Journal of Computers Vol. 29 No. 6, 2018, pp. 193-200
doi:10.3966/199115992018122906018

Partition Connectivity Recovery Based on Relay Node Deployment for Wireless Sensor Networks

Ji-Kai Zhang, Dan Tao*

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, 100044, China
{13211143, dtao}@bjtu.edu.cn

Received 9 January 2018; Revised 17 January 2018; Accepted 22 January 2018

Abstract. In this paper, we address the problem of partition connectivity recovery based on relay node deployment for wireless sensor networks. We firstly propose a graph theory based method to accurately detect partitions in the network which consists of vast low-energy sensor nodes. To restore the communication links between these partitions, we present a heuristic Steiner tree based partition recovery algorithm by deploying high-energy relay nodes. The suitable quadrilaterals or triangles are chosen to connect the disjoint partitions and their Steiner nodes are found. The minimum number of relay nodes are placed on the edges of Steiner tree. Experimental results show that our algorithm can achieve partition connectivity recovery for wireless sensor networks with a smaller number of relay nodes and less energy consumption of network communication compared to MST algorithm.

Keywords: graph theory, partition connectivity recovery, relay node deployment, Steiner tree, wireless sensor networks

1 Introduction

Stable network topology becomes one of the most important problems in the context of wireless sensor networks (WSNs) to guarantee data exchange and data transmission. However, energy-limited sensor nodes can be easily destroyed due to exhausting their batteries, hardware faults, externally inflicted damage caused by natural or human factors. The wireless sensor network may be separated into multiple disjoint partitions which cannot communicate with each other. Therefore, linking these disjoint partitions to re-establish a connected network topology is necessary to maintain the functional network operations [1].

At present, the common partition connectivity recovery methods for WSNs can be classified into two categories [2]. One is relocating movable sensor nodes. It requires that sensor nodes have moveable ability with extra auxiliary devices, so its realization cost is large. In addition, this method tends to trigger a cascaded movement of nearby sensor nodes resulting in increased overhead and widening the scope of the recovering throughout the network [3]. The other is redeploying extra relay nodes, which has no special requirements on the existing sensor nodes. It has been shown to be an effective method for restoring network connectivity. Here, the network connectivity recovery can be realized by redeploying extra relay nodes here.

In this paper, we address the problem of partition connectivity recovery in wireless sensor networks with relay node deployment. The main contributions of our works are two. (1) We propose a graph theory based disjoint partition detection algorithm to find partitions in the network. (2) We present a heuristic partition recovery algorithm by redeploying a small number of costly, but more powerful relay nodes, which can be illustrated in Fig. 1. A backbone communication network can be constructed by these relay nodes based on Steiner tree theory, in order to prolong network lifetime while preserving network connectivity. The shortest path feature of Steiner tree can ensure that the data transmission of the backbone network is low latency and energy consumption.

* Corresponding Author

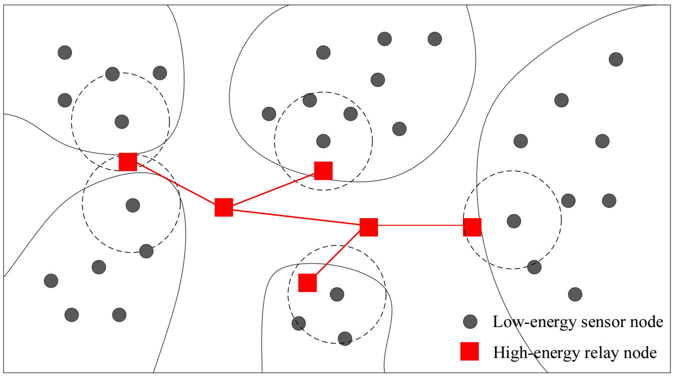


Fig. 1. Partition connectivity recovery based on relay node deployment

The remainder of this paper is organized as follows. Section 2 overviews the related work. In Section 3, a partition connectivity recovery algorithm based on relay node deployment for WSNs is proposed. Performance evaluations are given to testify the performance of the proposed solutions in Section 4 and we conclude this paper in Section 5.

2 Related Work

Extensive studies have been conducted on partition recovery problem in the context of WSNs, this problem can be divided into two subcategories: relocating movable sensor nodes and redeploying extra relay nodes.

2.1 Relocating Movable Sensor Nodes

Akkaya et al. [4] proposed a PADRA to detect possible partitions, and restored the network connectivity through controlled relocation of movable nodes. Imran et al. [5] presented a distributed partitioning detection and connectivity restoration algorithm to tolerate the failure of actors. DCR proactively identified actors that were critical to the network connectivity based on local topological information, and designated appropriate, preferably non-critical, backup nodes. Abbasia et al. [6] proposed a distributed least-movement topology repair algorithm, which strived to relocate the least number of nodes and reduce the traveled distance and message complexity.

2.2 Redeploying Extra Relay Nodes

Most of researches achieve network connectivity restoration by deploying extra relay nodes. To prolong network lifetime while preserving network connectivity, Lloyd et al. [7] deployed the minimum number of relay nodes to achieve communication with other sensor or relay nodes. The approach proposed in [8] opted to re-establish connectivity using the least number of relays while ensuring certain quality in the formed topology. Unlike the existing schemes that formed a minimum spanning tree among the isolated segments, the proposed approach established a topology that resembled a spider web, for which the segments were situated at the perimeter. Lee et al. [9] presented an approach for federating segments in the network by populating the least number of relay nodes. The optimization problem was then mapped to finding the cell-based least-cost paths that collectively met QoS requirements. Lee et al. [10] proposed heuristics which was QoS-aware relay node placement using minimum Steiner tree on convex hull. Chen et al. [2] proved the problem of finding the minimum relay nodes was NP-hard and hence heuristics methods were preferred. And then they presented a Steiner tree algorithm to address this problem.

3 Partition Connectivity Recovery based on Relay Node Deployment

3.1 Problem Statement

This paper focuses on the problem of relay node deployment involved in partition connectivity recovery in wireless sensor network. Two kinds of nodes are contained in the WSN: (1) low-energy sensor nodes deployed initially, are responsible for collecting and transmitting sensing data; and (2) high-energy relay nodes re-deployed, are responsible for linking disjoint partitions to recovery a connected network topology and achieve data transmission.

3.2 Disjoint Partition Detection

How to detect disjoint partitions in the WSN is a principal problem to be solved. Assume that a WSN can be modeled as a communicating graph $G(V,E)$, where V is the node set, and E is the edge set. For a pair of node $v_1,v_2 \in V$, the edge $(v_1,v_2) \in E$ if $d(v_1,v_2) \leq 2R$, where R denotes the communicating radius of low-energy sensor nodes. Then we define the partition as follows. Given a sub-graph $G_1(V_1,E_1) \in G$, if for any node $v_i \in V_1$, $v_j \in V\text{-}V_1$, $d(v_i,v_j)>2R$, then we call G_1 as a partition. To some extent, the number of partitions in a network reflects the connectivity performance of network.

We take *FindPartition* algorithm (see Algorithm 1) to find all the partitions in the WSN. Assume the network graph G be represented as an adjacency list. Partitioning a WSN into several isolated partitions helps to detecting communication holes. And thus, we eliminate these communication holes by re-deploying a small number of high-energy relay nodes to restore network connectivity.

Algorithm 1. *FindPartition(v)*

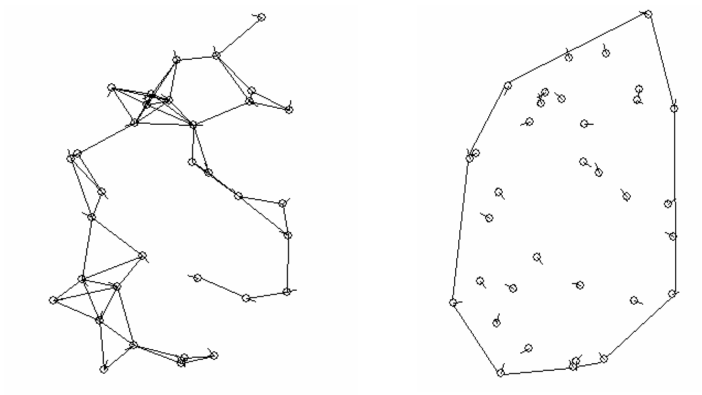
```
{v∈V, V denotes the node set of a wireless sensor network.};
//use a visit flag array Visited
Visited[v]=TRUE; // the node v is visited
v=*v.first; // take the first node
While (v is not NULL) Do
  If (!visited[*v.vertex])//if the node is not visited
    FindPartition (*v.vertex);
    v=*v.next;
  Endif
Endwhile
```

3.3 Intra-partition Relay Node Deployment

After finding multiple isolated partitions in the WSN, the sensor nodes in the same partition can communicate with each other. We firstly discuss the intra-partition relay node deployment. Assume that the communicating radius of a relay node be represented as R_r, where $R_r>R$. A high-energy relay node deployed in a partition can be used to forward sensing data from sensor nodes to sink node. For low-energy sensor nodes within the partition, if they can forward their sensing data to certain relay node, this relay node has to lie in the communicating radius of a sensor node (R), as is shown in Fig. 1.

One of the simplest intra-partition relay node deployment method is finding the center of the partition and placing relay node at this location. However, the number of relay nodes re-deployed in this deployment method won't be the least. There is no doubt that it will increase the deployment cost of relay nodes. Hence, this paper designs a scheme to deploy intra-partition relay node for each partition based on convex hull which is the most ubiquitous structure in computational geometry.

We can always find a convex hull to contain all the low-energy sensor nodes in a partition, as shown in Fig. 2. A relay node will be placed at the location which must satisfies two conditions: (1) the distance to certain sensor node on the convex hull is no more than R to guarantee the information exchange between intra-partition sensor nodes and the relay node; and (2) the distance to the center of region is the least to ensure a smaller number of relay nodes to be deployed.



(a) (b)

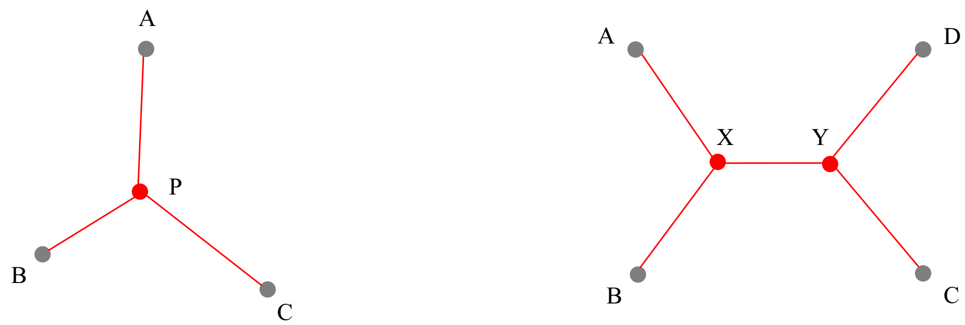
Fig. 2. Convex hull of a partition in the network

3.4 Inter-partition Relay Node Deployment

After determining all the location information of intra-partition relay nodes, inter-partition relay nodes will be deployed to connect each disjoint partition together to achieve the whole network communication connectivity. Considering the requirements of transmission delay and energy consumption, a Steiner tree will be built which is based on intra-partition relay nodes to guarantee the shortest communication path of the whole high-energy communication network.

The Steiner tree problem [11] is defined as follows: given a set of points (vertices), interconnect them by a network (graph) of shortest length, where the length is the sum of the lengths of all edges. The difference between the Steiner tree problem and the minimum spanning tree problem is that [12], in the Steiner tree problem, extra intermediate vertices and edges may be added to the graph in order to reduce the length of the spanning tree. These new vertices introduced to decrease the total length of connection are known as Steiner points (vertices). It has been proved that the resulting connection is a tree, known as the Steiner tree.

In Fig. 3(a), we show an example of the Steiner tree of three points, A, B and C, where an extra Steiner point P is added. In our work, we utilize the idea of the Steiner tree of four points. In this case, two extra Steiner points X and Y are added. Fig. 3(b) shows an example of a Steiner tree of four points, A, B, C and D.



(a) A Steiner tree of three points (b) A Steiner tree of four points

Fig. 3. An example of a Steiner tree

The essence of inter-partition relay node deployment algorithm is to find the minimum Steiner tree. It has been proved that the solving of minimum Steiner tree is an NP-hard problem. Therefore, when the node scale is large, heuristic methods are preferred. The process of finding minimum Steiner tree contains three core problems:

(1) the suitable quadrilaterals or triangles are chosen to connect the disjoint partitions (refer to literature [2]);

(2) their Steiner nodes are found;

(3) the minimum number of relay nodes are placed on the edges of Steiner tree.

Assume *Dis*(*Rei*,*S*) represent the distance between the intra-partition relay node *Rei* and a Steiner node *S*. The calculation method of the number of relay nodes to be deployed along an edge of Steiner tree can be described as *Ceiling*(*Dis*(*Rei*,*S*)/R_r-1).

The algorithm description of Steiner tree based inter-partition relay node deployment can be given in Algorithm 2.

Algorithm 2. *Inter-partition relay node deployment*

Step 1: Enumerate all the combinations of non-degenerate convex quadrilaterals and store them in list *Q*, and sort *Q* by the perimeter in ascending order. For each non-degenerate convex quadrilateral *q* in *Q*, if the number of disjoint partitions that the vertexes of *q* represent is larger than 3, then compute Steiner edge and deploy relay nodes along the Steiner edge.

Step 2: Enumerate all the combinations of triangles and store them in list T, and sort *T* by the perimeter in ascending order. For each triangles *t* in *T*, if the number of disjoint partitions that the vertexes of *t* represent is larger than 2, then compute Steiner edge and deploy relay nodes along the Steiner edge.

Step 3: For the remaining two partitions, select the shortest Steiner edge which connects these two partitions to form a connected network.

In Fig. 4, we illustrate the generation of minimum Steiner Tree. The minimum non-degenerate convex quadrilateral is composed of relay nodes Re2, Re3, Re4 and Re5. Two Steiner nodes of this quadrilaterals are found. Several relay nodes are then placed to the appropriate positions on the edges of Steiner tree to connect four partitions in the network (see Fig. 4(a)). If three angles of a triangle are less than 120°, then the Steiner point can be found within the triangle. Otherwise, if one angle of a triangle is greater than 120°, this is a triangle with degeneration. Re1, Re6 and Re7 constitute a degenerate triangle. Hence, we consider forming a non-degenerate triangle which consists of Re4, Re6 and Re7. One Steiner node of this triangle is found. Extra relay nodes are then deployed on the edges of Steiner tree to connect three partitions in the network (see Fig. 4(b)). After the above operation, the current number of partitions is 2. In this case, we find the shortest edge to connect these two partitions into a connected network (see Fig. 4(c)).

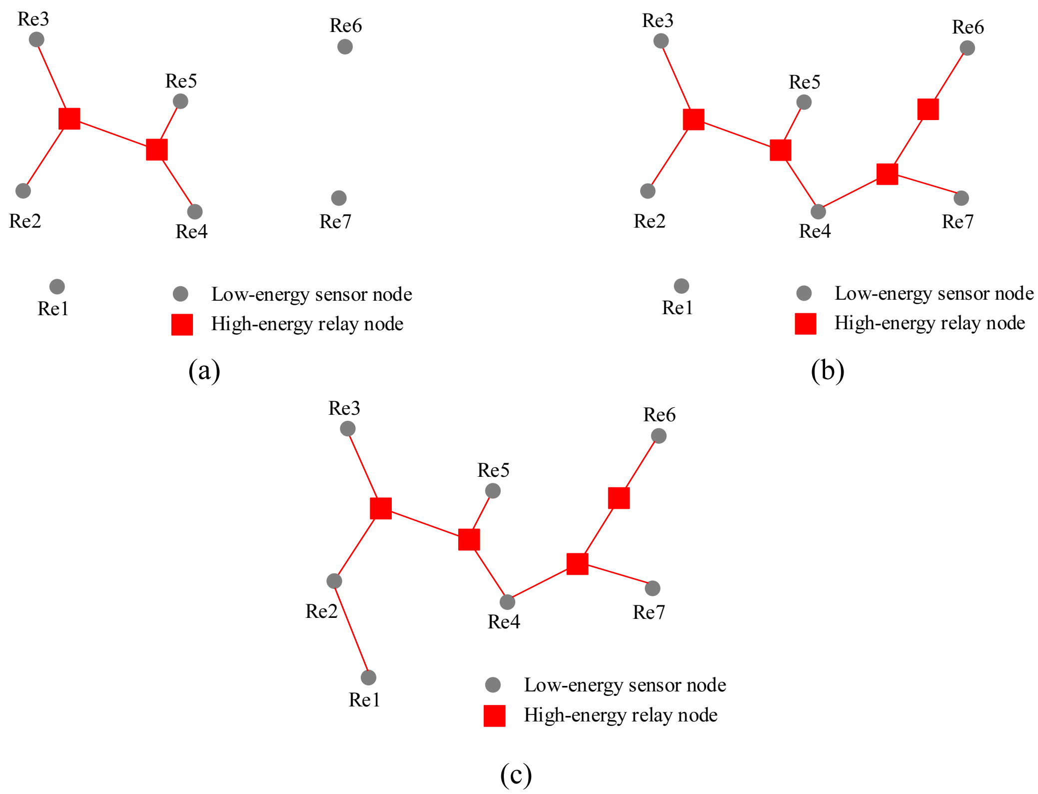


Fig. 4. Generation of minimum Steiner tree

4 Performance Evaluation

In this section, we evaluate the performance of the proposed solutions via extensive simulations. Without the loss of generality, sensor nodes are deployed in a region with the size of 1500m*1500m. We explore two primary parameters: communicating radius (R), the number of partitions (N_p), the number of sensor nodes (N) and evaluate their effects on the number of relay nodes (N_r). Each result shown here is the statistical average of 20 simulations.

Different from other works, in the proposed algorithm, high-energy relay nodes re-deployed are used for data forwarding in each partition. Here, we only compare the number of inter-partition relay nodes deployed in our quadrilateral Steiner tree based algorithm and Minimum Spanning Tree (MST) algorithm.

4.1 Effect of Communicating Radius

Fig. 5 gives the effect on the number of relay nodes from the communicating radius of sensor nodes. In this example, the number of partitions N_p is 9. When some parameters (*e.g.,* the size of region, the number of sensor nodes) are unchanged, the length of connection path of Steiner tree is relatively fixed. From the data, we can find that as R increases from 50m to 200m, N_r required to achieve network communication connectivity will gradually reduce.

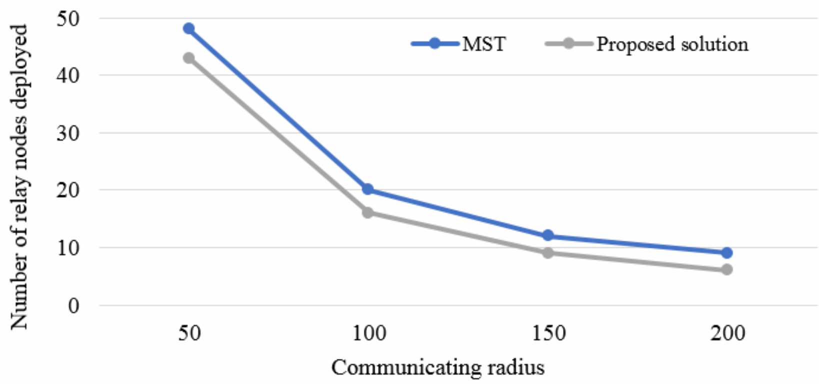


Fig. 5. Effect of the communicating radius of sensor nodes

For example, when R=50m, the numbers of relay nodes required N_r are 43 and 48 in our proposed solution and MST algorithm, respectively. With the increase of the communicating radius, the number of relay nodes required is less than that in MST algorithm. When R=200m, N_r are 6 and 9 in our proposed algorithm and MST algorithm, respectively. It is mainly because that the increase of communicating radius will lead to the decrease of the length of connection path.

4.2 Effect of the Number of Partitions

Fig. 6 shows the effect on the number of relay nodes from the number of partitions (R=100m). With the increase of N_p, N_r required in our proposed algorithm and MST algorithm exist a directly proportional relationship. Obviously, the more N_p is, the more N_r required to restore communication connectivity will become. For instance, when N_p=4, N_r required in our proposed algorithm and MST algorithm are 13 and 15, respectively. This is because that the more N_p is, the more the number of partitions which can be connected by using quadrilateral Steiner tree becomes.

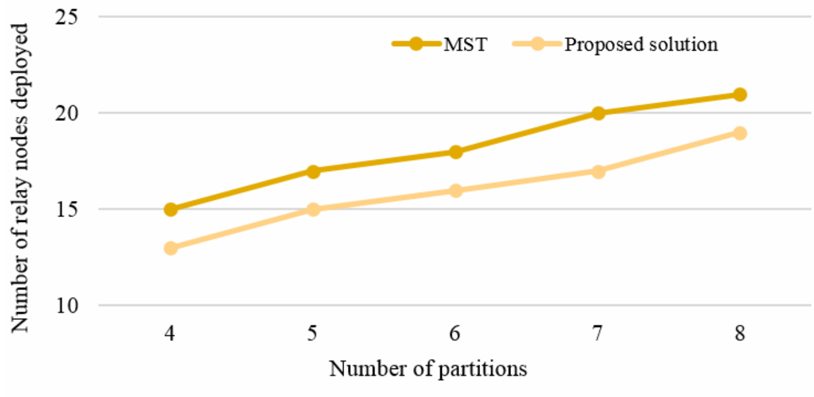


Fig. 6. Effect of the number of partitions

4.3 Effect of the Number of Sensor Nodes

Fig. 7 shows the effect on the number of relay nodes from the number of sensor nodes. In this example, R=100m. We can get that with the increase of N, N_r required will gradually decrease. For example, when N=75, N_r required are 12 and 15 in our proposed solution and MST algorithm, respectively. When the size of region, the communicating radius are unchanged, the larger number of sensor nodes will alleviate less partitions, and thus the less number of relay nodes is required.

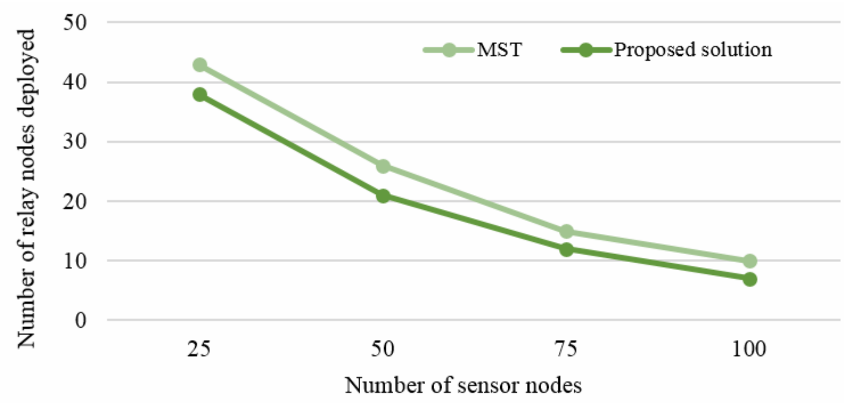


Fig. 7. Effect of the number of sensor nodes

In conclusion, experimental results show that compared to MST algorithm, our proposed solution can achieve partition connectivity recovery for WSNs with a smaller number of relay nodes and less energy consumption of network communication.

5 Conclusion

In this paper, we propose a partition connectivity recovery solution based on relay node deployment for wireless sensor networks. Intra-partition and inter-partition relay node deployment solutions are designed. Extensive simulation is conducted to verify the effectiveness of our proposed solution and we give a detailed discussion on the effects of some primary parameters.

In our future work, we plan to investigate the partition connectivity recovery scheme with the consideration of transmission delay and computation complexity.

Acknowledgements

This work was partly supported by the Fundamental Research Funds for the Central Universities under Grant No. 2016JBM011 and the National Science Foundation of China under Grant No. 61872027.

References

[1] N.N. Qin, D. Wu, Y.H. Yu, Connectivity recovery algorithm in partition based on triangle steiner tree, Chinese Journal of Sensors and Actuators 29(3)(2016) 423-428.

[2] H.S. Chen, K. Shi, Quadrilateral steiner tree based connectivity restoration for wireless sensor networks, Chinese Journal of Computers 37(2)(2014) 457-468.

[3] M. Younis, R. Waknis, Connectivity restoration in wireless sensor networks using steiner tree approximations, in: Proc. IEEE Global Telecommunications Conference, 2010.

[4] K. Akkaya, F. Senel, A. Thimmapuram, S. Uludag, Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility, IEEE Transactions on Computers 59(2)(2010) 258-271.

[5] M. Imran, M. Younis, A.M. Said, H. Hasbullah, Localized motion-based connectivity restoration algorithms for wireless sensor and actor networks, Journal of Network and Computer Applications 12(2)(2011) 1-13.

[6] A.A. Abbasi, M. Younis, U. Baroudi. Restoring connectivity in wireless sensor actor networks with minimal node movement, in: Proc. the 7th International Conference on Wireless Computations and Mobile Computing, 2011.

[7] E.L. Lloyd, G. Xue, Relay node placement in wireless sensor networks, IEEE Transactions on Computers 56(1)(2006) 134-138.

[8] F. Send, M. Younis, K. Akkaya, A robust relay node placement heuristic for structurally damaged wireless sensor networks, in: Proc. the IEEE 34th Conference on Local Computer Networks, 2009, pp. 633-640.

[9] S. Lee, M. Younis, EQAR: effective QoS-aware relay node placement algorithm for connecting disjoint wireless sensor subnetworks, IEEE Transaction on Computers 60(12)(2011) 1772-1787.

[10] S. Lee, M. Lee, QRMSC: efficient QoS-aware relay node placement in wireless sensor networks using minimum steiner tree on the convex hull, in: Proc. International Conference on Information Networking, 2013.

[11] F.K. Hwang, D.S. Richards, P. Winter, The steiner tree problem, Elsevier, North-Holland, 1992.

[12] C.K. Liang, C.H. Lee, J.D. Lin, Steiner trees grid routing protocol in wireless sensor networks, in: Proc. IEEE International Conference on Wireless Communications Networking & Information Security, 2010.