Effective Coverage and Connectivity Preserving in Wireless Sensor Networks

Nurcan Tezcan Wenye Wang
Department of Electrical and Computer Engineering
North Carolina State University
Email: {ntezcan,wwang}@eos.ncsu.edu

***Abstract*— In this paper, we address the problem of finding an optimal coverage set by effectively eliminating redundant nodes with guaranteed connectivity without using centralized control and accurate location information. Using a fully distributed approach, we propose an effective redundant node elimination method that considers even the smallest overlapping regions to establish a coverage set. Further, an extension scheme is presented that finds the minimum number of sensors among the coverage set, where the network connectivity is guaranteed. We present the simulation results to illustrate achievable coverage set while preserving connectivity, and energy saving to verify our approaches.**

I. Introduction

In recent years, wireless sensor networks (WSN) have been used in many event-critical applications, such as habitat monitoring, health applications, battlefield monitoring, wild animal protection. To support such applications, an important problem that should be addressed is *how well a given area can be monitored to catch all the events by the WSN* - an issue often related to QoS, and known as *coverage* [11], [18]. On the other hand, sensors have very limited energy supply. They are usually hard to recharge their batteries after deployment, either because of the number of nodes is too large, or because the deployment area is hostile. Hence, previous studies [1], [15]–[17] focused on the problem of finding an optimal set of sensors maintaining the coverage and connectivity of the network to achieve energy-efficient communication. By this way, only a group of sensors actively work for event detection, data transmission or aggregation, where redundant nodes save energy for the next rounds that makes network lifetime prolonged in an efficient way.

In particular, finding an optimal coverage or connectivity set can be implemented using either *centralized* [1], [5], [15] or *distributed* algorithms [4], [8], [14], [16], [17]. *Centralized solutions* basically organize sensors to preserve the sensing coverage without leaving blind points in the sensing field by use of a central authority (the sink) and global location information of sensors. Even though they find near-optimal solutions, they can not be applied to high-dense sensor networks with large number of sensor nodes due to the communication expense of having global information at the sink. Therefore, many *distributed solutions* have been proposed [4], [8], [14], [16], [17] where coverage is achieved by forming connected set-covers in a distributed manner for each query [8]. In [5], rather than sensing field, a set of targets with known locations are necessarily covered by each set cover. Therefore, only active nodes in the set-cover send and receive data. On the other hand, in [16], a distributed scheduling algorithm has been proposed where each node turns itself off using local neighbor information where all nodes have identical sensing range that is the same as their transmission range. However, the sensing range of a sensor node might be approximately in between *1-30 m*, whereas the transmission range of that sensor might be in between *50-300 m* [18].

In this paper, we focus on the problem of *distributed calculation* of coverage of each individual sensor to effectively eliminate redundant nodes using relative location information, i.e., *virtual coordinates*. Although, location information can be readily obtained by GPS-supported sensors or localization schemes proposed for sensor networks [6], [9], it is usually costly and might be quite misleading in some cases. For example, even though two nodes are geographically close to each other, signal strength may not be strong due to the obstacles between them. Thus, rather than exact location information of sensors, we use received signal strength (RSS) measurements-based virtual coordinates and virtual distances between sensor nodes while determining their coverage boundaries [13]. In particular, obstacles may cause ineffective detection of events [7]. Therefore, using virtual distances may also help us to reduce the effect of irregularity in sensing ranges due to obstacles, such as buildings, and trees.

We summarize the contributions of this paper are as follows. First, we present a distributed coverage calculation algorithm that finds overlapping sensing regions to effectively discover redundant sensor nodes using virtual coordinates. After discovering the possible redundant nodes, an energy-aware algorithm is used to construct optimal-coverage set which can be used for node scheduling, topology control, etc. Third, an extensive method is proposed to find minimum number of dominating sensors among the coverage set, where the network connectivity is guaranteed. In this sense, the routing problem is simultaneously solved with the connected dominating set which forms a connected backbone to carry out the data traffic to the sink, thus no extra routing protocols are needed. Finally, we carry out performance evaluation and demonstrate the simulation results to illustrate achievable set of sensors for coverage while preserving connectivity.

The rest of paper is organized as follows. In Section II, we explain the network model and some assumptions and terms that will be used in the paper. We describe the distributed coverage calculation algorithm and redundant node discovery in Section III. Using this algorithm, effective coverage and connectivity sets constructions are presented in Section IV. Following, simulation results are presented in Section V. Finally, Section VI concludes the paper.

II. Preliminaries

Let $\mathbf{S} = \{s_1, s_2, s_3, \ldots, s_N\}$ be the finite set of sensors, distributed randomly in a two-dimensional area $\mathbf{A}$, where there are sufficient sensors to monitor the field. Each sensor s_i has a unique *identifier* (such as MAC address). All sensor nodes know their *sensing range*, denoted by r, which is assumed to be identical for all sensors, and their *transmission range*, t. In this paper, we assume that transmission range is greater or equal to the sensing range, i.e., $r \leq t$ [18]. We use *virtual coordinates* and *virtual distances* given in [13]. A sensor discovers its neighbors in its transmission range by periodically sent $hello$ messages and collect received signal strength (RSS) measurements of its neighbors. By this way, each sensor places its one-hop neighbors to a virtual coordinate system centered at itself [13] and calculates the virtual distances.

The following notations will be used in the paper:

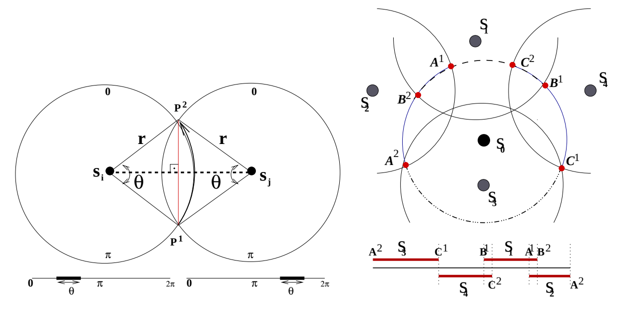
1) $\mathcal{S}_i$, the *sensing region* of a node s_i is a circular area centered at (vx_i, vy_i) and radius of r, where (vx_i, vy_i) is the virtual coordinates of s_i.
2) $\mathcal{T}_i$, *transmission neighbors* of sensor s_i with which s_i can communicate directly.
3) $\mathcal{C}_i$, *coverage neighbors* of sensor s_i which have a common sensing area with s_i. We classify the coverage neighbors of a sensor as 1-hop and 2-hop based on virtual distances.
4) For each sensor pairs s_i and s_j that have common sensing area, we associate a triple $(P_{ij}^1, P_{ij}^2, \theta_{ij})$ where (P_{ij}^1, P_{ij}^2) are the two intersections between s_i and s_j arranged in the counterclockwise order, and θ is the effective angle.

Let us denote the virtual distance between s_i and s_j by $vd(i,j)$, i.e., $vd(i,j) = \sqrt{(vx_i - vx_j)^2 + (vy_i - vy_j)^2}$. If $vd(i,j) < r$, then sensor s_i records the sensor s_j as its *1-hop coverage neighbor*, which means the s_j is inside its sensing range, $s_j \in \mathcal{C}_i^{1-Neigh}$. If $vd(i,j) < 2r$, then sensor s_i takes the sensor s_j as its *2-hop coverage neighbor*, that is not in its sensing range but shares a sensing area in common, then, $s_j \in \mathcal{C}_i^{2-Neigh}$, where $\mathcal{C}_i^{1-Neigh} \cup \mathcal{C}_i^{2-Neigh} = \mathcal{C}_i \subseteq \mathcal{T}_i$.

Definition 1: Let $\mathcal{S}_i$ be the sensing region of sensor s_i. If $\mathcal{S}_i \subseteq \bigcup_{s_j \in \mathcal{C}_i} \mathcal{S}_i \cap \mathcal{S}_j$, we call sensor s_i is a *redundant* sensor, since its sensing region can be covered by its coverage neighbors, $\mathcal{C}_i$.

A subset of sensors, $\mathbf{C} \subseteq \mathbf{S}$ is called a *coverage set* if the union of the sensing regions of the $s_i \in \mathbf{C}$ covers the entire field $\mathbf{A}$, that is $\mathbf{A} \subseteq \bigcup_{s_i \in \mathbf{C}} \mathcal{S}_i$. We consider a sensor node to be an *essential* node (E-node) in $\mathbf{C}$ if $s_i \in \mathbf{C}$. Otherwise, it is a *redundant* node (R-node). Our goal is to construct a coverage set having minimum number of E-nodes.

After constructing the coverage set, we find a subset of the coverage set, called *dominating coverage set*, $\mathbf{D}$, such that each E-node $\in (\mathbf{C}/\mathbf{D})$ can directly communicate with one of the sensors in $\mathbf{D}$. We consider a sensor s_i to be an *essential dominating* (ED) node if $s_i \in \mathbf{D}$.



(a) (b)

Fig. 1. Illustration of distributed coverage problem.

Next, we present the distributed coverage calculation and redundant node discovery algorithm to construct the coverage and dominating coverage sets.

III. Coverage Calculation and Discovery of Redundant Nodes

One of the main challenges in distributed coverage calculation is to determine whether the sensor is redundant using its 1-hop and 2-hop coverage neighbors. In this context, we use the term of *coverage calculation* to calculate whether a sensing region $\mathcal{S}_i$ of a sensor s_i is fully covered. To the best of our knowledge, existing coverage methods [4], [17] only considered 1-hop coverage neighbors, i.e., neighbors inside the sensing range. In this paper, we also take into account the 2-hop neighbors that share area in common as well to decrease the lower bound for the number of coverage neighbors while effectively eliminating redundant nodes.

Next, we will show proposed coverage calculation steps that considers both 1-hop and 2-hop coverage neighbors. The first step, called *perimeter-test*, checks whether there are enough coverage neighbors such that all points in the perimeter should be within a sensing range of a coverage neighbor. This is a necessary condition based on the assumption of densely deployed nodes [11]. The second step is called *center-test* in which it is examined whether the center of a sensor's coverage can be covered by at least one of its neighbors. The third step is called *distance test*, those coverage neighbors must be close enough to the sensor, so that there may not be uncovered area inside the sensing region.

Here, we explain these three conditions in detail as follows:

Perimeter-test: A sensor first determines whether the perimeter of its sensing region is covered. By examining each 1-hop and 2-hop coverage neighbor, a sensor can determine if intersected arcs in total are sufficient to enclose its perimeter from 0 to 2π [11]. If the perimeter is enclosed, we refer that "perimeter-test" is passed. To do this, we simply determine the intersection points of the arcs and scan the perimeter as illustrated in Fig. 1 (a).

We show an example of the perimeter-test in Fig. 1 (b), where sensor s_0 has 4 coverage neighbors. Using their virtual coordinates, sensor s_0 can find the intersection points of the arcs [2]. Let the line segment from 0 to 2π in Fig. 1 (b) denote the perimeter of s_0. After we list the arcs and then scan the perimeter in as shown in Fig. 1 (b), we can see that the entire perimeter is enclosed by $[A^2C^1]$ (node s_3), $[C^1C^2]$ (node s_4), $[B^1B^2]$ (node s_1) and $[A^1A^2]$ (node s_2).

Although, perimeter-test ensures that a sensor has sufficient number of coverage neighbors, this does not guarantee the full coverage of the sensing region. There may have some uncovered area in the middle of the region. Thus, we propose a *center-test* and a *distance-test*, so that a sensor ensures that neighbors are close enough to the center and provide full coverage.

Lemma 1: $\mathcal{S}_i \subseteq \bigcup_{s_j \in \mathcal{C}_i'} \mathcal{S}_i \cap \mathcal{S}_j$ if and only if any point P_i on the perimeter of s_i is covered by an arc $\widehat{P_{ij}^1 P_{ij}^2}$, where $s_j \in \mathcal{C}_i'$.

Proof: Since the perimeter circle that surrounds the sensing region $\mathcal{S}_i$ is also in $\mathcal{S}_i$, any arc $\widehat{P_{ij}^1 P_{ij}^2}$ on this perimeter circle must also be covered by one or more sensors in $\mathcal{C}_i'$. ■

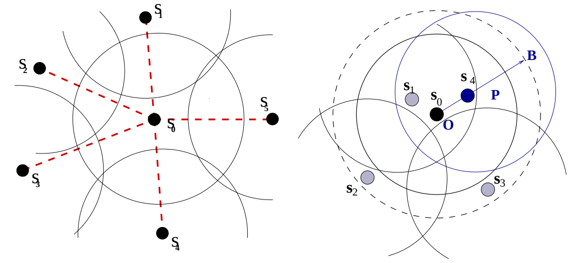
Center-test: When a sensor passes the perimeter-test, it has necessary number of coverage neighbors. However, this is not sufficient to claim that the sensing region is fully covered. For instance, there are two examples in Fig. 2 (a) and (b), where perimeter-test has been passed. In Fig. 2 (a), neighbors of s_0 are successfully covers the sensing region. However, in Fig. 2 (a), even there are 5 coverage neighbors, there is an uncovered area in the middle of the region. Therefore, the motivation of the center-test is to ensure that the center of a sensing region can be covered by at least one of a node's neighbors.

In this step, sensor s_i chooses one of its coverage neighbor s_j as *primary* neighbor which satisfies $d(i,j) \leq r$, where $d(i,j)$ is the distance between s_i and s_j. One intuitive necessary condition is that there should be at least one primary neighbor to cover the center point of a sensing region. If there is no primary neighbor as in Fig. 2 (a), i.e., distances between all coverage neighbors of the s_0 are greater than r, then neighbors are not sufficient to achieve full coverage. Therefore, center-test is a necessary condition in finding a coverage set.

In Fig. 2 (b), the *primary* neighbor of s_0 is s_4, where $d(0,4) = |OP| \leq r$. In case there are multiple sensors satisfying $d(i,j) \leq r$, we select the one having the minimum distance as the primary neighbor.

Corollary 1: For each s_i, $\mathcal{S}_i \supseteq \bigcup_{s_j \in \mathcal{C}_i'} \mathcal{S}_i \cap \mathcal{S}_j$ if and only if $1 \leqslant |\mathcal{C}_i' \cap \mathcal{C}_i^{1-hop}|$.

Proof: Consider a sensor s_i having coverage neighbors $\mathcal{C}_i$. Let all coverage neighbors are 2-hop neighbors, i.e., $\forall s_j \in \mathcal{C}_i, s_j \in \mathcal{C}_i^{2-hop}$. In this case, all coverage neighbors should be outside its sensing range, where $\forall s_j \in \mathcal{C}_i, vd(s_i, s_j) > r$. If all $s_j \in \mathcal{C}_i$ are more than r away from the sensor itself, there is a gap in the inner region where sensor is located that cannot be covered by any neighbors. By this contradiction, we show if there is no 1-hop coverage neighbor of s_i, $\mathcal{S}_i$ can not be covered. ■



(a) Distance test: fail. (b) Distance test: pass.

Fig. 2. Two examples where perimeter-test is passed for sensor s_0.

Distance-test: When a sensor passes the perimeter-test, and center-test, it is still possible that there are uncovered area of a sensor's coverage. The motivation of *distance-test* is to verify that coverage neighbors are close enough to the center and satisfy full coverage, which is based on the selection of primary neighbor in the center-test.

Let $s_p(i)$ be the primary neighbor of s_i. In this test, we check if for all $s_j \in \mathcal{C}_{Neigh}$, the following condition satisfies:

$$d(i,j) \leqslant r + d(i,p). \tag{1}$$

In Fig. 2 (b), to illustrate the distances between s_0 and neighbors clearly, we draw an *extended coverage* range of node s_0, where the center is s_0 and the radius is $R' = |OB| = r + |OP|$. We called the original sensing range of s_0 as R and the extended range as R' in the Fig. 2 (b). The condition in (1) can be verified by two extreme cases: for $d(i,p) = 0$, that is, node s_p and node s_i are in the same location, then the radius of the extended coverage, $R' = r$, which is exactly the same as s_i, that is, there is no extension of coverage from node s_p; for $d(i,p) = r$, that is, node s_p is on the perimeter of node s_i, then $R' = r + r = 2r$, which means that the extended coverage is enlarged one time. Therefore, the extended coverage shows the maximum distance that the primary neighbor can reach. If all sensors in the coverage neighborhood of s_i are closer than $r + d(i,p)$, then we say that full coverage is achieved and distance-test is passed.

Therefore, the first condition, *perimeter-test* is a necessary condition to cover the perimeter; and the second condition, *center-test* is also a necessary condition to cover the center. The third condition, *distance-test* is very effective for the full coverage after many tests, though it is an approximate condition for coverage calculation.

Therefore, sensors that have passed the *perimeter*, *center*, and *distance-tests*, are marked as *R-nodes* and will be eliminated to find an optimal coverage set which will be explained in the next section.

Lemma 2: Let s_i, $\bigcup_{s_j \in \mathcal{C}_i^*} \mathcal{S}_i \cap \mathcal{S}_j \subseteq \mathcal{S}_i$, where $\mathcal{C}_i^*$ is the subset of coverage neighbors that is sufficient for perimeter-test, then if $\forall s_j \in \mathcal{C}_i^*, d(i,j) < r + d(i,p)$[1].

[1]We omit inclusion of the proof of this lemma due to space restrictions.

Using the proposed method, a sensor s_i can be covered by at least three coverage neighbors. If only 1-hop coverage neighbors are considered then all three neighbors are needed to be closer than r similar to the coverage calculation methods in [4], [17]. On the other hand, if we also consider 2-hop coverage neighbors, then only a single 1-hop coverage neighbor together with two 2-hop coverage neighbors can be sufficient to cover $\mathcal{S}_i$. Note that two sensors can not be at the same location.

Lemma 3: If $\bigcup_{s_j \in \mathcal{C}_i} \mathcal{S}_i \cap \mathcal{S}_j = \mathcal{S}_i$, then there must be a subset of $\mathcal{C}_i$, denoted by $\mathcal{C}_i^{'}$, such that $\bigcup_{s_j \in \mathcal{C}_i^{'}} \mathcal{S}_i \cap \mathcal{S}_j = \mathcal{S}_i$. So, we can write $3 \leqslant |\mathcal{C}_i^{'}|$, where $\mathcal{C}_i^{'}$ satisfies either

- $1 \leqslant |\mathcal{C}_i^{'} \cap \mathcal{C}_i^{1-Neigh}|$ and $2 \leqslant |\mathcal{C}_i^{'} \cap \mathcal{C}_i^{2-Neigh}|$, or
- $\mathcal{C}_i^{'} \subseteq \mathcal{C}_i^{1-Neigh}$.

Proof: From Corollary 1, we know that at least one sensor in $\mathcal{C}_i^{1-Neigh}$ is needed for full coverage. The angle θ of the arc that is not covered by any sensor s_j, $s_j \in \mathcal{C}_i^{1-Neigh}$, satisfies $\theta > \pi$. Since all arcs that consitute the 2π perimeter must be covered for full coverage (see Lemma 1), the uncovered arc can be covered by sensors in $\mathcal{C}_i^{2-Neigh}$. However, the angle β of the arc covered by any sensor s_k, $s_k \in \mathcal{C}_i^{2-Neigh}$, satisfies $\beta < \frac{2\pi}{3}$. Hence for full coverage, at least two sensors in $\mathcal{C}_i^{2-Neigh}$ are needed.

Now let us consider the second case where only 1-hop coverage neighbors are used. The angle α of the arc covered by any sensor s_m, $s_m \in \mathcal{C}_i^{1-Neigh}$, satisfies $\alpha < \pi$. Similar to the first case, each sensor in $\mathcal{C}_i^{1-Neigh}$ can only cover less than π, therefore at least three sensors are needed in $\mathcal{C}_i^{1-Neigh}$ for full coverage. ■

IV. Constructing Coverage and Dominating Coverage Sets

In this section, we will explain how coverage and connected dominating sets are constructed in a distributed fashion. First step is to discover the possible redundant nodes, which is explained in the previous section. However, each redundant node can not be removed from the coverage set unless its necessary coverage neighbors are in the coverage set. Therefore, we use an energy-aware redundant elimination method, where sensors calculate their benefits of being in the coverage set in terms of energy and then start sending announcement messages to built the coverage set. After coverage set is established, dominating coverage is dynamically constructed starting from the sink. Here, we give the details of the steps that each sensor follows.

A. Constructing Coverage Set

To construct the coverage set, we first need to determine essential nodes to preserve coverage and eliminate the redundant ones. If sensor s_i calculates that its coverage neighbors, $\mathcal{C}_i$, are not sufficient to monitor its sensing region, then it is *mandatory* member of the coverage set, i.e., if $\mathcal{S}_i \nsubseteq \bigcup_{s_j \in \mathcal{C}_i} \mathcal{S}_i \cap \mathcal{S}_j$, then $s_i \in \mathbf{C}$.

In this case, a sensor broadcasts an *I-AM-ESSENTIAL* message and become a member of coverage set. Otherwise, it is a candidate of being redundant node and follows the *benefit calculation* step that determines its benefit to be in the coverage set. Any sensor that is not a mandatory E-node has to calculate its benefit. Our goal is to choose the coverage set of sensors to maximize the *benefit* in terms of coverage and the residual energy, i.e., the largest uncovered sensing region is covered with the least sensors having maximum residual energy.

Consider the sensor s_i with sensing region $\mathcal{S}_i$. If $\mathcal{S}_i$ is fully covered by the current E-nodes, i.e., $\mathcal{S}_i \subseteq \bigcup_{s_j \in \mathcal{C}_i^E} \mathcal{S}_i \cap \mathcal{S}_j$, then sensor s_i sends an *I-AM-REDUNDANT* message and go to sleep mode. Otherwise, it calculates and broadcasts its *benefit*, which is:

$$benefit(s_i, t) = \frac{e_i(t)}{S_i \cap \mathcal{S}_{\mathcal{C}_i^E}}, \tag{2}$$

where $\mathcal{S}_{\mathcal{C}_i^E}$ is the total region covered by the mandatory E-node neighbors of sensor s_i, and $e_i(t) \in [0, 1]$ is the residual energy level. Nodes having higher residual energy, and smaller covered area will have a better chance of being in coverage set whereas others will be eliminated as redundant nodes.

In this step, mandatory nodes have already announced and each node is aware of its benefit and its neighbors benefit. Then, nodes start to broadcast *I-AM-ESSENTIAL* messages after a short back-off time, denoted by $T_{back-off}$, where $0 < T_{back-off} \leq CB_{MAX}$. CB_{MAX} denotes the maximum back-off and is determined based on the average one-hop latency during neighbor discovery. We assume that sensors have globally synchronized clocks [12]. A sensor determines its back-off time based on its benefits, i.e., a node having the maximum benefit will have the shortest back-off time, thus announcing the *I-AM-ESSENTIAL* message earlier.

When a sensor receives *I-AM-ESSENTIAL* message from its coverage neighbor, it should update its benefit since its covered region, $S_i \cap \mathcal{S}_{\mathcal{C}_i^E}$, might increase by newly announced E-node neighbors. If $S_i \cap \mathcal{S}_{\mathcal{C}_i^E} \supseteq S_i$, sensor reset its back-off timer, sends an *I-AM-REDUNDANT* message and go to sleep. Otherwise, the benefit will decrease proportional to the covered area which may also prolong the back-off time. Coverage tier is established at the end of CB_{MAX}. Next, we will discuss how the dominating coverage is established to preserve connectivity using minimum number of nodes.

In sharp contrast to earlier studies [8], [17], we decompose the coverage and connectivity features of the WSN in this work. After establishing the optimal coverage set, all E-nodes are not necessarily be active all the time. Only a small number of E-nodes can work as a backbone to forward the data traffic and delivery tasks sent by the sink. To achieve this, we establish a dominating coverage set among coverage set where an E-node is either a dominating node or a direct (one-hop) neighbor of a dominating node. The dominating nodes always stay active to preserve the connectivity of the network and forward the data traffic to/from the sink. E-nodes can communicate at least with one ED-node and send/receive their measurement/query via their neighboring ED-nodes.

B. Constructing Connected Dominating Set

Following, we need to construct a dominating set among the nodes in the coverage set. We use a greedy approach similar to the centralized algorithm, i.e., sensors are removed one by one as long as the remaining set is connected. However, when such a greedy approach is run by sensors, a distributed algorithm, e.g., distributed breadth-first search, is necessary to ensure that the remaining network is connected in each iteration. In a large-scale sensor network, distributed breadth-first search may incur high overhead due to its computational complexity, i.e., $O(Dlog^3 N)$, where D is the diameter of the network [3]. Therefore, we propose a dynamic dominating set construction approach triggered by the sink.

During dominating set construction, sensors broadcast three types of messages.

- *JOIN-Backbone*: It indicates that there is no dominator in the neighborhood, thus sensor may become a dominating node.
- *CANDIDATE-Backbone*: A sensor broadcasts this message after receiving *JOIN-Backbone*.
- *NOT-IN-Backbone*: This message is sent by sensors which are already connected to a dominating node and they decide not to be dominating nodes.

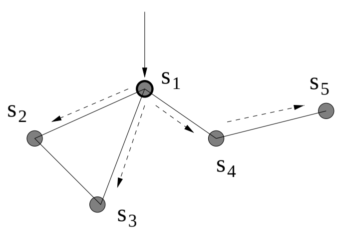


Fig. 3. Given sensors $s_1, s_2, s_3, s_4, s_5 \in \mathbf{C}$, where s_1 receives *Join-BACKBONE* message and broadcast to its neighbors, thus become an ED-node.

Our dominating set construction starts from the sink by sending a broadcast *JOIN-Backbone* message. The idea is that sensors which will forward the message, are included to the connectivity set as a dominator. In the first step, *JOIN-Backbone* message is received by the neighbors of the sink, which are called *candidate dominator*. A candidate dominator, again sets a back-off time $\in [0, DB_{MAX}]$ to forward the message, where DB_{MAX} denotes the maximum back-off while establishing connectivity-tier and is determined based average one-hop latency, and the node density. Back-off time will be inversely proportional to residual energy level and the degree of connectivity. In this context, degree of connectivity is the number of neighbors which have not sent a *JOIN-Backbone* or *NOT-IN-Backbone* message. For example, sensor s_i has 4 neighbors among which two of them have sent a *NOT-IN-Backbone* message, whereas the other neighbor has sent a *JOIN-Backbone* message. This implies that two neighbors are connected to ED-nodes and one neighbor has already become an ED-node. In this case, degree of connectivity of s_i is 1 in calculating its benefit. Similar to the previous step, a node having greater benefit has shorter back-off time, thus forwarding *JOIN-Backbone* message earlier to be an ED-node.

When a node receives a *JOIN-Backbone* message, it (i) updates its its connectivity is decreased; because one of its neighbors becomes a dominator; (ii) sets/updates its

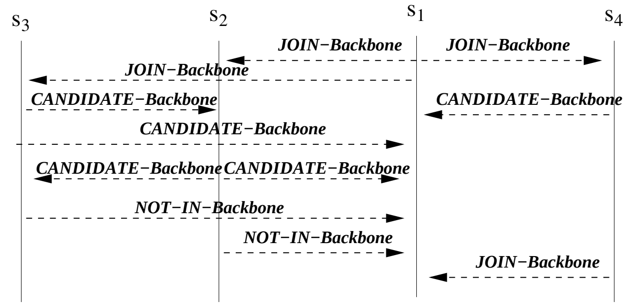


Fig. 4. Signaling diagram showing the messages while constructing dominating coverage set for the example sensors given in Fig. 3.

back-off time based on newly calculated benefit; and (iii) broadcasts a *CANDIDATE-Backbone* message. By receiving *CANDIDATE-Backbone* messages during the back-off time, candidate dominators can be noticed if their neighbors are also candidates. If all neighbors of a candidate node is either dominator or candidate dominators, it can safely give up of being dominator, since all its neighbors are already received a *CANDIDATE-Backbone* message. In this case, a sensor node sends a *NOT-IN-Backbone* message indicating that it will not be an element of dominating coverage.

At the end of a back-off period, a candidate which has not been self-removed, forwards *JOIN-Backbone* message and becomes an ED-node. Following the forwarded *JOIN-Backbone* message, new candidates appear and send *CANDIDATE-Backbone* messages. This process continues until all nodes have received at least one *JOIN-Backbone* message. Note that, a sensor updates its benefit after receiving *JOIN-Backbone* or *NOT-IN-Backbone* messages.

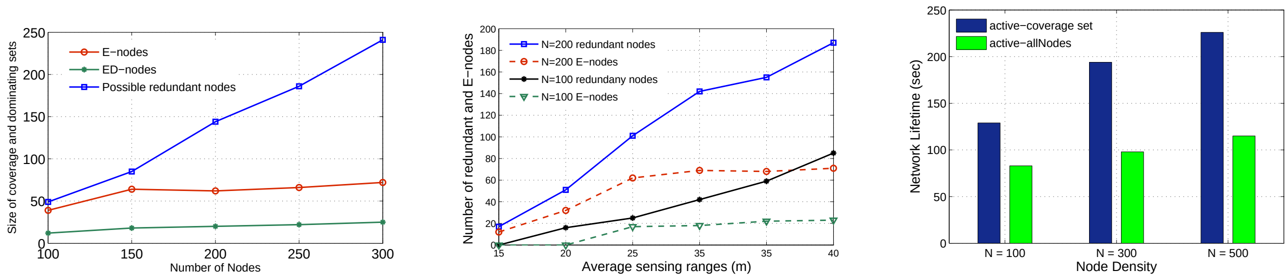
The energy consumption of ED-nodes may be higher than the E-nodes; and R-nodes may have the lowest energy consumption due to continuous sleep. Thus, to acquire a fair energy consumption among sensors, coverage and connectivity sets should be updated throughout the lifetime of the network.

In this paper, we make use of *global update* where all E-nodes and ED-nodes are re-selected independent from the current set. In particular, global update is the process of repeating algorithms with latest residual energy levels of sensors. By this way, sensors that have overlapping regions and were E-Nodes in the previous round might be R-nodes in the next update because more energy has been consumed when they were E-node before.

V. Performance Evaluation

The performance of our approach is evaluated using simulations that are performed in an 250 m x 250 m area consisting of different numbers of sensors distributed randomly. In the basic scenario, 100 fixed sensor nodes having transmission range of 100 m and sensing ranges vary from 15 m to 40 m are used. We use the radio power consumption parameters in [10]. The energy consumption of turning the radio on/off is negligible. The buffer size of sensor nodes is chosen as 50 and the packet length is 100 bytes.

In the first experiment we investigate the performance of our coverage calculation under three metrics: the total number



(a) (b) (c)

Fig. 5. Performance of the redundant discovery and elimination algorithms.

of locally detected redundant sensors, the total number of nodes for coverage (E-nodes), and the total number of nodes to preserve connectivity (ED-nodes). To cover a 250 m x 250 m area with sensing range of 30 m on average, we use random placement of 100 to 300 nodes. From Fig. 5 (a), we can observe that among these five scenarios, the network having 300 nodes has the highest number of locally detected redundant nodes, whereas the number of E-nodes and E-nodes do not significantly change with node density. As shown in Fig. 5 (a), even in the low density, E-node ratio is above 50%, which shows that only 50% of the nodes are necessarily be active when the proposed method is used. Moreover, we evaluate the number of redundant and E-nodes of higher sensing ranges. When the ratio of sensing range is increased, the number of redundant nodes is increased up to 70% where the number of E-nodes and ED-nodes remains the same. Figure 5 (b) shows the number of nodes in a low dense network (N=100) and higher density with N=200. Finally, we evaluate the possible energy saving when the coverage set is used, where R-nodes are scheduled to sleep to save energy and only E-nodes and ED-nodes are active. In Fig. 5 (c), we demonstrate the network lifetime compared to the scheme when all nodes are active. We consider a WSN as alive when the sensing field is fully covered. In other words, a network is alive when every point in A is covered by at least one sensor. According to this, we observe that network lifetime is prolonged significantly when R-nodes are scheduled to sleep while preserving coverage and connectivity, especially in high density networks. Even in low density network with 100 nodes, network lifetime is prolonged up to 28% which shows the effective energy savings of using coverage and dominating coverage sets.

VI. Conclusions

In this paper, we focus on the problem of finding an optimal coverage and dominating coverage sets that can be used for energy-efficient communication in large scale sensor networks. We present a distributed algorithm that effectively eliminates redundant nodes with guaranteed connectivity for heterogeneous sensors having different sensing ranges. Further, an extension scheme is presented that finds the minimum number of dominating sensors among the coverage set which can also be used in effective node scheduling and topology control.

References

[1] Z. Abrams, A. Goel, and S. Plotkin. Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks. In *Proc. of Int. Conference on Information Processing in Sensor Networks*, Berkeley, California, USA, April 2004.

[2] D. Arganbright. *Practical Handbook of Spreadsheet Curves and Geometric Constructions/Book and Disk*. CRC Press, 1993.

[3] S. Butenko, X. Cheng, C. Oliveira, and P. Pardalos. A New Heuristic for the Minimum Connected Dominating Set Problem on Adhoc Wireless Networks. *Cooperative Control and Optimization, Kluwer Academic Publisher*, pages 61–73, 2004.

[4] B. Carbunar, A. Grama, J. Vitek, and O. Carbunar. Coverage Preserving Redundancy Elimination in Sensor Networks. In *Proc. of IEEE SECON*, 2004.

[5] M. Cardei, M. Thai, and W. Wu. Energy-Efficient Target Coverage in Wireless Sensor Networks. In *Proc. of IEEE Infocom*, March 2005.

[6] X. Cheng, A. Thaeler, G. Xue, and D. Chen. TPS: A Time-Based Positioning Scheme for Outdoor Wireless Sensor Networks. In *Proc. of IEEE Infocom*, March 2004.

[7] S. S. Dhillon and K. Chakrabarty. Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks. In *Proc. of IEEE WCNC*, pages 1609 – 1614, March 2003.

[8] H. Gupta, S. R. Das, and Q. Gu. Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution. In *Proc. of ACM Mobihoc*, Annapolis, Maryland, USA, June 2003.

[9] T. He, C. Huang, B. M. Blum, and J. A. Abdelzaher. Range-free Localization Schemes for Large Scale Sensor Networks. In *Proc. of ACM Mobicom*, pages 81–95, San Diego, California, USA, September 2003.

[10] J. Hill and D. Culler. Mica: A Wireless Platform for Deeply Embedded Networks. *IEEE Micro 22*, 22(6):12 – 24, 2002.

[11] M.-F. Huang and Y.-C. Tseng. The Coverage Problem in a Wireless Sensor Network. In *Proc. of ACM WSNA*, San Diego, CA, USA, September 2003.

[12] Q. Li and D. Rus. Global Clock Synchronization in Sensor Networks. In *Proc. of IEEE Infocom*, 2004.

[13] H. Lim, C. Lim, and J. C. Hou. A Coordinate-Based Approach for Exploiting Temporal-Spatial Diversity in Wireless Mesh Networks. In *Proc. of ACM Mobicom*, September 2006.

[14] C. Liu, K. Wu, Y. Xiao, and B. Sun. Random Coverage with Guaranteed Connectivity: Joint Scheduling for Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, in press, 2006.

[15] S. Slijepcevic and M. Potkonjak. Power Efficient Organization of Wireless Sensor Networks. In *Proc. of IEEE International Conference on Communications*, pages 472 – 476, Helsinki, June 2001.

[16] D. Tian and N. D. Georganas. A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks. In *Proc. of First ACM Int. Workshop on Wireless Sensor Networks and Applications*, Georgia, USA, 2002.

[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. of SenSys*, pages 28–40, 2003.

[18] H. Zhang and J. Hou. Maintaining Sensor Coverage and Connectivity in Large Sensor Networks. In *Proc. of NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Adhoc Wireless, and Peer-to-Peer Networks*, 2004.