International Innovative Research Journal of Engineering and Technology
ISSN: 2456-1983 Vol: 5, Issue No: 1, September 2019

Retrieving Relay Node Connectivity for WSN

Zarul Fitri Zaba[1], Zhamri bin Che Ani[2]
[1]School of Computer Sciences, Universiti Sains Malaysia, 11800 Minden, Pulau Pinang, Malaysia
[2]School of Computing, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia

Abstract: In this article, we discuss the issue of retrieving partition connectivity for wireless sensor networks based on relay node deployment. First, we suggest a graph theory-based technique for correctly detecting network partitions consisting of vast low-energy sensor nodes. We present a heuristic Steiner tree-based partition recovery algorithm by deploying high-energy relay nodes to restore the communication connections between these partitions. To connect the disjoint partitions, the appropriate quadrilateral or triangles are selected and their Steiner nodes are discovered. On the edges of the Steiner tree, the minimum amount of relay nodes is put. Experimental results show that compared to the MST algorithm, our algorithm can achieve partition connectivity recovery for wireless sensor networks with fewer relay nodes and lower network communication energy consumption.

Keywords: Graph theory, partition connectivity recovery, relay node deployment, Steiner tree, Wireless sensor networks.

I. INTRODUCTION

Stable network topology is one of the most significant issues in the field of wireless sensor networks (WSNs) to ensure information exchange and data transmission. Due to exhausting their batteries, hardware faults, externally inflicted damage induced by natural or human factors, energy-limited sensor nodes can be easily demolished. The network of wireless sensors can be separated into multiple partitions that cannot communicate with each other. Therefore, to maintain functional network operations, it is necessary to link these disjoint partitions to restore a connected network topology [1].

The common methods of partition connectivity recovery for WSNs can currently be classified into two categories [2]. One is the relocation of nodes with movable sensors. It requires sensor nodes with additional auxiliary devices to have mobile capabilities, so its cost of realization is large. Furthermore, this technique tends to cause a cascaded motion of nearby sensor nodes resulting in improved overhead and expanding the scope of recovery across the network [3]. The other technique is to redeploy additional relay nodes, which have no unique requirements on current sensor nodes. It is an efficient way to restore connectivity to the network. Here, recovery of network connectivity can be achieved by redeploying additional relay nodes here.

In this article, we discuss the issue of recovering partition connectivity with relay node deployment in wireless sensor networks. Our works' primary contributions are two. (1) To discover partitions in the network, we suggest a graph theory-based disjoint partition detection algorithm. (2) A heuristic partition retrieval algorithm is presented by redeploying a tiny amount of expensive but stronger relay nodes that can be illustrated in Fig. 1. These relay nodes can build a backbone communication network based on Steiner tree theory to prolong the lifetime of the network while maintaining network connectivity. Steiner tree's shortest data transmission is low latency and energy consumption.

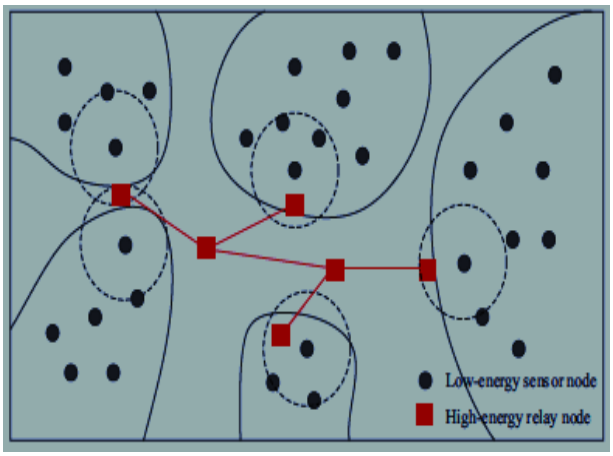


Figure 1. Partition connectivity recovery based on relay node deployment

This paper remaining is organized as follows. Section 2 reviews the work involved. A partition connectivity recovery algorithm is proposed in Section 3 based on the deployment of relay node for WSNs. Performance assessments are given in Section 4 to testify to the performance of the proposed solutions and we conclude this paper in Section 5.

International Innovative Research Journal of Engineering and Technology
ISSN: 2456-1983 Vol: 5, Issue No: 1, September 2019

II. RELATED WORK

In the context of WSNs, extensive studies have been conducted on the partition recovery problem; this problem can be divided into two sub-categories: relocation of movable sensor nodes and redeployment of additional relay nodes.

2.1 Relocating Movable Sensor Nodes

Akkaya et al.[4] suggested a PADRA detect potential partitions and to restore network connectivity by controlling movable node relocation. Imran et al.[5] introduced a distributed detection of partitioning and restore connectivity algorithm to tolerate actor failure. DCR proactively recognized and designated suitable, preferably non-critical, backup nodes that were critical to network connectivity based on local topological data. Abbasia et al.[6] proposed a distributed topology repair algorithm for the least movement, which sought to relocate the least number of nodes and reduce the distance and complexity of the message traveled.

2.2 Redeploying Extra Relay Nodes

Most research achieves the restoration of network connectivity by deploying additional relay nodes. To extend the life of the network while maintaining network connectivity, Lloyd et al.[7] deployed the minimum number of relay nodes to reach communication with other sensors or relay nodes. The strategy proposed in [8] has chosen to restore connectivity using the least number of relays while maintaining a certain quality in the topology created. Unlike existing schemes that formed a minimum spanning tree between the isolated segments, the proposed approach established a topology resembling a spider web for which the segments were located at the perimeter. By populating the smallest number of relay nodes, Lee et al.[9] presented an approach for federating segments in the network. The problem of optimization was then mapped to find the least cost paths based on the cell that collectively met the requirements of QoS. The heuristics proposed by Lee et al.[10] was QoS-aware relay node placement using the minimum Steiner tree on the convex hull. Chen et al. [2] proved that finding the minimum relay nodes was NP-hard and therefore preferred heuristics methods.

III. PARTITION CONNECTIVITY RECOVERY BASED ON RELAY NODE DEPLOYMENT

3.1 Problem Statement

This article focuses on the issue of relay node implementation in the wireless sensor network involving partition connectivity recovery. The WSN contains two types of nodes: (1) low-energy sensor nodes initially deployed to collect and transmit sensing information; and (2) re-deployed high-energy relay nodes are responsible for connecting disjoint partitions to recover linked network topology and transmitting information.

3.2 Disjoint Partition Detection

The main problem to solve is how to identify disjoint partitions in the WSN. Assume a WSN can be modeled as a G(V, E), where V is the set of nodes, and E is the set of edges. The edge $(v_1,v_2)\in E$ if $d(v_1,v_2) \leq 2R$, where R indicates the communicating radius of low-energy sensor nodes, for a set of nodes $v_1,v_2\in V$.Then we describe as follows the partition. Given a subsection $G_1(V_1, E_1)\in G$, if we call G_1 as a partition for any node $v_i\in V_1$, $v_j\in V-V_1$, $d(v_i,v_j)>2R$. The number of partitions in a network reflects some extent the network's connectivity performance.

To find all the partitions in the WSN, we take the Find Partition algorithm (see Algorithm 1). Assume the adjacency list of the network graph G is represented. It helps to detect communication holes by partitioning a WSN into several separate partitions. And therefore, by redeploying a small number of high-energy relay nodes to restore network connectivity, we eliminate these communication holes.

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

The sensor nodes in the same partition can interact with each other after finding various isolated partitions in the WSN. First, we discuss the implementation of the intra-partition relay node. Assume that a relay node's communicating radius is represented as R_r, where $R_r > R$ is present. To forward sensing data from sensor nodes to sink nodes, a high-energy relay node deployed in a partition may be used. For low-energy sensor nodes inside the partition, if they can forward their sensing information to certain relay nodes, this relay node must be in a sensor node (R) communication radius, as shown in Fig. 1.

International Innovative Research Journal of Engineering and Technology
ISSN: 2456-1983 Vol: 5, Issue No: 1, September 2019

One of the simplest methods of deploying relay nodes intra-partition is to find the partition center and place relay nodes at this location. However, there won't be the least number of relay nodes re-deployed in this deployment method. There is no doubt that the cost of deploying relay nodes will increase. This paper, therefore, designs an intra-partition relay node deployment scheme for each partition based on the convex hull, which is the most omnipresent structure in computational geometry. We can always discover a convex hull containing all the low-energy sensor nodes in a partition, as shown in Fig. 2. A relay node will be put at the place where two requirements must be satisfied:(1) the distance to certain sensor nodes on the convex hull shall be no more than R to ensure the exchange of data between the intra-partition sensor nodes and the relay node; and (2) the distance to the middle of the region shall be the least to ensure a least number of relay nodes to be used.

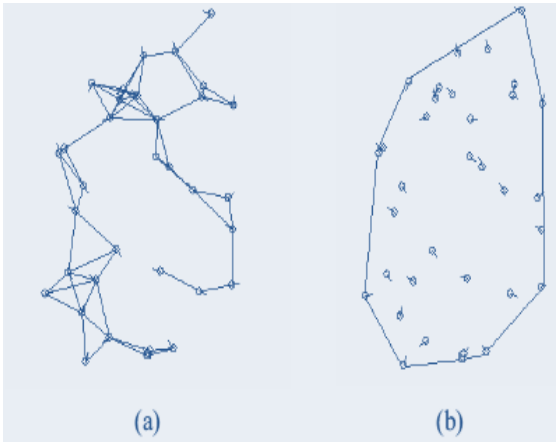


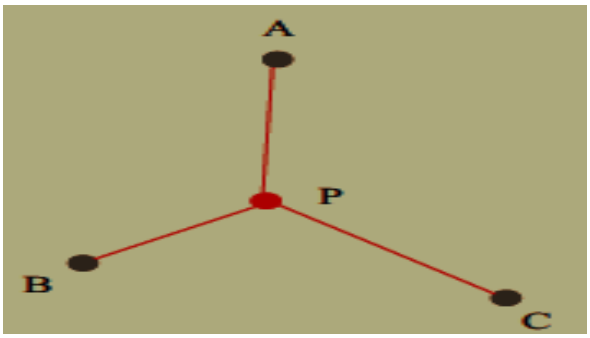
Figure 2.The convex hull of a partition in the network

3.4 Inter-partition Relay Node Deployment

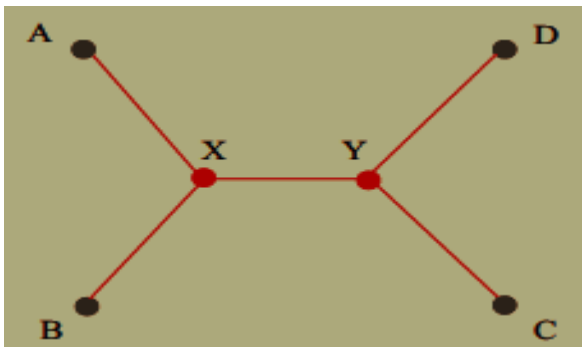
After all location data of intra-partition relay, nodes have been determined, inter-partition relay nodes will be implemented to link each disjoint partition together to accomplish the connectivity of the entire network communication. A Steiner tree will be constructed based on intra-partition relay nodes to ensure the shortest communication route of the entire high-energy communication network, considering the transmission delay and energy consumption requirements.

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

In the figure. 3(a), we demonstrate an example of the three-point Steiner tree, A, B and C, where an additional Steiner point P is added. We use the concept of the four-point Steiner tree in our work. Two additional Steiner points X and Y are added in this case. Fig. 3(b) is an example of a four-point Steiner tree, A, B, C, and D.



(a) A Steiner tree of three points



(b) A Steiner tree of four points

Figure 3. An example of a Steiner tree

The essence of the deployment algorithm for the inter-partition relay node is to locate the minimum tree for Steiner. Solving a minimum Steiner tree has been proven to be an NP-hard issue. Thus, heuristic techniques are preferred when the node scale is large. The Steiner tree minimum finding process contains three core issues:

(1) The appropriate quadrilaterals or triangles are selected for connecting the disjoint partitions (see literature [2]);

(2) Their Steiner nodes are discovered;

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

Assume Dis(Rei, S) is the distance between Rei and a Steiner node S intra-partition relay node. It can be defined as Ceiling (Dis(Rei, S)/Rr-1) the calculation technique of the number of relay nodes to be deployed along a Steiner tree edge.

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

International Innovative Research Journal of Engineering and Technology
ISSN: 2456-1983 Vol: 5, Issue No: 1, September 2019

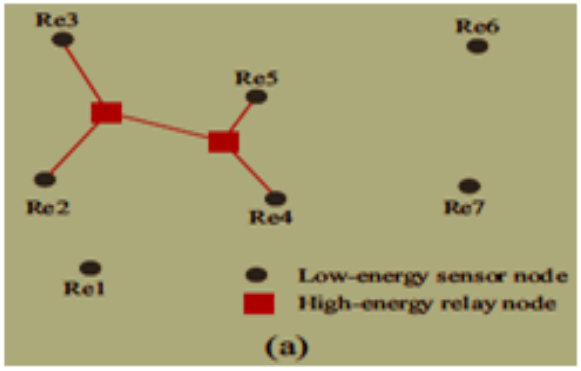
Algorithm 2. *Inter-partition relay node deployment*

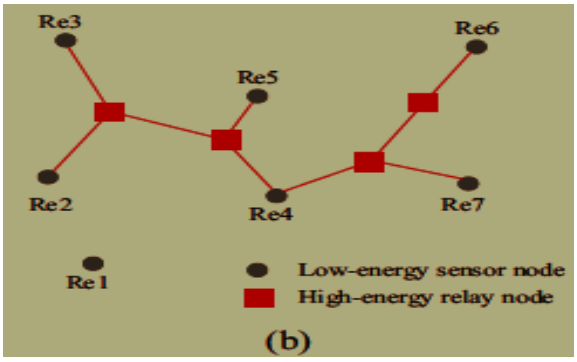
Step 1: Enumerate all the combinations of non-degenerate convex quadrilaterals and store them in list *Q*, and sort *Q* by the perimeter in ascending order. For each non-degenerate convex quadrilateral *q* in *Q*, if the number of disjoint partitions that the vertexes of *q* represent is larger than 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 Figure. 4, we illustrate a minimum of Steiner Tree generation. The quadrilateral minimum non-degenerate convex is composed of Re2, Re3, Re4, and Re5 relay nodes. Two quadrilaterals Steiner nodes are found. Several relay nodes are then placed on the edges of the Steiner tree to connect four partitions in the network at the appropriate positions (see Fig. 4(a)). If there are three angles of a triangle below 120°, the Steiner point can be found inside the triangle. Otherwise, if one angle of a triangle is greater than 120°, it is a degeneration triangle. Re1, Re6, and Re7 constitute a degenerate triangle. Consequently, we consider the formation of a non-degenerate triangle consisting of Re4, Re6, and Re7.This triangle's Steiner node is discovered. Extra relay nodes are then implemented to connect three partitions in the network at the edges of the Steiner tree (see Fig. 4(b)).The current number of partitions is 2 after the above operation. We discover the shortest edge in this situation to connect these two partitions to a linked network (see Fig. 4(c)).





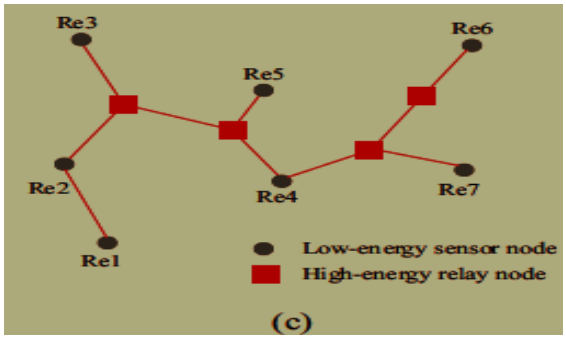


Figure 4.Generation of minimum Steiner tree

4. Performance Evaluation

In this section, through extensive simulations, we evaluate the performance of the proposed solutions. Sensor nodes are implemented in a region with a size of 1500m*1500m without the loss of generality. We investigate two main parameters: the radius (R), the number of partitions (N_p), the number of sensor nodes (N) and the number of relay nodes (N_r) to assess their impacts. Each outcome shown here is the 20 simulations statistical average.

Different from other works, high-energy relay nodes re-deployed in the suggested algorithm are used for data transmission in each partition. The number of inter-partition relay nodes deployed in our quadrilateral Steiner tree-based algorithm and Minimum Spanning Tree (MST) algorithm is only compared here.

4.1 Effect of Communicating Radius

Fig. 5 provides an impact on the number of relay nodes in the communication radius of the sensor nodes. In this example, the number of N_p partitions is 9. When certain parameters (e.g. area size, number of sensor nodes) remain unchanged, the length of the Steiner tree's link route is relatively set. From the data, as R increases from 50m to 200m, N_r will gradually reduce the connectivity required to achieve network communication.

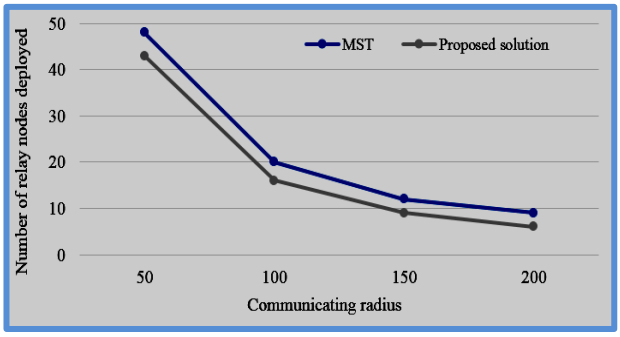


Figure 5. Effect of the communicating radius of sensor nodes

For example, in our proposed solution and MST algorithm, when R=50m, the number of relay nodes required N_r is 43 and 48. The number of relay nodes needed in the MST algorithm is lower than that with the rise of the communicating radius. In our proposed algorithm and MST algorithm, when R=200 m, N_r is 6 and 9 respectively. This is mainly because the increase in the communicating radius will lead to a decrease in the connection path length.

4.2 Effect of the Number of Partitions

Fig. 6 shows the effect of the number of partitions on the number of relay nodes (R=100 m). There is a directly proportional connection with the rise of N_p, Nr needed in our proposed algorithm and MST algorithm. The more N_p becomes, the more N_r will become necessary to restore connectivity to communication. For example, when N_p=4, Nr required is 13 and 15 in our proposed algorithm and MST algorithm. This is because the more N_p is, the more it becomes the number of partitions that can be linked by the quadrilateral Steiner tree.

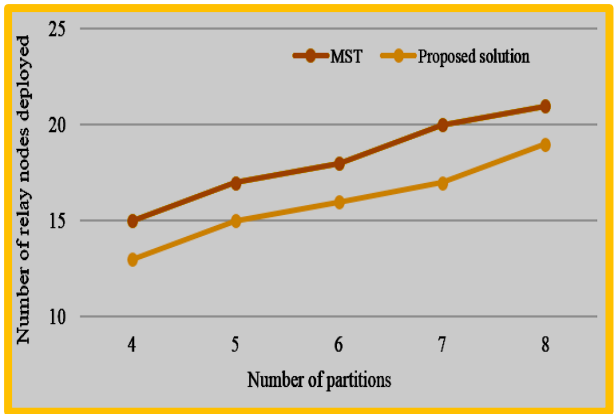


Figure 6. Effect of the number of partitions

4.3 Effect of the Number of Sensor Nodes

Fig. 7 displays the impact of the number of sensor nodes on the number of relay nodes. R=100m in this instance. With the rise of N, we can get that; the necessary N_r will gradually reduce. For instance, in our proposed solution and MST algorithm, when N=75, N_r needed is 12 and 15 respectively. If the region's size, the communicating radius remains unchanged, the larger number of sensor nodes will alleviate fewer partitions, thus requiring fewer relay nodes.

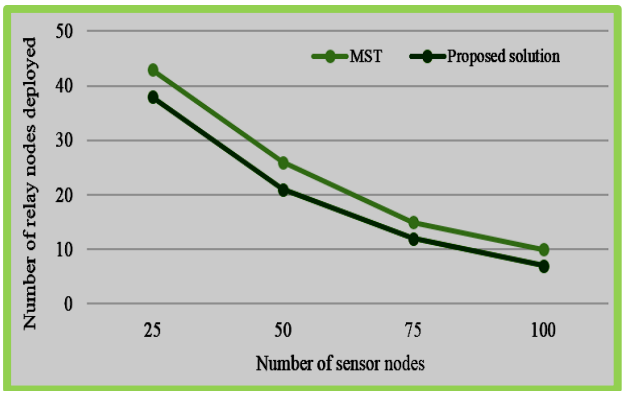


Figure 7. Effect of the number of sensor nodes

IV. RESULT & CONCLUSION

In conclusion, experimental results indicate that our proposed solution can accomplish partition connectivity recovery for WSNs with fewer relay nodes and lower network communication energy consumption relative to the MST algorithm. We propose a relay node deployment partition-based connectivity retrieval solution for wireless sensor networks. Deployment solutions for the intra-partition and inter-partition relay node are intended. Extensive simulation is performed to confirm the efficacy of our proposed solution and the impacts of some main parameters are discussed in detail.

REFERENCES

[1] N.N. Qin, D. Wu, Y.H. Yu, Connectivity recovery algorithm in a 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.

