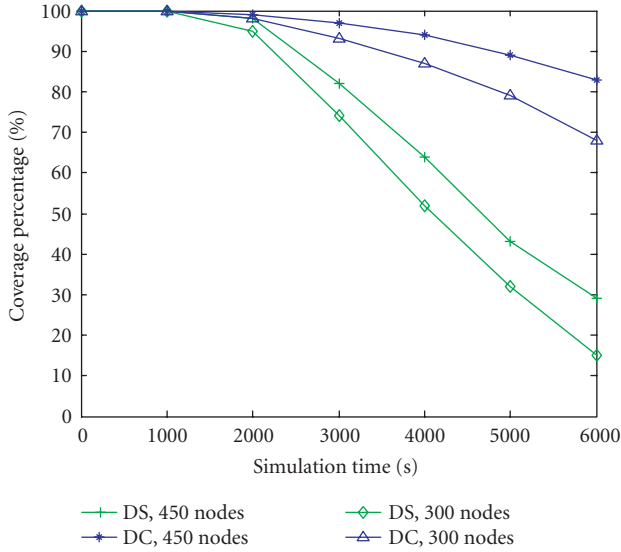


FIGURE 5: Sensing coverage over time.

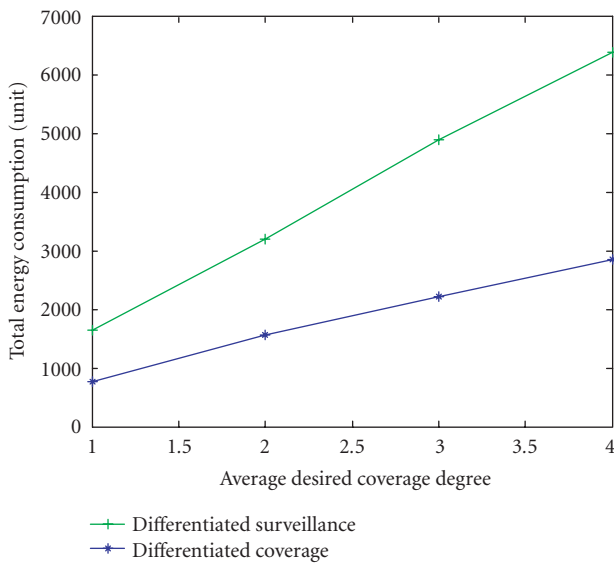


FIGURE 6: Total energy consumption for differentiated coverage.

have different desired coverage degrees. To make the comparison meaningful, the same differentiated coverage requirements are used for both DC and DS algorithms, that is, the same desired coverage degree is used for the same grid point in both DC and DS. The average required coverage degree (over the network) tested includes 1, 2, 3, and 4. The test results are reported in Figure 6, where a sensor network with 600 L-sensors is used. From Figure 6, we can see that the total energy consumption increases linearly in the desired coverage degree, in both DC and DS algorithms. The energy consumed at a higher average coverage degree- k is a little bit less than k times the energy consumed at coverage degree-1, because the communication overhead does not increase proportionally as the desired coverage degree. Figure 6 shows

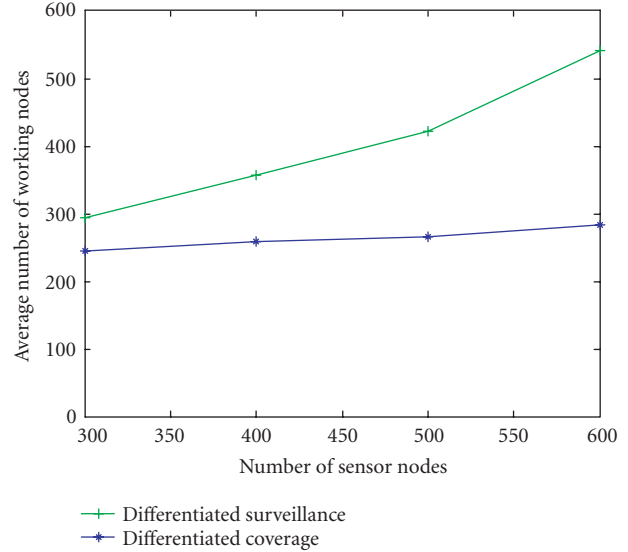


FIGURE 7: The number of working nodes for different node densities.

that the total energy consumed in DC is much lower than that in DS, for all the desired coverage degrees tested.

4.5. The number of working nodes

In order to reduce the total energy consumption in sensor networks, the number of active sensors should be kept to the minimum. The average number of working nodes is measured for different sensor node densities, varying from 300 to 600. The results under DS and DC are plotted in Figure 7, where the required average coverage degree is two. Figure 7 shows that the number of working nodes in DC does not change much as sensor density increases. In DC, cluster heads combine node working slots (for covering different grid points) together and thus dramatically reduces the total work time of a node, which in turn reduces the average number of working nodes in the network. Since the required coverage degree does not change, the number of working nodes in DC does not change much. In DS, the work time of a node is the union of schedules for covering multiple grid points, and in many cases it is much longer than the work time in DC. Thus, the average number of working nodes in DS is larger than that in DC. When node density increases, the higher node density is not well utilized by DS because of the randomness in setting work time. As node density increases, there are more nodes in DS having long work time, so the difference of working node number between DS and DC becomes larger.

5. CONCLUSIONS

In this paper, we adopted a heterogeneous sensor network model to overcome the poor scalability and performance bottleneck of homogeneous sensor networks. A small number of high-end sensors are mixed together with a large number of low-end sensors to form a heterogeneous sensor network. We proposed the Differentiated coverage (DC)

algorithm for heterogeneous sensor networks, which can provide different coverage degrees for different areas. In DC, cluster heads integrate sensor's work time for covering multiple grid points and dramatically reduce the total active time for each sensor. Various energy consumptions and sensing coverage of DC algorithm are evaluated through simulation experiments and compared with another differentiated coverage algorithm—DS. Our test results show that DC algorithm performs much better than DS algorithm.

REFERENCES

- [1] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, 2000.
- [2] K. Xu, X. Hong, and M. Gerla, "An ad hoc network with mobile backbones," in *Proc. IEEE International Conference on Communications (ICC '02)*, vol. 5, pp. 3138–3143, New York, NY, USA, April–May 2002.
- [3] E. Shih, S.-H. Cho, N. Ickes, et al., "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proc. 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 272–287, Rome, Italy, July 2001.
- [4] F. Ye, G. Zhong, S. Lu, and L. Zhang, "PEAS: a robust energy conserving protocol for long-lived sensor networks," in *Proc. 23rd International Conference on Distributed Computing Systems (ICDCS '03)*, pp. 169–177, Providence, RI, USA, May 2003.
- [5] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *International Journal of Wireless Ad Hoc and Sensor Networks*, vol. 1, no. 1-2, pp. 89–124, 2005.
- [6] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Trans. Comput.*, vol. 51, no. 12, pp. 1448–1453, 2002.
- [7] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for target detection," in *Proc. 1st International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 42–48, Atlanta, Ga, USA, September 2002.
- [8] D. Tian and N. D. Georganas, "A coverage-preserved node scheduling scheme for large wireless sensor networks," in *Proc. 1st International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 32–41, Atlanta, Ga, USA, September 2002.
- [9] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *Proc. 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 51–62, Los Angeles, Calif, USA, November 2003.
- [10] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proc. 6th Annual International Conference on Mobile Computing and Networking (MobiCom '00)*, pp. 243–254, Boston, Mass, USA, August 2000.
- [11] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "A minimum cost heterogeneous sensor network with a lifetime constraint," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 4–15, 2005.
- [12] E. J. Duarte-Melo and M. Liu, "Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 1, pp. 21–25, Taipei, Taiwan, November 2002.
- [13] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Pers. Commun.*, vol. 7, no. 5, pp. 28–34, 2000.
- [14] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 3, pp. 1655–1663, Anchorage, Ark, USA, April 2001.
- [15] Crossbow Technology San Jose Calif USA, <http://www.xbow.com/>.
- [16] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proc. 4th ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc '03)*, pp. 267–278, Annapolis, Md, USA, June 2003.
- [17] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proc. 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 28–39, Los Angeles, Calif, USA, November 2003.
- [18] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A two-tier data dissemination model for large scale wireless sensor networks," in *Proc. 8th Annual International Conference on Mobile Computing and Networking (MobiCom '02)*, pp. 148–159, Atlanta, Ga, USA, September 2002.

Xiaojiang Du is an Assistant Professor in the Department of Computer Science at North Dakota State University. He received his B.E. degree in electrical engineering from Tsinghua University, Beijing, China in 1996, and his M.S. and Ph.D. degrees in electrical engineering from University of Maryland, College Park, in 2002 and 2003, respectively. His research interests are wireless sensor networks, mobile ad hoc networks, wireless networks, computer networks, network security, and network management. He is a technical program committee member for several international conferences (including IEEE ICC 2006, Globecom 2005, BroadNets 2005, WirelessCom 2005, IPCCC 2005, and BroadWise 2004). He is a Member of IEEE.



Fengjing Lin is currently a Ph.D. student in the Department of Computer Science at North Dakota State University. She received her B.S. degree in education from Jia-Ying University, China, in 1999, and her M.S. degree in computer science from Southeastern University, Washington, DC, in 2003, respectively. Her research interests are wireless sensor networks, mobile ad hoc networks, and computer networks.

