A Fuzzy Logic Based Sensor Relocation Betterment for Mobile Wireless Sensor Networks

Ehsan Khazaei¹, Mahmood Fathi² and Mohammad Mehrani³
1. Department of Computer Science, Islamic Azad University, Abdanan Branch, Abdanan 76119-69718, Iran
2. University of Science and Technology, Tehran 13114-16846, Iran
3. Department of Computer Science, Islamic Azad University, Shoushtar Branch, Shoushtar 645171117, Iran

Received: August 14, 2011 / Accepted: August 22, 2011 / Published: March 31, 2012.

Abstract: Using mobile sensor networks has increased as well in recent years. It is because of the movement ability of sensor nodes which can be very useful in such networks. Having failure for network nodes occurs usually in wireless sensor networks which can be avoided by having the ability of movement in sensor nodes. It means that when a node fails consequently a hole appears in the coverage area of the network. By moving mobile nodes in a mobile sensor network the holes can be covered. This process should not lead the network to have much increasing in energy consumption factor. In this paper we present an algorithm for wireless mobile sensor networks to achieve at full coverage during network functioning. This algorithm assumes some redundant nodes in the network using for removing possible holes from coverage area. The presented algorithm executes in two phases: recognizing redundant nodes and selecting best ones for relocating that are both managed by cluster heads. Also the proposed method employs fuzzy logic to find out the best path for moving the nodes. Simulation results demonstrate that using the presented algorithm has significant improvement in exploring the redundant nodes and relocating operations in comparison with similar works. Also this algorithm has decreased energy consumption in relocation operations and consequently caused increasing in network lifetime.

Key words: Mobile sensor network, fuzzy logic, redundant node, relocation, rows and columns.

1. Introduction

Due to many attractive characteristics of sensor nodes such as small size and low cost, sensor networks [1-5] have become adopted to many military and civil applications including military surveillance, smart homes [6], remote environment monitoring, and in-plant robotic control and guidance. In order to properly sense the phenomena of interest, sensor nodes must be deployed appropriately to reach an adequate coverage level for the successful completion of the issued sensing tasks [7-8].

In many potential working environments, such as remote harsh fields or disaster areas sensor deployment cannot be performed manually or precisely. In addition, once deployed, sensor nodes may fail, requiring nodes to be moved to overcome the coverage hole created by the failed sensor. In these scenarios, it is necessary to make use of mobile sensors [9-10], which can move to provide the required coverage. One example of a mobile sensor is the Robomote [11].

In this paper we address the problem of sensor relocation, i.e., moving previously deployed sensors to overcome the failure of other nodes, or to respond to an occurring event that requires that a sensor be moved to its location. This sensor relocation is different from existing work on mobile sensors which concentrate on sensor deployment; i.e., moving sensors to provide certain initial coverage [9-10, 12-13]. Compared with sensor deployment, sensor relocation has many special

---

Ehsan Khazaei, graduate student, research fields: wireless sensor networks, Ad hoc. E-mail: eh.khazaei@gmail.com.
Mahmood Fathi, Ph.D., professor, research fields: image processing and processor design, wireless networks, Ad hoc. E-mail: mahfathi@iust.ac.ir.
Corresponding author: Mohammad Mehrani, graduate student, research fields: wireless networks, Ad hoc. E-mail: m.mehrani@iau-shoushtar.ac.ir.
difficulties. 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 to replace it as soon as possible. Second, relocation should not affect the application currently using the sensor network, which means that the relocation should minimize its effect on the current sensing topology. Finally, since movement may be much more expensive in terms of energy than computation and communication, any algorithm must balance energy costs with response time. In particular, care must be taken to balance the energy costs of an individual node with the overall network energy cost to ensure maximum network lifetime [14].

In this paper, we propose an energy efficient algorithm for relocating mobile sensors in a balanced manner, and at the same time, maintaining the original sensing topology is satisfied as much as possible. In our work, the algorithm is divided into two phases: finding the failure node and relocating a node to that node. First of all, network nodes are to be divided into some grids and sub grids. For the second phase, we use fuzzy logic to achieve at a good balance between energy and distance for determining the sensor relocation path.

The rest of the paper is organized as follows. In section 2, we propose the algorithm. In section 3, we show performance evaluations and Section 4 concludes this paper.

2. The Proposed Algorithm

The problem that proposed algorithm attempts to solve is to cover all over the region during any given time of network functioning. The algorithm is divided into two phases: finding the failure node and relocating a node to that node. First of all, network nodes are to be divided into some grids and sub grids as can be seen in Fig.1. This can be done using three different algorithms VEC, VOR, Minimax, etc.. After doing such and after achieving at Fig.1 there will be some redundant nodes in some grids.

The network nodes are grouped in several grids, so these grids are seen in different rows and columns. The number of redundant nodes is depending on the number of sensor nodes and the size of the region that the nodes are dispersed. The task of redundant nodes is to cover the possible existing holes between the covered parts of the covered nodes.

The network nodes are mobile and can change their positions if necessary. It means that after some holes appear in the network some nodes can cover and remove these holes. This can be done by using relocation in the network. Relocating is to be satisfied via redundant nodes and some other regular nodes. Relocation will lead to cover all the holes and consequently achieve at full coverage in the network.

Those of cluster heads who have redundant nodes in their grids send some messages to the other cluster heads existing in the same columns to let them know about their redundant nodes. Also the positions of the redundant nodes are sent via these messages. This can be done using multi hob transmitting algorithms.

When a regular node fails its cluster head will be aware about this event because cluster head does not receive any message from the regular node and will find that a hole has appeared in its grid. Thus, cluster head tries to cover the hole using relocating algorithm. For doing such, cluster head first of all informs the other cluster head existing in the same row via multi hob transmitting algorithms. If any neighbor cluster head recognizes any redundant nodes existing in the
same column it will inform the cluster head of the failure node about the names and positions of the redundant nodes. So, the cluster head belonging to failure node knows everything about all the redundant nodes of the network and decide to select the best and most suitable redundant node to start relocating in a significant way. The same presented algorithms [1] are able to recognize just one redundant node for each hole and use just distance factor for denoting the redundant node and relocating path, meanwhile they pay less attention to energy factor. The presented algorithm is able to find and recognize all the redundant nodes existing in the network after a hole appears. Also a relocation path is satisfied for every one of the redundant node as the starting node for relocation. Cluster head can do all the mentioned works to remove the holes.

Assume that a node fails and \( k \) redundant nodes are recognized to start relocating. Cluster head calculates an energy-distance factor for every one of the redundant nodes.

Denoting energy-distance factor by cluster head is done using fuzzy logic. Mamdani’s fuzzy inference method is used as the fuzzy inference method for this approach. Distance and energy are two factors considered for every one of the relocation paths. The membership functions of distance factor, energy factor and output are shown in Fig. 2, Fig. 3, and Fig. 4.

All the paths are given to the fuzzy system as inputs, then after defuzzitation the best path will be selected by using Center of Gravity Defuzzifier (CGD) method. The fuzzy rule base for Distance and energy factors is shown in the Table 1.

After calculating energy-distance factor for each redundant node, cluster head chooses the best one as the starting node for relocation such that an energy efficient relocation is satisfied. Relocating is to be applied by utilizing cascading movement. To understand cascading movement assume a node fails and a redundant node is selected as the starting node for relocating. If the redundant node moves into the direct path to reach the hole it will lose lots of energy and consequently may fail when it arrives to this point. Also moving into this path takes so much time and there would be a big delay in covering the hole which can be a negative score for the functionality of the algorithm.
A fuzzy logic based sensor relocation betterment for mobile wireless sensor networks

Fig. 5 The simulation results.

A useful technique is instead of moving the redundant node directly towards the hole, some other nodes existing at the path would have a shift in their positions towards the next nodes existing at the path in a parallel way. So, all of these nodes contribute in relocating. By doing such, the relocation operations take less time, overall energy consumption will decrease and no any redundant or regular node will fail during these operations. These operations are known as cascading movement. The words must be said is the proposed algorithm selects the shortest path for relocation by using Dijkstra’s algorithm. After replacement the cluster head belonging to redundant node informs all of its neighbor cluster heads in the same column that one of its redundant node has been used.

By doing such, we can make sure that full coverage in mobile sensor network is satisfied. The algorithm can be considered as shown as follows:

Step1: dividing network into some grids, if there is a faulty node then go to step2.
Step2: recognizing redundant nodes.
Step3: finding all the paths between faulty and redundant nodes using Dijkstra’s algorithm.
Step4: finding the optimum path using fuzzy logic and considering energy and distance factors.
Step5: cascade movement between redundant node to faulty node.
Step6: revealing decrease of redundant nodes by cluster head of the redundant node to same column cluster heads.

3. Performance Evaluation

We used MATLAB software to simulate our algorithm and compare it with other similar work. We assumed 5j energy for each node at the beginning of the algorithm. The number of sensors is 1,000 and they are dispersed in a 100 × 100 area. Energy and distance are the basic factors for comparing the algorithms. The simulation results are shown in Fig. 5.

4. Conclusions

In our paper, we explained the problem of sensor relocation which means replacing the faulty sensor node by getting assist from redundant sensor node. We divided all the operations into two parts: (1) Finding the redundant sensor node and (2) Relocation solution. The first part is done by dividing the network nodes into some grids while the fuzzy logic is satisfied for the second part meaning that energy and distance factors
are defined for every one of the paths between failure and redundant nodes. Also the best path from energy and distance point views is to be chosen using fuzzy logic.

Simulation results verified that the proposed solution outperforms others in terms of shortest path for relocation and energy of path.

References