Effective Sensor Relocation Technique in Mobile Sensor Networks

Ahmed M. Khedr [1] and Hager Ramadan[2]

[1] Dept. of Computer Science, Faculty of Sciences, Sharjah University, Sharjah, UAE.
akhedr@sharjah.ac.ae

[2] Mathematical Department, Faculty of Science, Zagazig university, Zagazig, Egypt.
hramadan@ zu.edu.eg

ABSTRACT

In a wireless senor network, one of the main design challenges is to achieve long network lifetime by turning off some redundant sensor nodes while maintaining sufficient coverage for sensing and connectivity. In this paper, two improved algorithms are proposed, the first one is to find if a sensor is completely covered by its neighbors so as to decide whether it is redundant and can be used to replace failed nodes. The second algorithm is to relocate the failed node by a proper one. Simulation results show that the redundant discovery algorithm can computes all redundant nodes, and relocation algorithm can significantly reduce the energy consumption.

KEYWORDS

Redundant node, Relocation, Sector, Wireless Sensor Networks.

1. INTRODUCTION

A wireless sensor network (WSN) is designed to collect and process data, and transmit sensed information and detect events or phenomena for interested users. Distributed WSNs are important for a number of strategic coordinated target applications such as agriculture and environmental monitoring, structural engineering, military applications, health monitoring and surgery. Sensor deployment has received considerable attention recently. Many of these works assume that the environment under control is adequately known, and sensors can be deployed manually.
Extending network lifetime is challenging because these networks are built from short-lived sensor nodes. In such networks, energy source is usually a battery power technology, which has yet to reach a stage where sensors can operate for a long time without recharging. Moreover, sensors are often intended to be deployed in a remote or hostile environment, such as a battlefield or desert, it is undesirable or impossible to recharge or replace the battery power of all the sensors. However, a long system lifetime is expected for many monitoring applications [1], [2], [3], [4], [5], [6]. The system lifetime is measured by the time until all nodes have been drained out of their battery power or the network no longer can provide an acceptable event detection ratio, thereby directly affecting network usefulness.
Therefore, energy efficient design for extending system lifetime without sacrificing system reliability is one important challenge to the design of large wireless sensor networks [7], [8]. In a WSN, all nodes share common sensing tasks. This implies that not all sensors are required to perform the sensing task during the whole system lifetime. Turning off some nodes does not affect the overall system function as long as there are enough working nodes to assure the coverage and connectivity to the base station.
In case of an unknown or a hostile environment, such as remote harsh fields [9], disaster areas and toxic urban regions [10], sensor deployment cannot be performed manually. To scatter sensors by low flying or UAV aircraft is one possible solution. However, using this technique, the actual landing position of sensors cannot be controlled due to the presence of wind and

obstacles such as trees, valleys and tall buildings. Consequently, the coverage may be marginal as compared to the application requirements and could be true no matter how many sensors are dropped. In these scenarios, it is necessary to make use of mobile sensors, which can move to the right places so as to ensure the required coverage [11]. In this paper, the problem of sensor relocation is presented, i.e., moving previously deployed sensors to overcome any void in the coverage, or the failure of some nodes or to respond to an occurring event that requires precise measurement by moving a sensor to a desired location. When a hole occurs, the obvious solution seems to relocate the closest redundant sensor node to take care of the coverage hole. How to find the closest excess node may seem trivial at a first glance, but unfortunately it is not so easy. It has been pointed out that there are indeed important issues to be taken into account such as minimizing total energy consumption, optimizing time required by the overall movements via cascaded relocations, and minimizing average moving distances in cascaded relocations of several sensor nodes [12]. Such sensor relocation is different from existing work on mobile sensors which concentrate on sensor deployment, i.e., moving sensors to provide certain initial coverage. Therefore, as Compared to initial deployment, sensor relocation requires many special considerations. First, sensor relocation may have a strict response time requirement. For example, if the sensor monitoring a security-sensitive area dies, another sensor should move in to replace its functionality as soon as possible. Second, relocation should not affect the current application using the WSN, which means that the relocation should minimize its effect on the existing network topology. Finally, since movement may be much more expensive in terms of energy than computation and communication. Any algorithm to be used, must balance energy costs with response time. In particular, care must be taken to balance the energy costs of a moving individual node with the overall network energy cost so as to ensure maximum network lifetime.

Contribution: In this paper, we propose a redundant discovery algorithm that finds out all the redundant nodes in the network by dividing the sensing region into six equal sectors and then test the coverage of each of them, and we also propose a framework for relocating mobile sensors in a timely, efficient, and balanced manner, while maintaining the original network topology as much as possible. We divide the network into clusters to take the advantage of local information. The clusters heads (CHs) have the information about the redundant nodes in their clusters and recover any holes occurring due to the node failure by searching for the nearest redundant node closer to the failed node.

The rest of the paper is organized as follows: Section 2 covers the related work of our problem. In section 3, we describe the redundant nodes discovery algorithm. Section 4 presents the technique to relocate the failed nodes. Section 5 performs the complexity analysis of the proposed algorithms. Section 6 does the performances evaluation of the proposed algorithms. We conclude our work in Section 7.

2. RELATED WORK

Sensing coverage and network connectivity are two of the most fundamental problems in WSNs [7], [8]. Finding an optimal node deployment strategy that would minimize cost, reduce computation and communication overhead, be resilient to node failures, and provide a high degree of coverage with network connectivity is extremely challenging.

There have been some researches to improve the coverage in mobile sensor network. Wang et. al. [7] suggested non-uniform deployment of sensors to enhance lifetime of a WSN. In [13], Wang et. al. use Voronoi diagrams to discover the coverage holes and they designed three movement-assisted sensor deployment protocols, VEC (VECtorbased), VOR (VORonoi-based), and Minimax based on the principle of moving sensors from densely deployed areas to sparsely deployed areas, where VECtorbased Pushes sensors away from a densely covered area, VORonoi-based pulls sensors to the sparsely covered area, and Minimax based Moves sensors to their local center area. With randomly deployed sensors, these algorithms provide high coverage within a short time and limited moving distance. If the initial distribution of the sensors is extremely uneven, disconnection may occur, thus, the Voronoi polygon constructed may not be accurate enough, which results in more moves and larger moving distance. Zou and Chakrabarty [14], proposed a centralized virtual force based mobile sensor deployment algorithm (VFA), to enhance the coverage after the initial random placement of sensors. Each sensor behaves as a source of force for all other sensors. This force can be either positive (attractive) or negative (repulsive), and assumed that obstacles to exert repulsive forces on a sensor. Likewise, areas of preferential coverage exert attractive forces on a sensor. In VFA, there is a powerful cluster head, which communicates with all the other sensors, collects sensor position information, and calculates forces and desired position for each sensor. The movement-assisted sensor deployment deals with moving sensors to meet coverage and load balance requirements. In [15], Wu et. al. focus on minimizing total moving distance and propose an optimal, but centralized solution, based on the Hungarian method. This solution is illustrated in an application where the monitoring area is a 2-D grid-based mesh. They also proposed a scan-based solution that does not resort to global (load) information. This solution can achieve load balance and minimize total moving distance of sensors in 1-D arrays. All the previous methods can not maintain the topology of the network when node failure takes place. All of these works improve the coverage by distributing the nodes over the environment to make sure that they are densely covered.

In [16], Wang et. al. proposed a solution to the placement problem by allowing an arbitrary-shaped sensing field possibly with arbitrary-shaped obstacles and an arbitrary relationship between the communication distance rc and sensing distance rs of sensors. This scheme works in two steps. First, it partitions a sensing field A into a number of regions. Regions are classified into single-row regions and multi-row regions. A single-row region is a belt-like area with width no larger than $\sqrt{r_{\min}}$, where $r_{\min} = \min\{ r_c; r_s \}$, so a row of sensors is sufficient to fully cover the region while maintaining connectivity. A multi-row region is perceivably larger and can be covered by several rows of sensors. The solutions to the dispatch problem include a centralized one and a distributed one was also proposed. The centralized one is based on adopting the former placement results and converting the problem to the maximum-weight maximum-matching problem. With a greedy strategy, the second scheme is distributed in that sensors will select the most suitable locations as their destinations and compete with each other to move to these locations. This work requires a global knowledge of the environment to partition the field into a number of regions which in real cannot happen.

In [17], Wang et. al. proposed a bidding protocol to heal the coverage holes, while the protocol requires many round of iterative process. Each round is composed of three phases: Service advertisement, Bidding, and serving phase. Mobile sensors broadcast their base prices and locations in the service advertisement phase. Static sensors detect coverage holes and send bidding messages to the closest mobile sensors in the bidding phase. In the serving phase, mobile sensors choose the highest bid, move to heal the hole and update the base price. The consumed energy is high because of the zigzag movements of the sensors, and technique does not provide either exchange of holes or any load balance. To solve these problems, they proposed proxy-based sensor deployment protocol [18], where the target location is determined based on a distributed iterative algorithm, move logically, and exchange new logical locations with their new logical neighbours. In [12], Wang et. al. proposed two phases algorithm. The first one is to find redundant sensor location, and the second one is to relocate redundant sensor. A grid-quorum solution was proposed to quickly locate the closest redundant sensors to the target area, where a sensor failure occurs. This solution uses the concept of quorum to locate sensors with low message complexity. Then a cascaded movement scheme was developed to relocate the located redundant sensors in a timely, efficient, and balanced way. This algorithm requires a global knowledge about the environment. In [19], Li and Santoro proposed a distributed zone-based sensor relocation protocol (ZONER) for mobile sensors on the basis of a restricted flooding technique (ZFlooding). It requires zero-knowledge about the sensor field. The ZONER is able to effectively discover previously-deployed redundant sensors and relocates them by shifting to replace failed non redundant ones without changing the network topology. In zone-based Sensor Relocation Protocol, redundant nodes register themselves with all the non redundant nodes within its predefined vertical registration zone. The neighbors of a failed node inquire all the non redundant nodes inside their horizontal request zones for redundant node. The request zones intersect with a number of registration zones. The non redundant node in the intersection area is able to provide redundant node information. In case no available redundant node can be found, the size of the request zones is increased. When the size is larger than the area of the sensor field, all the non-redundant nodes will be included in the network. This leads to increasing the number of messages and hence incurs higher energy consumption. In this paper, we propose a cluster based relocation algorithm that combines the advantages of papers [12], [19]. It doesn't require either a global knowledge about the environment or needs any flooding technique. Only a local knowledge about the environment is desired and the relocation process is executed via CHs, and not via the sensor nodes which can reduce the number of relocation messages and so the energy consumed. The evaluation the redundant discovery and relocation algorithms and the comparison between the cluster based relocation algorithm and the zone based relocation algorithm are represented.

3. REDUNDANT NODES DISCOVERY

One of the main challenges in distributed coverage calculation is to determine whether the sensor is redundant using its 1-hop and 2-hops coverage neighbors. In this context, we use the term of coverage to calculate whether a sensing region of a sensor s_i is fully covered. To the best of our knowledge, existing coverage method [20], [21] only considered 1-hop coverage neighbors, i.e., neighbors inside the sensing range. In this paper, we propose a decentralized approach that only depends on local states of sensing neighbors, we take into account the 2-hops neighbors that share a common area as well decreases the lower bound for the number of coverage neighbors while effectively utilizing redundant nodes.
Intuitively, the relationship between connectivity and coverage depends on the ratio of the communication range to the sensing range. However, it is easy to see that a connected network may not guarantee its coverage regardless of the ranges. This is because coverage is concerned with whether any location is uncovered, while connectivity only requires that the locations of all active nodes are connected. The communication graph of a set of nodes that at least 1-cover of a convex region 'A' is connected if $R_c \geq 2\ R_s$ [27]. The algorithm only requires information about the locations of all sensing neighbors. It maintains a table of known sensing neighbors based on the beacons (Hello messages) that it receives from its communication neighbors. When $R_c \geq 2\ R_s$, the Hello message from each node only needs to include its own location. When $R_c < 2\ R_s$, however, a node may not be aware of all sensing neighbors through such Hello messages. Since some sensing neighbors may be hidden from a node, it might activate itself to cover a perceived sensing void that is actually covered by its hidden sensing neighbors. Thus the number of active nodes would be higher than necessary in this case.

3.1. Algorithm Assumptions

Our approach relies on the following key assumptions regarding the WSN and sensor nodes:

1. All the sensor nodes have the same characteristics (the same communication and sensing ranges).

2. The communication range (R_c) is greater than or equal the twice of the sensing range (R_s) to enable the node to communicate with nodes outside the sensing range which is necessary to determine overlapped areas [22].

3. Localization: The position of every sensor node is known in any arbitrary global coordinate system possibly by using a localization system from [23], [24], [25], [26]. For simplicity, we assume that every node knows its location in space in terms of (x; y) coordinates. The neighbors of a particular node are determined based on its radio range.

4. Nodes are organized as clusters and the cluster heads are responsible for executing the relocation algorithm. All cluster members are able to communicate with their cluster head. This communication is necessary only for the failure report and the transmission of the new locations to the nodes. This organization of nodes and the selection of CHs can be performed by any clustering techniques according to the environment, application, and the number of nodes [28].

Definition: The sensor node s_i is said to be redundant if its sensing range is subset of the sensing range of its neighbors, i.e., The sensor node s_i is said to be redundant if its sectors ($sect_{j\in\{1;2;3;4;5;6\}}(s_i)$) are covered by the sensing range of the neighbors of s_i.
Lemma 1: The sector is covered, if there exits one sensor inside it.
Proof: It is enough to prove that the sector is covered if there is a sensor that lies on one of the sector's edges and on the boundary of the sensing circle of the sensor.
For any two sensor nodes s_1, s_2, we can draw a triangle with s_1, s_2, and one of the intersected points of the two sensing circles as shown in Figure 1. It is obvious that this triangle is equal sides' triangle. So the maximum coverage sector is the sector with angle 60^o.

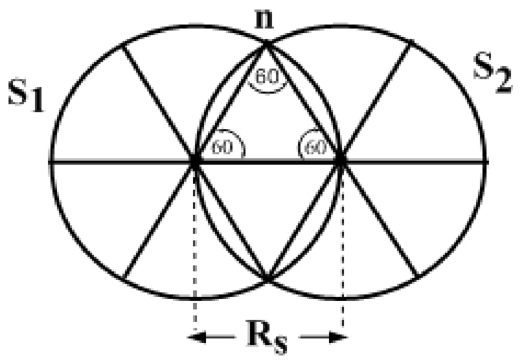

Figure 1: Sectors of angle 60

In some cases, the sector is covered although there is no sensor inside it, such as $sect_1(s_1)$ in Figure 2. For the sector that does not have sensors inside, we check the coverage of the boundary points of the sector if it is covered (i.e., the distance between each boundary point of a sub sector and one of the neighbors of the node is less than the sensing range (R_s)). If the boundary points of a sector/sub-sector are covered by the same node, then the sector/sub-sector is covered by that node. Otherwise, we divide the sector into two equal sub-sectors and repeat the same steps to each sub-sector until we reach sub-sectors with a threshold angle.

3.2. Algorithm Outline

Each node executes the following steps to determine whether it is necessary to become active or not.

Step 1: Send message to your neighbors, including location information along with ID.

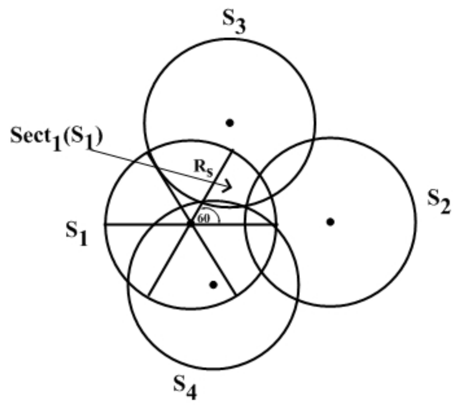

Figure 2: Example of covered sector without any node inside.

Step 2: Divide the sensing range into equal six sectors of angle 60^o.

Step 3: for each sector, (the sector is covered if it has a sensor node inside it). Otherwise execute the following steps:

1. If the boundary points of the sector belong to the same sensor, the sector is covered as shown in Figure 2 (the sector is not covered if there is at least uncovered boundary point and so the node is not redundant).
2. Otherwise, repeat the following steps till the angle of the sub-sector reach a threshold.
   i. Divide the sector into two equal sub-sectors.
   ii. Test the coverage of the boundary points of the sub-sectors.
   iii. If all the boundary points are covered, go to (i) to divide each sub-sector into two equal sub-sectors. Otherwise, the node is not redundant.

Each node s_i executes the following algorithm to identify whether it is a redundant node or not.

Algorithm 1 Redundant Discovery Algorithm

```
Output: True if it is a redundant node; Otherwise false.
1:  Divide your sensing range into six equal Sectors $sect_j(s_i)$, j= 1,....,6.
2:  For j =1 to 6 do
3:     If $sect_j$ boundary points are not covered by any of $s_i$ neighbors then
4:         Return false
5:    Else if there exists a sensor node inside $sect_j$ or the boundary points of $sect_j$ belong to
    one sensor.
6:         Continue
7:      Else
8:        While Q < Threshold repeat
9:            Divide $sect_j$ into two equal sub-sectors with Angle Q.
10:        For each sub-sector do
11:           Test the coverage of the boundary points of each Sub-sector.
12:            If one of the boundary points is not covered by any of the neighbor of $s_i$.
13:               Return false
14:            Else
15:            Go to step 8
16:           End if
```

```
17:         End for
18:     End if
19:           Return true
20: End for
```

4. THE CLUSTER BASED RELOCATION ALGORITHM

In this section, the relocation algorithm based on clusters is presented. In this algorithm, each cluster has a leader called the CH, which can communicate with other neighbor CHs through nodes called Gateways (GWs). The remaining nodes called Cluster Members (CM). The list of redundant nodes inside each cluster is known to the CH. Our algorithm is executed by the CH that has failed node.

4.1. Algorithm Outline

The algorithm is composed of two phases, the first phase is to find the nearest redundant node in the cluster of the failed node or in the neighboring clusters. In the second phase, the relocation of the nearest redundant node to the failed node is executed.

If the number of redundant nodes is not equal to zero ($R \neq \emptyset$).

Step 1: Find the nearest redundant node from the failed node by calculating the distance between the failed node and each redundant node in the cluster.

Step 2: Send recover message to the neighbors CHs to find the nearest redundant node from the failed node. The message includes the coordinates of the failed node, the minimum distance at the failed node cluster, and the value (1) means that the failed node's cluster contains redundant node and the cluster head requests from its neighbors (CHs) to search only inside themselves.

Step 3: The received CH will find the nearest redundant node to the failed node by computing the distances from each redundant node and the failed node, if the distance of the nearest redundant node is less than the received one, the CH will reply back to the sender with minimum distance.

Step 4: Invoke the redundant node with smallest distance to relocate the failed node.
Otherwise i.e., $R = \emptyset$.

Step 1: Send recover message to each neighbor of the CH to find the nearest redundant node from the failed node. The message includes the coordinates of the failed node, present time, the threshold time and the value (0). (the present time and the threshold time are sent to prevent the CHs to send more massages when the threshold time is spent, and the value (0) means that the failed node's cluster does not contain any redundant nodes and it requests from its neighbors to search inside themselves and among their neighbors).

Step 2: At the receiving CH, start the time counters to determine the elapsed time, and find the nearest redundant node to the failed node.

Step 3: Invoke to relocate the node with smallest distance to take care of the failed node.

Each cluster head that has failed report, will execute the following code:

Algorithm 2: Failed Node Relocation

```
Input: $(x_i, y_i)$ is the coordinates of the failed node.
          R: is the set of redundant nodes in the cluster.
          CN: is the set of neighbor cluster heads.
1: If  $R \neq Ø$ then
2:    For each r in R do
3:       Calculate the distance to the failed node
4:    End for
5:    Find the redundant node with minimum distance d.
6:    For each CH in CN do
7:        Send recover message with $(x_i, y_i)$, and the minimum distance at the the failed node
            cluster $(x_i, y_i, d, 1)$.
8:    End for
9:    Up on receiving respond messages from each cluster's neighbors, find the nearest
          redundant node and its cluster ID.
10:   Invoke the redundant node with the smallest distance to relocate the failed node
11: Else
12:    For each CH in CN do
13:      send recover message with $(x_i, y_i)$, the present time, and the threshold
          $((x_i, y_i), T, T_{th}, 0)$.
14:   End for
15:   Start counting for time (T).
16:   Up on receiving respond message from the cluster's neighbors, find the smallest
      distance to the failed node and its cluster ID.
17:  If $T > T_{th}$ then
18:   Invoke the redundant node with smallest distance to relocate the failed node.
19:   End if
20: End if
```

For clarity, in Figure 3, CH_1 executes the relocation algorithm to find the nearest redundant node. It first finds a in its members as the nearest redundant node, but after receiving messages from its neighbour CHs, *b* will be the nearest redundant node to be moved to the new location.

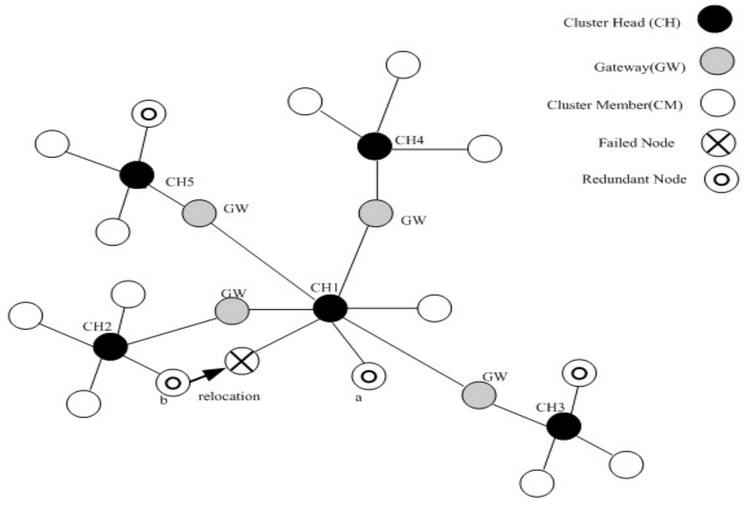

Figure 3: Cluster Based Network

5. COMPLEXITY ANALYSIS

Since the computation cost at each sensor node is small as compared to the communication cost of two sensors. Therefore, our complexity analysis comes in terms of the communication complexity or in terms of number of exchanged messages in the network which is indicative of the energy consumption. Our proposed approach includes two tasks; the first task is to discover the redundant nodes. The second task is to relocate the suitable redundant node. In the redundant discovery task, each sensor node discovers its coverage neighbors by sending its location information along with its ID to its neighbors. If n is the total number of nodes in the network and N_{avg}, the total number of messages of this step will be n* N_{avg} exchanged messages are sent.

In our cluster based relocation task, two cases are introduced. In the first case, there exist redundant nodes inside the cluster that has failed report and we will have the following exchanged messages:

- One request message from the failed node.
- 2 * CN recover messages to the neighboring CHs.
- 2 * CN response messages from the CHs.
- t messages to invoke the nearest redundant node for replacing the failed node (t = 1 if the nearest redundant is in the same cluster of the failed node, otherwise t = 3) Therefore, the total number of exchanged messages for this case is: $n * N_{avg} + (4 * CN + t + 1)$.

In the second case, the cluster's neighbors will forward the recover message to their neighbors until they found redundant node and then send back the message to CH of the failed node. Therefore, in this case, we have:

- One request message from the failed node,
- m_1 recover messages sent in time T_{th},
- m_2 respond messages in time T_{th}, and
- t messages to invoke the nearest redundant node.

Therefore, the total number of exchanged messages for this case is $n * N_{avg} +(m_1 + m_2 + t + 1)$

6. PERFORMANCE EVALUATION

The evaluation of our proposed algorithm includes the following two parts: the evaluation of the redundancy discovery algorithm, and the evaluation of relocation algorithm. In our simulation, the nodes are organized as clusters by refinement of the Linked cluster algorithm (LCA) [29]. First, each node broadcasts its ID and listens to transmission of other nodes. In the next round, a node broadcast the set of neighbors that it heard from and thus every node will eventually know its 1-hop and 2-hops neighbors. The idea is to pick a node x at random as the first CH and assign its neighbors to such first cluster. Then the node y with the lowest ID in the cluster is nominated as a CH. The neighbors of y that are not reachable to x would join the second cluster. The procedure is repeated for the third cluster and so on.

6.1. Redundancy Evaluation

In our simulation environment we deployed 1000 sensors, uniformly and randomly distributed over the region of 200 m $\times$ 200 m. The environment is covered by only 238 nodes with sensing range radius (R_s) equal to 10 m. Figure 4 shows the relation between the total number of nodes and the number of redundant nodes. This shows that as the number of deployed nodes increases the number of the redundant nodes will increase with the same number and the algorithm can be executed with any number of nodes(i.e., does not require a special density of the nodes). Figure 5 shows the relation between sensing range (R_s) and the number of redundant nodes. It is clear to say that the number of redundant nodes increases by increasing the sensing range.

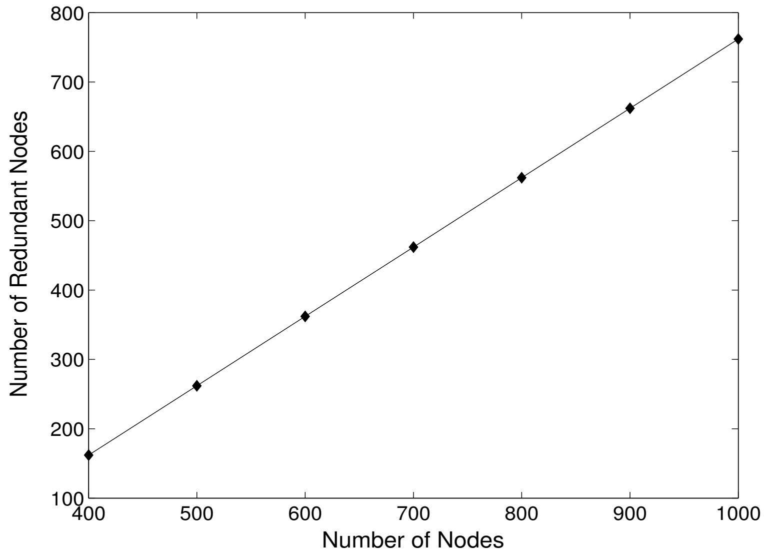

Figure 4: The relation between the total number of nodes and the number of redundant nodes

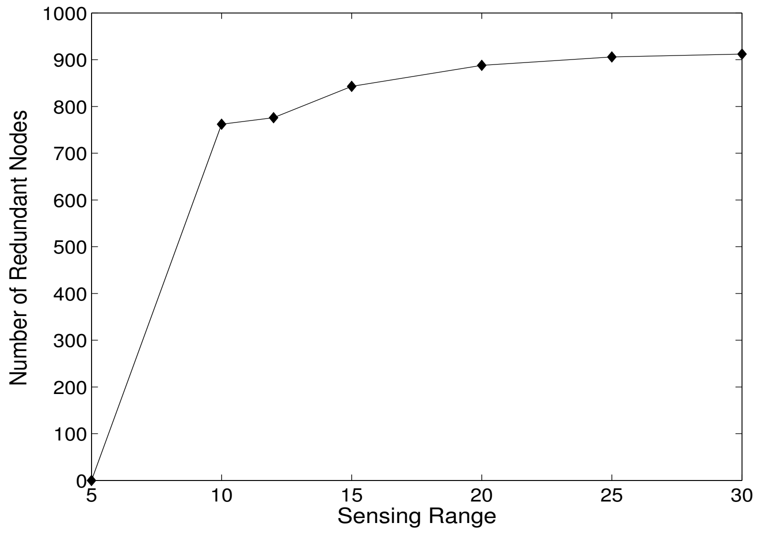

Figure 5: The relation between the sensing range and the number of redundant nodes

6.2. Relocation Evaluation

In this section, we give the evaluation of our relocation algorithm by testing consumed energy and the number of messages, then the results will be compared with the zone based relocation algorithm. From the data of the mobile sensor platform [30], the mobile sensor consumes 37.96J to move one meter and consumes about 0.1J to send a message. Figure 6 shows the energy consumed in the relocation process versus the distance between the redundant node and the failed node. Figure 7 shows the total number of exchanged messages for each node failure's relocation.

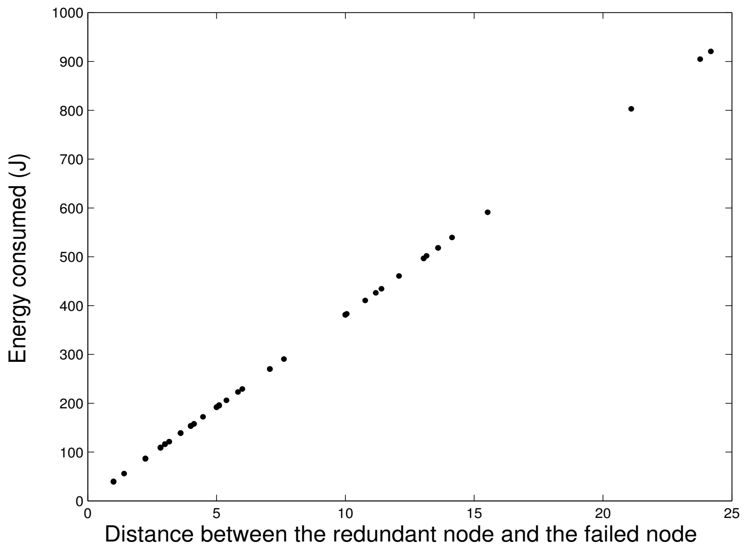

Figure 6: Energy consumed

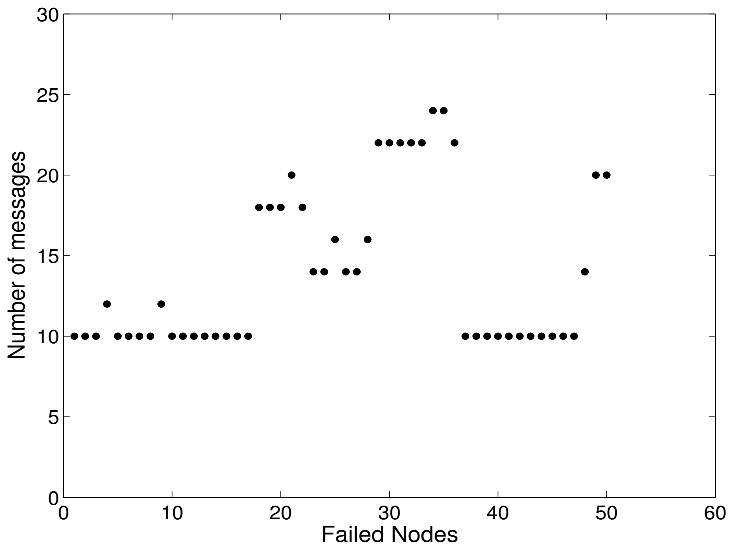

Figure 7: Total number of messages

6.2.1. Zone Based Relocation Vs Cluster based relocation

In this section, we compare between cluster based relocation algorithm and the zone Based Relocation Protocol in term of energy consumed and the number of messages. The simulation result in Figure 8 and Figure 9 shows that the consumed energy and the number of messages can be significantly reduced in the cluster based relocation algorithm.

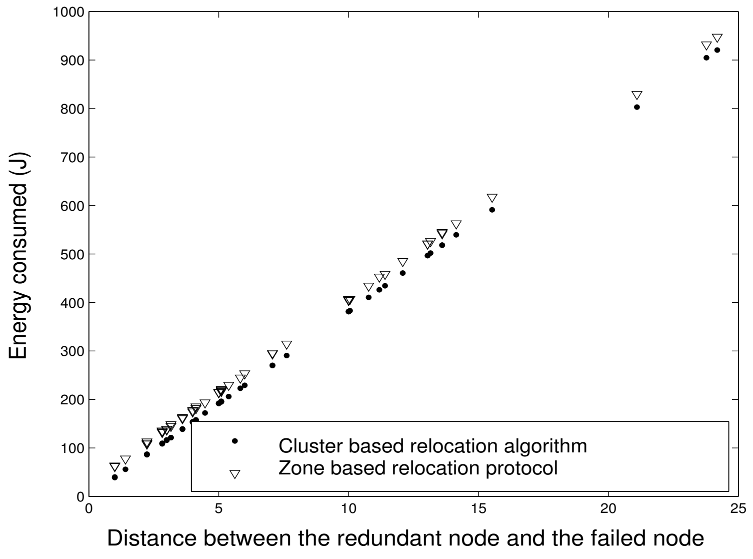

Figure 8: Energy consumed

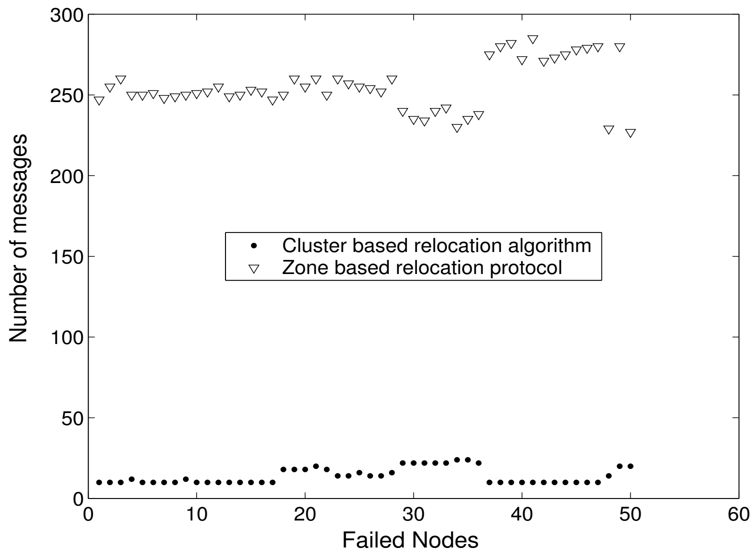

Figure 9: The total number of messages

7. CONCLUSION

In this paper, we have proposed two algorithms, the first one is to discover the redundant nodes by sectorizing the sensor's sensing range to six sectors and test the coverage of each sector, and the second one is to replace the failed node. We propose a cluster based relocation algorithm where the closest redundant node first is identified with low message complexity, and then it is relocated to the target location as mobile sensors can identify the final locations, and then move there directly. Our scheme doesn't require either a global knowledge about the environment or need any flooding technique. Only a local knowledge about the environment is desired and the relocation process is executed via CHs, and not via the sensor nodes which can reduce the number of relocation messages and so the energy consumed. The evaluation of the redundant discovery and relocation algorithms more efficient than the zone based relocation algorithm. The sensing area of each sensor is assumed to be a disk with radius 10m. This is the ideal case which provides us with a baseline of the sensor placement problem. In future work, we will address varying sensing ranges and investigate such cases.

REFERENCES

[1] C. Cordeiro and D. P. Agrawal, "Ad hoc and sensor networks," World scientific, 2006.

[2] Ahmed M. Khedr and Walid Osamy, "Tracking Mobile Targets Using Rando Sensor Networks," The Arabian Journal Science and Engineering, vol. 32, No. 2B pp. 301-315, October 2007.

[3] Ahmed M. Khedr, "New Mechanism for Tracking a Mobile Target Using Grid Sensor Networks," Computing and Informatics, vol. 28, pp.1001-1021, 2008.

[4] Ahmed M. Khedr, Walid Osamy, "Nonlinear Trajectory Discovery of a Moving Target by Wireless Sensor Network," Journal Computing and Informatics, vol. 29, No.4 pp.1001-1016, 2010.

[5] Ahmed M. Khedr, and Walid Osamy, "Finding perimeter of Query regions in wireless sensor networks," Computing and Informatic Journal, vol. 29, No. , pp.1001-1021, 2010.

[6] G. Anastasi, M. Conti, M. Di Francesco and A. Passarella, "Energy conservation in wireless sensor networks: A survey," Ad Hoc Networks, pp. 537-568, 2009.

[7] D. Wang, B. Xie, and D. P. Agrawal, "Coverage and life time Optimization of Wireless Sensor Network with Gaussian Distribution," IEEE Transaction on mobile computing, vol.7, no.12, pp. 1444-1458, Dec. 2006.

[8] A. Ghosh, and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: A survey," Pervasive and Mobile Computing, pp. 303-334, 2008.

[9] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler, 'Wireless sensor networks for habitat monitoring," In Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, WSNA02, Atlanta, GA, pp. 88-97, September 2002.

[10] D. Wang, D. P. Agrawal, W. Toruksa, C. Chaiwat Pong Sakorn, m. liu, and T. C. keener, "monitoring ambient air quality with carbon monoxide sensor based wireless network, "Communication of ACM, to appear.

[11] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks, " ACM MobiHoc, pp. 300-308, Urbana-Champaign, Illinois, USA, May 2005.

[12] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor Relocation in Mobile Sensor Networks," In Proc. of IEEE INFOCOM, pp. 2302-2312, 2005.

[13] G. Wang, G. Cao, and T. L. Porta, "Movement-Assisted Sensor Deployment," In Proc. of IEEE INFOCOM, vol. 4, pp. 2469-2479, 2004.

[14] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," In Proc. of IEEE INFOCOM, 2003.

[15] J. Wu and S. Yang, "Optimal Movement-Assisted Sensor Deployment and Its Extensions in Wireless Sensor Networks," Simulation Modeling Practice and Theory, vol. 15, pp. 383-399, 2007.

[16] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, "Efficient Placement and Dispatch of Sensors in a Wireless Sensor Network," IEEE Transactions on Mobile Computing, vol. 7, no. 2, pp. 262-274, 2008.

[17] G. Wang. G. Cao, and T. La Porta, "A Bidding Protocol for Deploying Mobile Sensors," IEEE International Conference on Network Protocols (ICNP), pp. 315-324, November 2003.

[18] G.Wang, G. Cao, and T. L. Porta, "Proxy-Based Sensor Deployment for Mobile Sensor Networks," In Proc. of IEEE MASS, pp. 493-502, 2004.

[19] X. Li and N. Santoro, "ZONER: A ZONE-based Sensor Relocation Protocol for Mobile Sensor Networks," IEEE International Workshop on Wireless Local Network, pp. 923-930, 2006.

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

[21] 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, pp. 28-40, 2003.

[22] 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, Ad hoc Wireless, and Peer-to-Peer Networks, 2004.

[23] A. C. C. Sawides and M. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," Proceedings of the ACM/IEEE International conference on Mobile computing and networking, Rome, July 2001.
[24] D. Niculescu, "Ad hoc positioning system," Proceeding of CLOBECOM, San Antonio, November 2001.
[25] J. Albowitz, A. Chen, and L. Zhang, "Recursive postion estimation in sensor networks," ICNP'01, 2001.
[26] K. F. S. Wong, I. W. Tsang, V. Cheung, S.-H.G. Chan, and J.T. Kwok, "Position estimation for wireless sensor networks," Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM2005),St.Louis,MO,USA,Nov2005,Postscript:http://www.cs.ust.hk/jamesk/Papers/globecom05.pdf.
[27] G. Xing, X. WANG, Y. ZHANG, C. LU, R. PLESS, and C. GILL," Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks," ACM Transactions on Sensor Networks, vol. 1, no. 1, pp. 36-72, 2005.
[28] M. Younis, and A. A. Abbasi, "A survey on clustering algorithms for wireless sensor networks," Computer Communications, PP. 28262841, 2007.
[29] A. Ephremides, J.E. Wieselthier, D.J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," Proceedings of IEEE, pp. 5673, 1987.
[30] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, "Robmote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks," Proceedings of The IEEE International Conference on Robotics and Automation (ICRA), 2002.

Ahmed M. Khedr received his B.Sc degree in Mathematics in June 1989 and the M.Sc degree in the area of optimal control in July 1995 both from Zagazig University, Egypt. In July 1999 he recived his M.Sc and in March 2003 he received his Ph.D. degree both in area of Computer Science and Engineering from University of Cincinnati, Ohio, USA. From March 2003 to January 2004, he was a research assistant professor at ECECS Department University of Cincinnati, USA. From January 2004 to May 2009, he worked as assistant professor at Zagazig University, Egypt, and from June 2009 till now he is working as associate professor at the Department of Computer Sciences, college of computers and Information Systems, Taif University, KSA. In June 2009, he awarded the State Prize of Distinction in Advanced Technology. He has coauthored 35 works in journals and conferences relating with optimal control, Wireless Sensor Networks, Decomposable Algorithms, and Bioinformatics.

Hagar Ramadan is currently a M.Sc student in faculty of science, Zagazig University, Egypt. She received her B.Sc degree in computer science in 2006. her research interests in Mobile Wireless Sensor Networks.