TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.4, April 2014, pp. 2735 ~ 2742
DOI: http://dx.doi.org/10.11591/telkomnika.v12i4.4304

Optimization Research of the DV-Hop Localization Algorithm

Wang Guo*, Juan Wei
Henan Mechanical and Electrical Engineering College, Xinxiang City, Henan Province, China 453003
*Corresponding author, e-mail: guo_wang0102@163.com

Abstract

The traditional DV-Hop sensor location algorithm estimates the distance between unknown nodes and anchor nodes by hop distance instead of straight-line distance. In the actual location condition, unknown nodes and anchor nodes mostly connect to each other through broken line. When the deviation of the estimated value and the actual value of average distance per hop is large, the error of estimated and actual distance between unknown node and the anchor node increases. This paper analyzes the compositions of DV-Hop positioning algorithm and the factors easily causing errors, proposes the DV-Hop positioning algorithm improved by mobile beacon dynamically selection. This algorithm is based on the basic ideology of the DV-Hop positioning algorithm in which a moving virtual node moves along the pre-set route and broadcasts its position information to form multiple virtual beacons. The unknown nodes can compute the average hop distance by weight processing and the distances among each virtual beacon. Finally, the position of the unknown node can be computed and the accurate positioning can be finished. Finally, the simulation experiment testifies the improved DV-Hop positioning algorithm to decrease cost and complexity and increase the positioning accuracy and efficiency.

Keywords*: wireless sensor network, mobile beacon, dv-hop, positioning system*

Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

The Wireless Sensor Network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure etc., and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity [1]. The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors [2].

The positioning algorithm based on beacon needs high beacon density which causes the cost increase. This paper introduces the mobile beacon which is moving in the network and broadcasting its own position grouping information in the network to construct virtual beacon to locate unknown nodes [3, 4]. It illustrates the mobile beacon dynamically selected DV-Hop positioning algorithm. This algorithm is based on the basic ideology of the DV-Hop positioning algorithm in which a moving virtual node moves along the pre-set route and broadcasts its position information to form multiple virtual beacons. The unknown nodes can compute the average hop distance by weight processing and the distances among each virtual beacon. Finally, the position of the unknown node can be computed and the accurate positioning can be finished [5]. This paper carries researches on dynamic selection algorithm and hopes to select the mobile beacon with maximum calculation accuracy to compute. The simulation proves the DV-Hop positioning algorithm improved by mobile beacon can lower the positioning cost and networking complexity and increase nodes positioning accuracy and efficiency compared with DV-Hop algorithm [6-8].

2. DV-Hop Positioning Algorithm

DV-Hop is a range-free location algorithm proposed by NICULESCUD etc [9]. The algorithm contains three procedures . First, all the unknown nodes in the network obtain the minimum hop count to the anchor node. Second, the average distance per hop obtained multiple by the minimum hop count to get the estimated distance. Third, when the unknown

Received August 17, 2013; Revised October 22, 2013; Accepted November 12, 2013

nodes got three or more estimated distance to the anchor node, the self locating can be achieved by trilateration or maximum likelihood estimation [10, 11].

Anchor nodes broadcast the packet of self-location information to the neighbor nodes including hop digital section which is initialized to 0. Nodes received the information record the minimum hop numbers to each anchor node and ignore the larger packet from same original anchor node. Then the digital section plus 1 and is sent to the neighbor nodes. By this way, all the nodes in the network could get the minimum hop count to each anchor node.

The average distance per hop is estimated by formula (1) for each anchor node according to the other anchor node coordinate values and the number of hops recorded in the first stage.

$$\text{Hopsize}_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} h_{ij}} \quad (1)$$

Wherein (xi,yi) and (xj,yj) are the coordinates of anchor nodes i and j, and hij is the hop count from i to j.

Then, the anchor node will broadcast the average distance per hop calculated to the network. The unknown nodes only record average per hop distance of their own recent anchor nodes, and then multiply the distance by the minimum hop count to estimate distance from the anchor node.

The basic principle of trilateration: in two-dimensional space, if we know the distance from one point to three anchor nodes whose coordinates are known, we can determine the coordinates of the point position. Assume that the three coordinates are (x1, y1), (x2, y2) and (x3, y3), the coordinate of the unknown node is (x, y) and the three distance are d1, d2, d3. Then we get:

$$\begin{aligned} d1 &= \sqrt{(x1-x)^2 + (y1-y)^2} \\ d2 &= \sqrt{(x2-x)^2 + (y2-y)^2} \\ d3 &= \sqrt{(x3-x)^2 + (y3-y)^2} \end{aligned} \quad (2)$$

Solving Equations (2) we can obtain the unknown node coordinates (x, y).

3. The DV-Hop Algorithm

DSB-DV-Hop algorithm, which is improved DV-Hop algorithm improved by mobile dynamic selection, can compute the average hop distance independently in each virtual node position by moving the beacon in the network along the pre-set orbit and broadcast the grouping positions. Each virtual beacon can record the hops of unknown nodes. The dynamic selection can compute the position of the beacon nodes with maximum accuracy to decrease the cost of positioning and increase the positioning accuracy.

3.1. Error Calculation

When applies DV-Hop positioning algorithm to compute, first the network floods the hops of the data package which contains the pre-set hop with the nodes' positions. Once the node receives a data package, it will add 1 to the hops in the time cycle. The nodes will stop forwarding the data package when it reaches the set hops.

In Figure 1, three data packages are picked from 5 beacon nodes forming a group to compute tri-edge positioning. The computation results will vary if the selected nodes are different which will cause obvious errors. We pick two combnaitons-A3, A4, A5 and A1, A2, A3. The figure 2 illustrates the computation scheme under the two combinations. The computable errors will obviously increase if select A4 and A5 to participate the tri-edge positioning computation. Therefore, the computation accuracy will be directly affected depends on how to select the optimal virtual beacon participating into positioning computation.

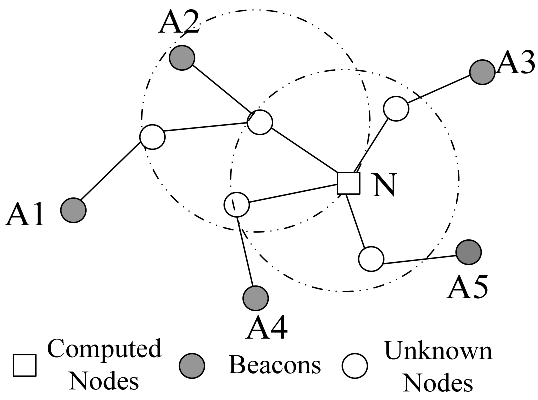
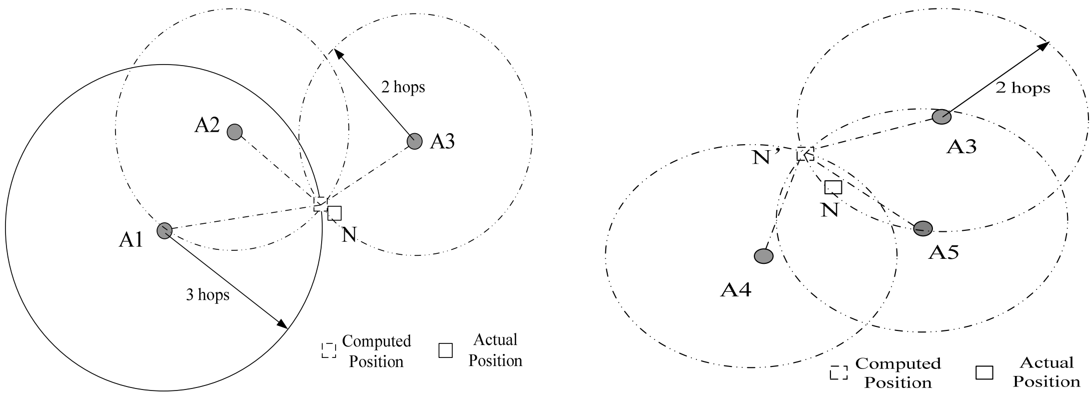


Figure 1. DV-Hop Positioning Algorithm Error Source Analysis



(a) Select A1, A2 and A3 to compute (b) Select A3, A4 and A5 to compute

Figure 2. The Scheme of Selecting Different Beacon Nodes

3.2. Error Correction

First, calculate cc the average hop distance of the whole network. Average hop distance error of the whole network is obtained by analyzing the deviation between the true distance and the calculated distance in the DV-Hop algorithm between every two anchor nodes, in order to reduce the deviation of average hop distance and the real average hop distance. The error per hop in the whole network is δ expressed as formula (3)

$$\delta = \frac{\sum_{i \neq j} \left| d_{true_{ij}} - d_{estimated_{ij}} \right|}{\sum_{i \neq j} hops_{ij}} \qquad (3)$$

$\left| d_{true_{ij}} - d_{estimated_{ij}} \right|$ is the absolute value of the differential real and estimated distance between every two anchor nodes i and j. $hops_{ij}$ represents the minimum hop count between i and j. The real and estimated distance can be calculated by formula (4).

$$d_{true_{ij}} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$
$$d_{estimated_{ij}} = cc * hops_{ij} \quad (4)$$

The real distance between every two of anchor nodes A, B,C are known. Then the average distance per hop of three anchor nodes can be obtained according to the locating procedure of DV-Hop:

$$C_A = \frac{40+75}{4+4} = 14.375$$
$$C_B = \frac{75+100}{4+6} = 17.5$$
$$C_C = \frac{40+100}{4+6} = 14 \quad (5)$$

The average distance per hop in the whole network:

$$cc = (C_A + C_B + C_C)/3 \approx 15.29 \quad (6)$$

The error of average distance δ :

$$\delta = \frac{|40 - 15.29 \times 4| + |75 - 15.29 \times 4| + |100 - 15.29 \times 6|}{4+4+6}$$
$$= 3.09 \quad (7)$$

3.3. Computation Procedure of the Mobile Beacon

The beacons are installed in mobile robotic and mobile platform. When the beacons move according to some rule, they can acquire their own positions and broadcast the information in some periods to help the unknown nodes get positioning. When the unknown can acquire enough virtual beacon information, the optimal virtual beacon dynamically selected participates into the computation to increase the accuracy. The details of the improved DV-Hop algorithm are described as follows.

3.3.1. Process of the Locating Algorithm

Beacon nodes periodically broadcast their location information in the network. Each node receives the hop count plus 1 and continues to broadcast. Each node will compare own value with a plurality of receiving value from neighbor nodes. If less than its number of hops, the broadcast packet is discarded.

The method contains the follows steps:

Step 1: The beacon at first position gets the current location, and package broadcast packets {LIDi, xi, yi, HOPs}. Wherein, LIDi is the current broadcast packet at ith position, xi, yi are the location coordinates and HOPs represents the hops experienced with 0 as its initial value.

Step 2: All neighbor nodes within the scope of the first position receive positioning information packet and compare it to own data packet to maintain the one with less hops. After that, hops count plus 1 and is recorded. Wait for the next information packet.

Step 3: Move to next position. The position information plus 1 and the new location information is broadcasted in the new scale. Receive packets sent back, take the packet with least hops comparing to the former position, and then broadcast the average distance per hop to the network.

Step 4: Repeat Step 3 until the position reach the threshold K set. Suppose N is the number of unknown nodes, the proportion of virtual beacon q can be calculated by $K = N \times q$.

Step 5: Unknown node select the average hop distance of the first virtual beacons received and use the dynamic selection algorithm to select the optimal virtual beacon nodes. According to the number of hops recorded, the distance to the selected virtual beacons is calculated, and then its own position is calculated by trilateration method.

3.3.2. The Scheme of Selecting Different Beacon Nodes

The detailed procedure of the algorithm is as follows.

It's supposed that the grouping with known node N has n virtual beacons. N determines A3 which firstly receives the virtual beacon as standard node. N randomly selects p virtual beacons from virtual positioning list except A3 to compute. If p=4, that's (A1,A2,A4,A5). The steps of the dynamic selection with 5 virtual beacons are as follows.

1st step: Select A1, A2, A4, A5 from virtual beacons and combine each two of them with A3 to construct tri-edge positioning computed nodes. The amount of combination is C42=6, they are (A3,A1,A2)、(A3,A1,A4)、(A3,A1,A5)、(A3,A2,A4)、(A3,A2,A5)、(A3,A4,A5). These combinations can be represented by N1、N2、N3、N4、N5 and N6 to compute the position of N by tri-edge measurement.

2nd step: take N1、N2、N3、N4、N5 and N6 as known nodes to replace the standard A3 in the 6 combinations to acquire another 6 combinations. They are (N1,A1,A2), (N2,A1,A4), (N3,A1,A5), (N4,A2,A4), (N5,A2,A5) and (N6,A4,A5). They are used to compute the positions of A3 respectively to attain 6 values- A31, A32, A33, A34, A35 and A36.

3rd step: Compare the position of standard A3 and the actual position of A3 to determine which combination in the first step is close to the actual position of A3. Suppose the coordinates (XA3,YA3) is the actual position A3, the computed coordinates (Xn,Yn) of A3 can be attained by the method. The error computation equation is:

$$d = \sqrt{\left(X_n - X_{A3}\right)^2 + \left(Y_n - Y_{A3}\right)^2}$$

4th step: select the one with minimum error. The combinations of the minimum d from equation 4 are to get the virtual beacon with the maximum accuracy.

After the steps of pervious dynamic selection, the unknown nodes will compute the tri-edge positioning algorithm according to the optimal three virtual beacon combinations to acquire the position with maximum accuracy.

4. Simulation

This paper applies OMNeT++ platform to simulate the improved mobile dynamic selection algorithm to testify the positioning accuracy. Additionally, MATLAB is used to analyze the data. In the OMNeT++ platform, there are randomly 100 sensor nodes in the area of 50□50. The communication radius of each node is 10. There is one mobile beacon which can acquire its own positions at any time which moves according to Gauss-Markov moving model.

4.1. Simulation Platform OMNeT++

OMNeT++ is a component-based, modular and open-architecture discrete event simulation framework. The most common use of OMNeT++ is for simulation of computer networks, but it is also used for queuing network simulation, and other areas as well. OMNeT++ is not a simulator in itself but rather a simulation framework. Instead of containing explicit and hardwired support for computer networks or other areas, it provides the infrastructure for writing such simulations. Specific application areas are catered by various simulation models and frameworks, most of them are open source. These models are developed completely independently of OMNeT++, and follow their own release cycles.

The model of OMNeT++ is composed by hierarchically nested modules as shown in Figure 3.

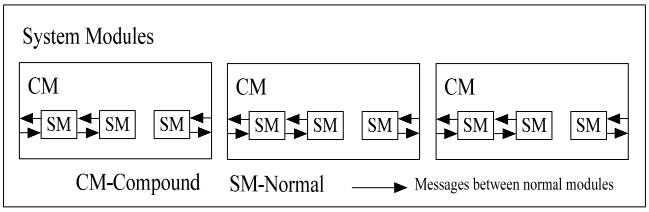


Figure 3. OMNeT ++ Grouping Nested Model

OMNe ++ model designs simulation topology with two different languages the NED and C++, and can define network, modules and link. C++ is responsible for the implementation of the NED language definition module, able to simulate any of the methods in the model in OMNeT++ simulation library, including the cancellation of the course of events and schedule, as well as Plove, a graphical tool used for statistics and generate data.

OMNe ++ simulation model has two operation modes: view and text. View simulation displays the process of the transmission of information between the two modules in form of animation. If it is a larger-scale simulation experiment, it can further show the specific information transfer between the various modules.

4.2. Simulation Experiment

Because the DV-Hop positioning algorithm itself isn't based on mobile beacon, the beacon is fixed in the network. The density and distribution of the beacons will affect the positioning errors. The positioning errors and coverage rates from 10 beacons to 20 beacons are analyzed with the same network model.

When the beacon density reaches 0.2 under the set conditions with DV-Hop positioning algorithm, the positioning error is about 23% and the positioning coverage rate is about 93%. When the density reaches 0.3, the positioning error and coverage rate will reach 21% and 97% respectively. When the density of the virtual beacon reaches 0.3, the positioning error and coverage rate will reach 14% and 98% respectively. The above data is attained by 20 independent simulation results. The density of the virtual beacons in the proposed improved positioning algorithm is more effective than set beacons which can decrease the network cost and positioning cost to increase the positioning efficiency.

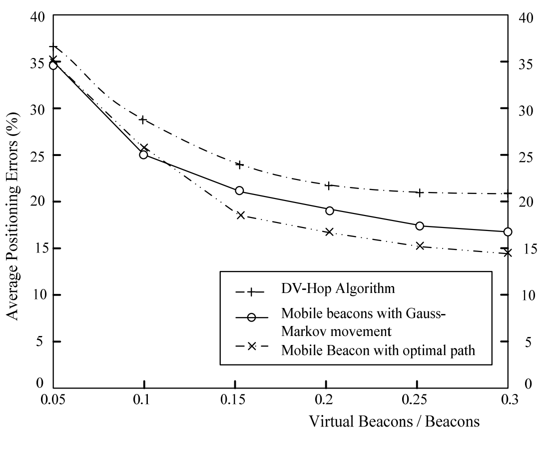


Figure 4. Error Analysis of Positioning Algorithm

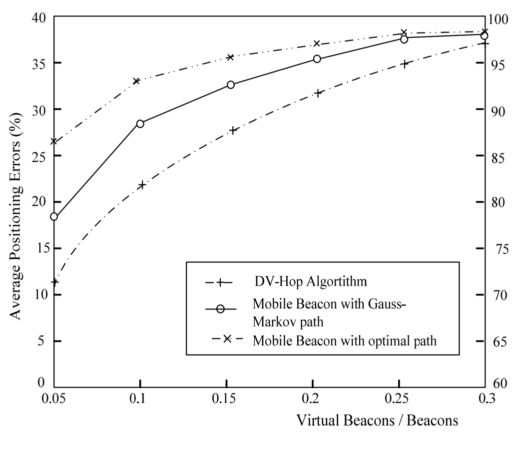


Figure 5. Coverage Analysis of the Positioning Algorithm

In the network, there are 30 unknown nodes before positioning. The mobile beacon moves and forms 9 virtual beacon nodes in the network. The density of the virtual beacons is

0.3 and the positioning error by multiple measurements reaches 14% and positioning coverage reaches 98%.

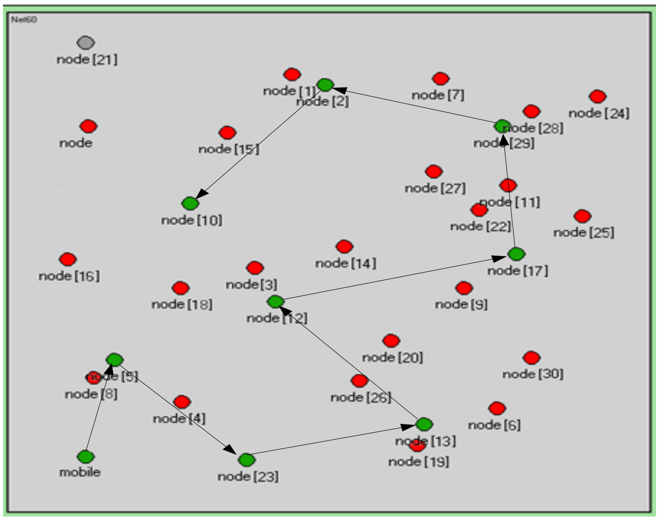


Figure 6. Nodes Distribution and Positioning Scheme

The Figure 6 is one of the nodes positioning scheme. From the figure, all of the nodes can be positioned except the nodes which are far away from the beacon nodes. The positioning coverage is far better than normal DV-Hop positioning algorithm. The positioning performance analysis is shown in Figure 4 and Figure 5 separately. The improved algorithm has better positioning performance.

5. Conclusion

This paper carries researches on positioning algorithm in Wireless Sensor Network by adding mobile beacon nodes in original DV-Hop positioning algorithm. It also carries researches the detailed algorithm of virtual beacon dynamic selection. The final simulation experiment proves the improved algorithm can effectively increase the accuracy of the positioning algorithm, efficiency of the positioning and lower the cost of the network positioning.

References

[1] Ahmed Attia, Ahmad ElMoslimany, Amr El-Keyi, Tamer ElBatt, Fan Bai, Cem Saraydar. MIMO Vehicular Networks: Research Challenges and Opportunities. *Journal of Communications*. 2012; 7(7): 500-513.

[2] Robert C Qiu, Changchun Zhang, Zhen Hu, Michael C. Wicks Towards A Large-Scale Cognitive Radio Network Testbed Spectrum Sensing System Architecture and Distributed Sensing. *Journal of Communications*. 2012; 7(7): 552-566.

[3] Carl K Chang, Hsin-yi Jiang. Time-line based model for software project scheduling with genetic algorithms. *Information and Software Technology*. 2008; 50(11): 1142-1154.

[4] Mozhgan Tavakolifard, Kevin C. Almeroth A Taxonomy to Express Open Challenges in Trust and Reputation Systems. *Journal of Communications*. 2012; 7(7): 538-551.

[5] D Powell, MM Skolnick. *Using genetic algorithm in engineering design optimization with nonlinear constraint in Proc. 5th. Int. Conf. Genetic Algorithms (ICGA-93)*. San Mateo. 1993: 424-431.

[6] Chad Michael Spooner, Nilesh Khambekar. A Signal-Processing Perspective on Signal-Statistics Exploitation in Cognitive Radio. *Journal of Communications*. 2012; 7(7): 567-574.

[7] SF Hwang, RS He. Improving real-parameter genetic algorithm with simulated annealing for engineering problem. *Advances in Engineering Software.* 2006; 37(6): 406-418.

[8] J Zhang, HSH Chung, WL Lo. Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms. *IEEE Transactions on Evolutionary Computation*. 2006; 11(3): 326-335.

[9] Shihong Duan, Yadong Wan, Zhiqiang Guo, Qin Wang. RAAH: Reliability Aware Adaptive Hopping Scheme on Time-varying Channel model in WSN. *JCIT*. 2013; 8(1): 16-25.

[10] LE-Le Qiu, Yan-jun Hu, Yi Wang, Jing - jing Liu. A Novel Multi-media Transmission Method Base on Quantized Compressive Sensing for Wireless Sensor Network. *IJACT*. 2013; 5(1): 496-504.
[11] Chanchan Zhao, Xiaowei Hai, Xiaohua Jiang, Bo Gao. Coal Mine Environment Monitoring with Wireless Sensor Networks. *IJACT*. 2013; 5(1): 505-514.
[12] Zhenjun Luo, Yingjiang Zhang, Zhangfan Pang. Research on Energy-Efficient Intelligent Method for WSN. *JCIT.* 2013; 8(1): 94-102.
[13] Javad Rezazadeh. Fundamental Metrics for Wireless Sensor Networks localization. *International Journal of Electrical and Computer Engineering*. 2012; 2(4): 452-455.