

## A novel anchor-free position algorithm based on cluster technique

Xingfu Wang, Renzhi Cao, Qiuyuan Huang, Zineng Ke, Zhiqiang Liu

School of Computer Science and Technology, University of Science and Technology of China  
Hefei 230027, China

wangxfu@ustc.edu.cn, {ahjxcrz, IDfree}@mail.ustc.edu.cn, kzn@ustc.edu.cn, lzqdename@mail.ustc.edu.cn

**Abstract**—The WSN has a large spectrum of applications, and most of the applications are associated with position information. Considering the hostile environment of the sensors, it is very important to acquire the node's position without GPS. This paper proposes one novel anchor-free algorithm based on cluster technique, node's energy is considered, connective degree and the geometric limit principles of triangle inequality to heuristically build clusters, then fuse the clusters into one cluster. This algorithm effectively solves the position problem without any anchor node, conquer the cumulative error problem in traditional anchor-free algorithm, improve the localization precision and save the sensors energy. Simulation results show that after improving the fusion section, our algorithm can improve localization precision by 30% to 70% comparing to the traditional algorithm ABC.

**Keywords**—Localization; Wireless Sensor Network; Cluster; fusion

### I. INTRODUCTION AND RELATED WORK

The Wireless Sensor Network (WSN) is composed of large quantity of sensor nodes, nodes can communicate with each other by wireless signal. In September 1999[1], Business Week heralded it as one of the 21 most important technologies for the 21st century. Nowadays, WSN plays an important role in many domains, including environmental monitoring, target tracking, wireless routing, generating new capabilities for reconnaissance and surveillance as well as other tactical applications [2]. The position information is the basic of all of these applications.

There are two kinds of nodes in WSN. One holds the GPS localization system, we call them anchors. The other without GPS localization system calls normal nodes. At present, we always classify localization algorithms into two categories [3], one is the range based algorithms, which need either node-to-node distances or angles for estimating localizations, it involves many techniques such as time of arrival (TOA)[4], which computes the position through the signal propagation time difference, time difference of arrival (TDOA)[5], which computes the position through arriving time difference of two different waves, received signal strength indication (RSSI)[6], which estimates the position through the signal strength, and angle of arrival(AOA)[7], which needs the information of antenna angle to compute the position. Another is the range free algorithms, in which the position of the nodes is acquired

from other characters of the communication information, such as the number of jumps between two nodes.

In the practical applications, it's very common that all the sensor nodes are deployed indoor or can not receive the signal of the satellites because of the bad circumstances. The anchor-free position algorithm, which will compute the position of the nodes in the network not by GPS, but by communicating with each other to get the relative position of the nodes, is very important and efficient for this case.

The representational anchor-free position algorithms at present are ABC algorithm[8] and AFL algorithm[9]. The former is an increment algorithm. It makes use of the positioned nodes to locate the not positioned nodes. This algorithm is simple and easy to apply, but the demerit is its low position precision because of the cumulative effect on estimation errors, especially for the nodes far away from the origin node. The other algorithm is a parallel algorithm, it will find out one sensor's topology graph which is relatively similar to the practical case, and then introduce the concept of action power to revise the coordinate. The precise is high, but the algorithm is too complex and needs more computation and time. Camillo Gentile[10] improves the position precise by introducing the geometric limited principle of triangle inequality to adjust the border length of the triangle. Hady S. AbdelSalam et al.[11] proposes an algorithm to divide the network into several hexagon regions and select the node which is closest to the centre of the hexagon to be the anchor node. Antonio Abramo et al.[12] introduces the concept of formal node and informal node and find out the nodes which have lower possibility to get errors to improve the position precision. Radu Stoleru et al.[13] proposes an algorithm from the probability aspect to compute each node's position. Ameer Ahmed Abbasi et al.[14] gives a conclusion of the cluster algorithms. Hongyi Wu et al.[15] proposes a local position algorithm which chooses one proper landmark to compute each node's coordinate. Adel Youssef et al.[16] proposes the gateway node which has higher energy to build clusters and build a spanning tree for localization. In this paper, we consider the case in two-dimensional, and it also makes sense to explore this position algorithm in three-dimensional. The rest of the paper is recognized into five parts. The first section is the algorithm of forming cluster, the second section is the localization algorithms within each cluster, the third section is the fusion algorithm between clusters, then the next section is the simulation and the discussion, the last section concludes the paper.

II. ALGORITHM OF FORMING CLUSTER

A. Preparing Knowledge .

One character about a vertex inside a polygon is that the summation of all angles formed by one vertex with other vertices of the polygon is  $2\pi$ . In this paper we will make use of this character to find the outside nodes which are not joining any clusters, and select the cluster head from the outside nodes, form clusters from the outside of the network, continue this process. Each cluster will be formed from the outside of the not positioned network, so finally all the nodes will join clusters.

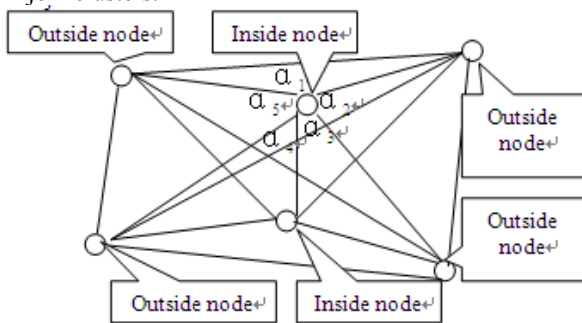


Figure 1. Picture of the outside nodes and inside nodes.

From Figure 1, we know:  $\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_n = 2\pi$ . But the summation of angles of outside nodes is less than  $2\pi$ .

Each node will firstly collect all the distance information of its neighbors, and then compute the angle according to the cosine theorem. We assume the three borders are a, b and c, the vertex angle is  $\alpha$ . The triangle test method is as follows:

**The triangle test method** :Selecting two smallest border from a, b and c, and then judge whether the summation of the two border length is larger than the largest border length, if so, we say the triangle has passed the triangle test, otherwise, we say the triangle has not passed the triangle test. We have the following theorem:

**Theorem:** If one triangle has passed the triangle test, any summation of two borders length will be larger than the other border length, and any difference of two borders length will be smaller than the other border length. Proof:

We assume the three border length is a, b and c, to simplify the proof, we assume  $0 < b < a, 0 < c < a$ , the proof of the other case is very similar to our assumption. In this case, if  $b + c > a$ , then we can conclude the following:  $a - b < c; a - c < b; -a < b - c < a; a + b > c + b > c; a + c > b + c > b$ .

The proof of the conclusion is accomplished.

We can evaluate the ranging error of the node by this triangle test method, and also discover the abnormal case. The more number of times that the node does not pass the triangle test, the more possibility that the node is an abnormal node.

B. Inside Node Algorithm.

We assume the node already has all the distance information in its communication range. The Inside Node Algorithm can judge whether the node is an inside node or

an outside node. We call inside node which is not join any cluster and is inside the topology graphics formed by the sensor nodes. The outside nodes are the nodes at the brink of the sensor network. We assume the node has the number of N sensor nodes in its communication range, they are  $1, 2, \dots, N$ . The Inside Node Algorithm is as follows:

- 1) The node will get N triangles as their vertex according to the distance information. We can acquire all the border length of each triangle, and then we can memorize the number of times that is not passing the triangle test according to the method in section A.
- 2) We assume that the vertex angle based on the node is  $\alpha_i$  to each triangle. The border length of the triangle is A,B,C, we can compute the angle  $\alpha_i$  from the cosine theorem :
 
$$\cos(\alpha_i) = \frac{B^2 + C^2 - A^2}{2 * B * C}$$
- 3) After getting all the angles  $\alpha_1, \alpha_2 \dots \alpha_N$ , we will do a summation to all the angles, and then judge whether the result is  $2\pi$ . Considering the ranging error, the border length we get will have some errors, so we will judge whether the result is large than  $\frac{3}{2}\pi$ , if so, we consider the node for inside node, otherwise we consider the node for outside node.

We will consider the number of times that the node does the triangle test and passes the triangle test, and also the energy of each node to get an evaluation function as follows:

$$F(Head) = c_1 * Q + c_2 * \frac{D}{15} + c_3 * \frac{D - T}{D} \quad \dots (1)$$

In formula (1),  $c_1, c_2$  and  $c_3$  represent a weight value, Q is the radio of the node remainder energy  $Q_i$  and the maximum energy  $Q_{max}$ , D represents the number of triangle test times, that is also the connection degree, if D is larger than 15, we will set D to be 15, so the maximum of D is 15. T represents the number of times not passing the triangle test. In this formula, the larger is  $c_1$ , the more important is the energy, and similar to  $c_2$  and  $c_3$ , they represent the importance of connection degree and triangle test. Despite different value for each parameter, the summation of  $c_1, c_2$  and  $c_3$  is 1.

C. The formation of the cluster.

We propose one algorithm to select the cluster head, and then build a cluster based on this cluster head. The cluster's formation algorithm is as follows:

- 1) Nodes communicate with other neighbor nodes to get the distance information.
- 2) Using the inside node algorithm proposed in the last section to decide whether it is an inside node or not.
- 3) The node computes its evaluation value  $F_i(Head)$  by the use of evaluation function in formula (1).

- 4) If the node is an outside node, it will send one apply message to its neighbor to apply for being a cluster head. The node with the largest evaluation value will be the cluster head, and the cluster head will send one request message to its neighbor to invite them to join the cluster. If the node is an inside node, it will not send any message. When the node receive a request message, it will join the cluster, if the node doesn't receive a request message for a long time, it will delete the node which joins other clusters from its neighbor list, and then go to step 2.
- 5) The cluster head collects all the nodes' information in its communication range to form one cluster.

### III. LOCALIZATION ALGORITHM WITHIN THE CLUSTER

In the last chapter, we have selected the cluster head and formed the basic cluster, in this chapter, we will build up one local coordinate system in the basic cluster, and then we will compute the local coordinate of the nodes. According to vary methods of selecting the coordinate origin, we propose two different local position algorithms, and compare their results in the simulation part. We assume the node O is selected as the coordinate origin, it's coordinate is (0,0), then we select another node A in the same cluster, the distance between them is  $D_1$ , we build up the coordinate system according to the radial OA, the coordinate of A is (0, $D_1$ ), we assume another node B is in the first quadrant, the distance between B and O is  $D_2$ , the distance between B and A is  $D_3$ .As the following picture illustrated:

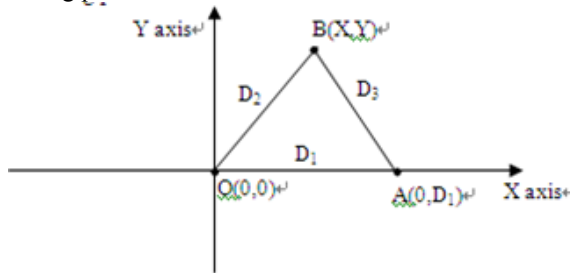


Figure 2. The local coordinate system.

The coordinate of B is (x, y), and then we can infer from the cosine theorem that

$$\cos(\angle BOA) = \frac{X}{D_2} = \frac{D_1^2 + D_2^2 - D_3^2}{2 * D_1 * D_2}$$

And then we can easily get the coordinate:

$$\begin{cases} X = \frac{D_1^2 + D_2^2 - D_3^2}{2 * D_1} \\ Y = \sqrt{D_2^2 - X^2} \end{cases} \dots (2)$$

According to the selection method of node O and node A, we propose two local position algorithms:

#### A. Not Heuristic Local Position Algorithm(NHLPA)

- 1) Selecting the cluster head as the coordinate origin O.

- 2) Selecting the node that is farthest from the cluster head as the node A.
- 3) The position of B can be computed by formula (2).

#### B. Heuristic Local Position Algorithm(HLPA)

- 1) According to the formula (1), the cluster head will select the node with the largest evaluation value as the coordinate origin O.
- 2) Selecting the node that is farthest from the cluster head as the node A.
- 3) The position of B can be computed by formula (2).

We propose one evaluation function  $F(x, y)$  to evaluate the performance of the positioned basic cluster, the evaluation function is as follows:

$$F(x, y) = \sum_{i=1}^{i=n} (\sqrt{(x - x_i)^2 + (y - y_i)^2} - d_i)$$

(x, y) is the coordinate of node C, which is any node except the coordinate origin O, and  $(x_i, y_i)$  is the coordinate of any other node except node O, node A and node C,  $d_i$  is the distance between node C and node i. We can evaluate the positioning accuracy of the basic cluster through this evaluation function. The lower evaluation value illuminates the higher positioning accuracy.

### IV. THE FUSION ALGORITHM BETWEEN CLUSTERS

In this chapter, since we have divided the whole sensor network into several clusters, we will fuse these clusters into one large uniform coordinate system.

If two clusters are overlapping, we call them overlapping clusters. If there are more than three corporate nodes belong to two overlapping clusters, we can fuse the two clusters to get one large cluster. It is important to decide which cluster joins another cluster. We propose one improvement method in the fusion part. Unlike randomly choose one cluster joins another one, we use the evaluation function  $F(x, y)$  in the last chapter to evaluate the corporate nodes belong to the overlapping clusters. The corporate nodes have different evaluation values in different clusters. We find out the cluster in which the corporate nodes have lower evaluation values, and then make the other cluster join this cluster. We will compare the result between the improved position algorithms and not improved position algorithms in the simulation part.

We consider the two-dimensional situation in the fusion part, but it is easy to expand it to three-dimensional. When in two-dimensional, we must have at least three nodes in both of the two clusters. We assume there are three nodes A, B and C, they all belong to cluster one and cluster two. the coordinate in cluster one is  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ , the coordinate in cluster two is  $(x_1', y_1'), (x_2', y_2'), (x_3', y_3')$ , we assume T is a 2 \* 2 matrix, B is a 1 \* 2 matrix, and  $(x', y') = (x, y) * T + B$ , we can easily conclude :

$$\begin{cases} (x_1', y_1') = (x_1, y_1) * T + B \\ (x_2', y_2') = (x_2, y_2) * T + B \\ (x_3', y_3') = (x_3, y_3) * T + B \end{cases}$$

Solve this equation, we can get the result:

$$T = \begin{pmatrix} x_1 - x_2, y_1 - y_2 \\ x_1 - x_3, y_1 - y_3 \end{pmatrix}^{-1} \begin{pmatrix} x_1' - x_2', y_1' - y_2' \\ x_1' - x_3', y_1' - y_3' \end{pmatrix} \dots\dots (3)$$

$$B = (x_1, y_1) * T - (x_1', y_1')$$

So any node  $(x, y)$  in cluster one can compute the coordinate in cluster two by  $(x', y') = (x, y) * T + B$ , the T and B can be computed by formula (3). The fusion part can be carried out in a parallel way, so we can finally get one large uniform position system through fusion.

### V. SIMULATION

In the simulation experiments, we randomly deploy 100 nodes in a 100\*100 m2 target field on the OMNET++ platform. We assume the maximum transmission range of all nodes is 20m. We assume the ranging error obeys the normal distribution. According to whether adopt the improvement method in the fusion part, we will get the following four algorithms:

- 1) Improved Not Heuristic Local Position Algorithm (INHLPA).
- 2) Not Improved Not Heuristic Local Position Algorithm (NINHLPA)
- 3) Improved Heuristic Local Position Algorithm (IHLPA)
- 4) Not Improved Heuristic Local Position Algorithm (NIHLPA)

In this section, we will give the comparison of effect of different parameters and compare the performance of the four algorithms with ABC algorithm and then analyze the results.

#### A. Comparing the influence of different parameter.

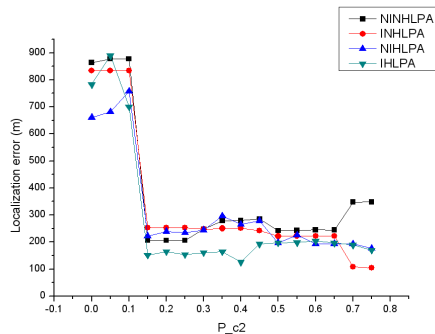


Figure 3. Localization error with different c2.

Fig 3 shows the comparison of localization error using different parameter in each algorithm. We assume c1 equals to 0.25, which represents the importance of the energy of the node. In this simulation we do not change the value of c1.

We can see from this figure that when c2 is in the range of [0.0, 0.75], c2 + c3 = 0.75, the localization error of IHLPA is the smallest. We can easily conclude from the simulation that when c2 is very close to 0, that means we seldom consider the node's connection degree, we will find out the localization error is very large. When c2 is in the range of 0.2 and 0.6, localization error of the four algorithms is relatively very small. When c2 is close to 0.75, that means we seldom consider the abnormal nodes, the improved algorithms get a better performance. The NINHLPA algorithm gets the worst performance in this case. Considering the localization error of the four algorithms, we choose c1 equals to 0.25, c2 equals to 0.4, c3 equals to 0.35 to get the best performance of all algorithms.

#### B. Comparing the influence of the heuristic method.

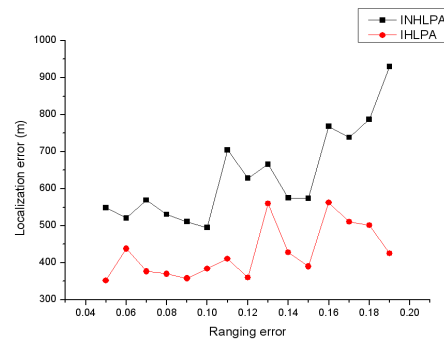


Figure 4. Localization error of IHLPA and INHLPA

Figure 4 demonstrates that when we adopt the heuristic method, the localization error will be largely decreased. The two algorithms are all improved in the fusion part, but the IHLPA adopts the heuristic method to form the local coordinate system. The localization error of algorithms adopt the heuristic method will improve about 30% comparing with algorithms not adopt the heuristic method.

#### C. Comparing the influence of improving the fusion part.

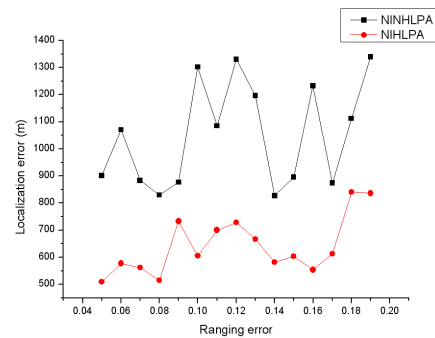


Figure 5. Localization error of NINHLPA and NIHLPA

Figure 5 illuminates that the localization algorithms revise the position result in the fusion section will have a better performance. We can see with the augment of ranging error, the localization error of both NINHLPA and NIHLPA increase, but the NIHLPA improves about 40% against NINHLPA.

#### D. Comparing the influence of different ranging error.

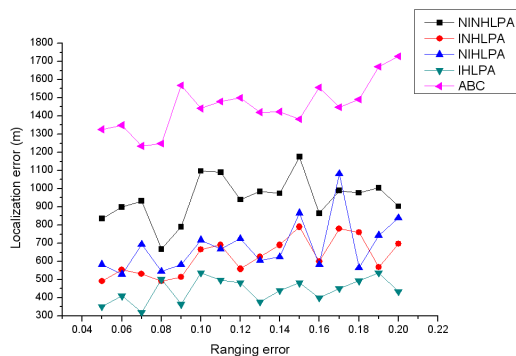


Figure 6. Localization error of all algorithms with the augment of ranging error

We compare the four algorithms with the traditional position algorithm ABC, and we can easily conclude from the simulation that our algorithms acquire a better performance in different ranging error. The IHLPA acquires the best performance, and improves the localization error about 70% against the ABC algorithm. The NINHLPA also improves about 30% against the ABC algorithm.

#### VI. CONCLUSION

This paper proposes an anchor-free algorithm based on the cluster technique, unlike anchor-based algorithm, all the nodes in the network are common nodes. We firstly divide the network into several clusters, and then form a local coordinate system in each cluster, and finally we propose a fusion algorithm to fuse all of the clusters into a uniform coordinate system. The algorithm we have proposed is a parallel algorithm, it does not have the error cumulative problem comparing with the traditional localization algorithm, and it can also effectively solve the position problem that all the sensors in the network are common nodes. Considering whether adopt the heuristic method and the improvement in fusion section, we propose four algorithms: INHLPA, NINHLPA, IHLPA, NIHLPA. Through the simulation results, we find different parameter will affect the location accuracy, and then we determine the appropriate range of the parameter's value. By comparing the traditional ABC localization algorithm with INHLPA, NINHLPA, IHLPA, NIHLPA in different ranging error, we find IHLPA has the highest localization precision, it reduces 70% average location error in our experiment platform. The future work is to expand our algorithm into three-dimensional, do some research on the fusion section and the local position section to improve the localization error, prove the value of the parameter in our algorithm which can get the highest localization precision, then expand this algorithm to the WSN secure domain<sup>[17][18]</sup>.

#### ACKNOWLEDGMENT

Supported by the National Natural Science Foundation of China under Grant No. 60773037, 60970128 and major national science and technology projects under Grant No. 2008ZX10005-013.

#### REFERENCES

- [1] "21 ideas for the 21st century", Business Week, pp.78-167, Aug 30, 1999.
- [2] Haiqing Jiang, Renzhi Cao, Xingfu Wang "Apply Modified Method of Nonlinear Optimization to Improve Localization accuracy in WSN", The 5th International Conference on Wireless Communications, Networking and Mobile Computing, ISBN: 978-1-4244-3693-4, 2009.
- [3] TA Alhmiedat, SH Yang "A Survey: Localization and Tracking Mobile Targets through Wireless Sensors Network" PGnet, ISBN 1-9025-6016-7 2007.
- [4] B. H. Wellenhoff, H. Lichtenegger and J. Collins, Global Positioning System: Theory and Practice: Springer Verlag, 1997.
- [5] A. Savvides, C. Han and M. B. Srivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in Proceedings of ACM MobiCom, 2001.
- [6] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," in Proceedings of IEEE INFOCOM, 2000.
- [7] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AOA," in Proceedings of IEEE INFOCOM, 2003.
- [8] C Savarese, JM Rabaey, J Beutel ICASSP IEEE INT CONF ACOUST SPEECH SIGNAL PROCESS PROC 4, 2037-2040, 2001.
- [9] Nissanka B. Priyantha, Hari Balakrishnan, Erik Demaine, and Seth Teller, Conference On Embedded Networked Sensor Systems. ISBN:1-58113-707-9 2003.
- [10] Camillo Gentile, Distributed Sensor Location through Linear Programming with Triangle Inequality Constraints. IEEE International Conference on Communications, v 9, p 4020-4027, 2006, ISSN: 05361486.
- [11] Hady S. AbdelSalam, Stephan Olariu. HexNet: Hexagon-Based Localization Technique For Wireless Sensor Networks. 7th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2009, March 9, 2009.
- [12] Antonio Abramo, Franco Blanchini, Luca Geretti, and Carlo Savorgnan. A Mixed Convex/Nonconvex Distributed Localization Approach for the Deployment of Indoor Positioning Services. IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 7, NO. 11, NOVEMBER 2008.
- [13] Radu Stoleru and John A. Stankovic. Probability Grid: A Location Estimation Scheme for Wireless Sensor Networks. 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON 2004, ISBN-10: 078038796.
- [14] Ameer Ahmed Abbasi, Mohamed Younis. A survey on clustering algorithms for wireless sensor networks. Computer Communications 30 (2007) 2826-2841.
- [15] Hongyi Wu, Member, IEEE, Chong Wang, Student Member, IEEE, and Nian-Feng Tzeng, Senior Member, IEEE. Novel Self-Configurable Positioning Technique for Multihop Wireless Networks. IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 13, NO. 3, JUNE 2005.
- [16] Adel Youssef and Ashok Agrawala, Mohamed Younis. Accurate Anchor-Free Node Localization in Wireless Sensor Networks. 24th IEEE International Performance, Computing, and Communications Conference, IPCCC 2005.
- [17] Xingfu Wang, Lei Qian, Haiqing Jiang. Tolerant majority-colluding attacks for secure localization in wireless sensor networks. Proceedings - 5th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2009. Accession number: 20100112609849.
- [18] Xingfu Wang, Dai Fuquan, Qian Lei; Dong Hao. A way to solve the threat of selfish and malicious nodes for Ad hoc networks. 2008

International Symposium on Information Science and Engineering,  
ISISE 2008, v 1, p 368-370.