

# A Design of Wireless Sensor Network Node Location Algorithm

Yun-he LI<sup>1</sup>, Shao-hua WU<sup>2</sup>, Hai-tao WU<sup>1</sup>

<sup>1</sup>Faculty of Electronic Information & Mechanical Electrical Engineering, Zhaoqing University, China

<sup>2</sup>Department of Electronics & Information Engineering Harbin Institute of Technology Shenzhen Graduate School, China

yicrane@qq.com

**Keywords:** Sensor network; Location algorithm; DV-hop

**Abstract.** Wireless sensor network belongs to an emerging concept and a technology for creating new application field. Range-free location algorithm can obtain distance information without any specific hardware, and thereby range-free location algorithm is more economic and effective, which is suitable for large-scale wireless sensor network. DV-hop algorithm is a distributed range-free location algorithm. It has advantages of simplicity and none dependence on the fact of measurement error. However, the algorithm has the disadvantage of low localization accuracy. In the paper, DV-hop location algorithm is improved. An improved DV-hop algorithm is proposed, namely a method of improving average hop distance calculation by introducing weight. It can be more rational. Topology relationship is utilized for selecting anchor node. Finally, weighted centroid algorithm is used for estimating location. The algorithm can reduce the algorithm complexity and location error.

## Introduction

Wireless sensor network belongs to an emerging concept and a technology for creating new application field. Currently, sensing technology and sensor network have been identified as one of the most important researches. The stable operation of wireless sensor network node acts as important guarantee of the whole network reliability. Study on low power wireless sensor module research has extremely important studying and research value [1]. DV-hop algorithm is a range-free location algorithm. It has advantages of simplicity and none dependence on the fact of measurement error. However, the algorithm has disadvantage of low localization accuracy [2]. In the paper, DV-hop location algorithm is improved.

## Basic principle of DV-hop location algorithm

DV-hop algorithm is composed of three parts. Firstly, typical distance vector routing protocol is utilized for providing all nodes in the network with minimum hop of distance anchor node. Then, the network average distance of each hop is calculated. It is regarded as a corrected value and broadcasted to the entire network. When unknown node receives the value, the distance to the anchor node is calculated. Finally, their location of unknown nodes is calculated through trilateral or multilateral location [3].

Firstly, each anchor node broadcasts information of a beacon to the whole network. The beacon information includes a hop value initialized to 1. Each node receiving the information will preserve each anchor nodes received by the node. All beacon information contains information of minimum hop value, and information of larger value will be discarded [4]. All nodes in the network will obtain the minimum hop value to each anchor node through the mechanism.

Secondly, once one anchor node obtains the minimum hop value to other anchor nodes, it can estimate average hop distance in the network. Then, controllable flooding method is adopted for broadcasting the value to the whole network [5]. When unknown node receives average hop distance and hops to other anchor node, they can estimate the distance of all receivable anchor nodes through multiplying average hop distance by the hops [6]. The average hop distance can be estimated in anchor node  $i$  by the following formula:

$$HopSize_i = \frac{\dot{a}_{j^i} \sqrt{(x_i - x_j) + (y_i - y_j)}}{\dot{a}_{j^i} h_{ij}} \quad (1)$$

Wherein,  $(x_i, y_i), (x_j, y_j)$  is coordinates of anchor node  $i$  and  $j$ , and  $h_{ij}$  is the hops between anchor node  $i$  and  $j$ .

Each unknown node regards the first received average hop distance value as its average hop distance, and other average hop distance value of other anchor nodes is discarded. Therefore, it can be ensured that most nodes can receive and obtain the first average hop distance value [7]. In the step, anchor node distance can be finally calculated by the unknown node on the basis of average hop distance and minimum hop to the anchor node.

Each unknown node regards the first received average hop distance value as its average hop distance, and other average hop distance value of other anchor nodes is discarded. Therefore, it can be ensured that most nodes can receive and obtain the first average hop distance value [7]. In the step, anchor node distance can be finally calculated by the unknown node on the basis of average hop distance and minimum hop to the anchor node.

$$d_{ij} = HopSize_i \cdot h_{ij} \quad (2)$$

Thirdly, each unknown node calculates its location coordinates. It is assumed that  $(x, y)$  and  $(x_j, y_j)$  are respectively location coordinates of unknown node  $D$  and the  $i$ th anchor node. The distance between the  $i$ th anchor node and unknown node  $D$ . The following formula can be obtained [8].

$$d_i = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (3)$$

Unknown node location  $X$  can be calculated through formula (4) ( $M$  represents anchor node quantity):

$$\begin{aligned} \dot{1} \quad & (x_i - x)^2 + (y_i - y)^2 = d_1^2 \\ \dot{1} \quad & \mathbf{M} \\ \dot{1} \quad & (x_M - x)^2 + (y_M - y)^2 = d_M^2 \end{aligned} \quad (4)$$

Formula (4) can be expanded as follows:

$$\begin{aligned} \dot{1} \quad & x_1^2 - x_M^2 - 2(x_1 - x_M)x + y_1^2 - y_M^2 - 2(y_1 - y_M)y = d_1^2 - d_M^2 \\ \dot{1} \quad & \mathbf{M} \\ \dot{1} \quad & x_{M-1}^2 - x_M^2 - 2(x_{M-1} - x_M)x + y_{M-1}^2 - y_M^2 - 2(y_{M-1} - y_M)y = d_{M-1}^2 - d_M^2 \end{aligned} \quad (5)$$

It is represented in matrix form as follows:

$$AC = b \quad (6)$$

$$\text{Wherein, } A = \begin{bmatrix} 2(x_1 - x_M) & 2(y_1 - y_M) \\ \mathbf{M} & \mathbf{M} \\ 2(x_{M-1} - x_M) & 2(y_{M-1} - y_M) \end{bmatrix}, b = \begin{bmatrix} x_1^2 - x_M^2 + y_1^2 - y_M^2 + d_M^2 - d_1^2 \\ \mathbf{M} \\ x_{M-1}^2 - x_M^2 + y_{M-1}^2 - y_M^2 + d_M^2 - d_{M-1}^2 \end{bmatrix}$$

$X$  least-square estimation is represented as follows:

$$\hat{X} = (A^T A)^{-1} A^T b \quad (7)$$

## DV-hop algorithm improvement

**Hop calculation.** The stage aims at achieving minimum hops<sub>i</sub> to distance anchor node  $i$  in the network, which is the same as traditional DV-hop algorithm.

**Node distance calculation.** The stage is further divided into three steps concretely, which are concretely represented as follows:

Step 1: average hop distance of each anchor node is calculated.

1. Firstly, temporary average hop distance of each anchor node is calculated through formula (8) and formula (9).

$$HopSize_{ij} = \frac{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{h_{ij}} \quad (8)$$

Wherein,  $(x_i, y_i)$  and  $(x_j, y_j)$  are coordinates of anchor node  $i$  and  $j$ ,  $h_{ij}$  is anchor node hop count between  $i$  and  $j$ .  $HopSize_{ij}$  represents average hop distance between anchor node  $i$  and  $j$ .

$$HopSize_i = \frac{\sum_{j=1}^N HopSize_{ij}}{N - 1} \quad (9)$$

Wherein,  $N$  is anchor node quantity,  $HopSize_i$  represents the average hop distance between anchor node  $i$  and all other anchor nodes [9].

2. Then, anchor node  $i$  is used for calculating anchor node  $j$  average hop distance weight and anchor node  $i$  average hop distance.

Average hop distance of each anchor node is established through formula (10) and (11).

$$Wd_{ij} = 1 - (HopSize_{ij} - HopSize_i)^2 / r^2 \quad (10)$$

Wherein,  $r$  is node communication distance.

$$DHop_i = \frac{\sum_{j=1}^N Wd_{ij} HopSize_j}{\sum_{j=1}^N Wd_{ij}} \quad (11)$$

$DHop_i$  represents average hop distance of anchor node  $i$ .

3. Finally, anchor node  $i$  is used for broadcasting  $DHop_i$  to all nodes in the network.

Step 2: All nodes obtain  $DHop_i$  of all anchor nodes after broadcasting in the first step. Formula (12) and (13) are utilized for calculating anchor node  $j$  hop count weight  $Wh_j$  and own average hop distance  $HopSize_i$  in unknown node  $i$  as follows:

$$Wh_j = \frac{1}{h_{ij}} \quad (12)$$

$$HopSize_i = \frac{\sum_{j=1}^N Wd_j DHop_j}{\sum_{k=1}^N Wh_k} \quad (13)$$

Wherein  $N$  refers to anchor node quantity,  $DHop_i$  refers to average hop distance of anchor node  $j$ ,  $h_{ij}$  refers to minimum hop between unknown node  $i$  and anchor node  $j$ .

Step 3: unknown node  $j$  can obtain the distance to anchor node  $i$  through  $s_i \times HopSize_j$  [10]. After the shortest path and unknown node average hop distance are known, the distance among nodes is further improved according to literature through distance error. Figure 1 shows that hop count between unknown node  $A, B$  and anchor node  $D$  is  $hop_{AD} = 5$ ,  $hop_{BD} = 3$  (it is assumed that unknown node  $A, B$  is on the shortest path of anchor node  $C, D$ ). Hop counts between unknown node  $A, B$  and anchor node  $C$  are  $hop_{AC} = 3$  and  $hop_{BC} = 5$ ; The hop count of anchor node  $D$  to anchor node  $C$  through the path is  $hop_{BD} = hop_{AB} + hop_{BD} = 8$ , the average hop distances of unknown node  $A$  and  $B$  are  $HopSize_A$  and  $HopSize_B$ . The distance between  $AD$  and  $AC$  can be calculated through the following formula:

$$\begin{aligned} e_A &= (HopSize_A \times hop_{CD} - l_{CD}) / hop_{CD} \\ l_{CD} &= \sqrt{(X_C - X_D)^2 + (Y_C - Y_D)^2} \\ d_{AD} &= (HopSize_A - e_A) \times hop_{AD} \\ d_{AC} &= (HopSize_A - e_A) \times hop_{AC} \end{aligned} \quad (14)$$

The distance between  $BD$  and  $BC$  can be calculated through the following formula:

$$\begin{aligned}
\hat{e}_A &= (HopSize_B \times hop_{CD} - l_{CD}) / hop_{CD} \\
\hat{l}_{CD} &= \sqrt{(X_C - X_D)^2 + (Y_C - Y_D)^2} \\
\hat{d}_{BD} &= (HopSize_B - e_A) \times hop_{BD} \\
\hat{d}_{BC} &= (HopSize_B - e_A) \times hop_{BC}
\end{aligned} \tag{15}$$

The distance between unknown node and anchor node is more accurate and reliable through the above method.

**Selection and location estimation of anchor node group.** Location coordinates of the unknown node are calculated in the stage. In fact, more basic parameters of the triangle are inner angles of the triangle. Therefore, the judgment method in the literature is adopted in the paper, and it can be obtained through triangle cosine law:

$$\begin{aligned}
\hat{C}_A &= \frac{b_{CA}^2 + c_{AB}^2 - a_{BC}^2}{2b_{CA}c_{AB}} \\
\hat{C}_B &= \frac{a_{BC}^2 + c_{AB}^2 - b_{CA}^2}{2a_{BC}c_{AB}} \\
\hat{C}_C &= \frac{a_{BC}^2 + b_{CA}^2 - c_{AB}^2}{2a_{BC}b_{CA}}
\end{aligned} \tag{16}$$

$A$ ,  $B$  and  $C$  represent three angles of the triangle composed of anchor node  $A$ ,  $B$  and  $C$ . The  $a_{BC}$ ,  $b_{CA}$  and  $c_{AB}$  respectively represent corresponding edges of angles  $A$ ,  $B$  and  $C$ . Therefore, we can immediately obtain the colinearity three-in-one anchor node:

$$NC = \max\{C_A, C_B, C_C\} \tag{17}$$

The cosine value of the minimum angle of triangle is the basis to judge they have colinearity or not. Its scope is from 0.5 to 1.0 (corresponding angle is from 0 to 60 degrees). Excellent anchor node group can be selected for realizing more accurate location estimation through colinearity.

Calculation process is divided into the following four steps:

- 1) All anchor node information is collected by unknown node.
- 2) Three anchor nodes are randomly selected as one group. NC of each group is calculated. Then, NC smaller than or equal to particular collinear threshold and the hop count TH smaller than or equal to specific hop count are selected as good anchor node group.
- 3) Trilateral location method is used for obtaining a series of locations of  $\{I\}$  aiming at selected anchor node group.
- 4) Final location estimation is obtained through a series of locations obtained by centroid algorithm processing.

$$(\bar{x}, \bar{y}) = \frac{\sum_{j=1}^K \bar{x}_j}{K}, \frac{\sum_{j=1}^K \bar{y}_j}{K} \tag{18}$$

Wherein,  $(\bar{x}, \bar{y})$  represents final location estimation.  $(\bar{x}_j, \bar{y}_j)$  represents the  $j$ th location calculated in step 3.  $K$  represents total obtained location quantity.

## Design and realization of improved 3D node location algorithm

**Hop count calculation.** The stage is regarded as the first step of DV-hop algorithm. Each anchor node broadcasts beacon information containing hop count value initialized to 1 and anchor node own ID to the whole network. Each node receiving the information will retain each anchor node received by it. All beacon information contains minimum hop value information. Information of larger value is discarded. All nodes in the network can obtain minimum hop value  $s_i$  to anchor node  $i$  through the mechanism.

**Distance calculation.** Distance is calculated by node distance calculation method in DV-hop improved algorithm.

**Location estimation.** In the stage, unknown node location coordinates are estimated as shown in figure 1.

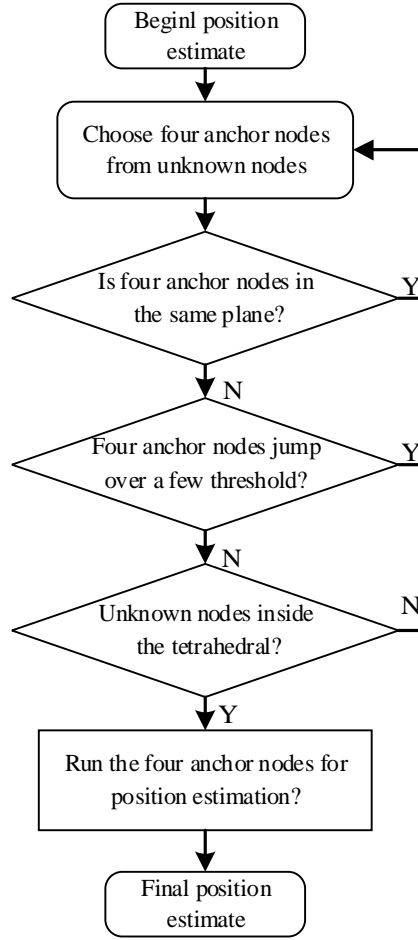


Figure 1 location estimation flowchart

1). It is judged that four selected anchor nodes are not in the same plane.

It is assumed that 4 anchor nodes are selected randomly, namely  $A(x_1, y_1, z_1)$ ,  $B(x_2, y_2, z_2)$ ,  $C(x_3, y_3, z_3)$ ,  $D(x_4, y_4, z_4)$ , whether four anchor nodes are located on the same plane or not is judged according to the formula (19).

$$d = \begin{vmatrix} x_4 - x_1 & y_4 - y_1 & z_4 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \end{vmatrix} \quad (19)$$

If  $|d| = 0$ , it can be judged that the nodes are located in the same plane, which can not form a ball. Namely, they can not be located. Four anchor nodes should be selected again for judgment.

2). It is judged that unknown nodes are within tetrahedron of anchor node.

If it is known that four anchor nodes  $A$ ,  $B$ ,  $C$ ,  $D$  not on the same plane are selected for forming tetrahedron, whether the unknown node  $M$  is within the tetrahedron or not is judged. Wherein, whether the unknown node  $M$  is within the tetrahedron is calculated firstly through calculating the volume of tetrahedron. It is assumed that the volume of the tetrahedron  $ABCD$  is  $V$ . The volume of tetrahedrons  $ABCM$ ,  $ABMD$ ,  $AMCD$  and  $MBCD$  is respectively  $V_1$ ,  $V_2$ ,  $V_3$  and  $V_4$ . If  $V_1 + V_2 + V_3 + V_4 - V = 0$ , it can be judged that  $M$  is within the tetrahedron, otherwise it is outside the tetrahedron.

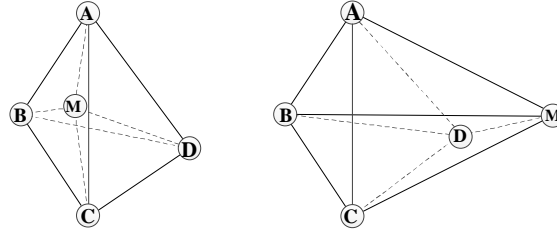


Figure 2 Node location plan

**Location estimation.** Appropriate frequency should be selected in the process of selecting anchor node group. Because it is randomly selected, some selected anchor nodes may be selected. For example, when there are 20 anchor nodes around the unknown node, 4 nodes are selected, there are  $C_{20}^4 = 4850$  selection methods. The selection frequency can be limited in order to reduce algorithm time cost. If set threshold of 1000 times has been able to satisfy requirements, and suitable anchor node groups are not still selected after selection for 1000 times. The nodes can be judged and selected continuously until all anchor nodes are exhausted. If suitable anchor node groups can not be selected finally for location, it is declared that the node is not positioned.

After one anchor node group is selected, centroid algorithm can be utilized for location estimation, thereby obtaining one location. Anchor node group selected each time can obtain one coordinate value. Then the coordinate value can be broadcasted to the nearest anchor node to the unknown node. The anchor node utilizes the received unknown node estimation coordinate values and other three anchor nodes for self-positioning. After one coordinate value is obtained, one location estimation deviation value between one anchor node estimation coordinate and actual coordinate value can be obtained.

It is assumed that actual coordinate value of anchor node  $M$  is  $(x, y, z)$ . The estimation coordinate value is  $(\hat{x}, \hat{y}, \hat{z})$ . The deviation value coordinate between actual coordinate and estimation coordinate is represented by  $(D_x, D_y, D_z)$ . Calculation of the deviation value is shown in formula (20).

$$\begin{aligned} D_x &= x - \hat{x} \\ D_y &= y - \hat{y} \\ D_z &= z - \hat{z} \end{aligned} \quad (20)$$

Then, the anchor node sends the deviation value back to the unknown node. The unknown node utilizes the deviation value to optimize and process the estimation coordinates of unknown nodes. Namely, the collected deviation value and the estimation coordinate value are used for weighted centroid algorithm. Final location coordinates of the unknown node are obtained at last as follows:

$$\bar{x} = \frac{\sum_{i=1}^n x_i \times \frac{1}{Dx_i}}{\sum_{i=1}^n \frac{1}{Dx_i}}, \bar{y} = \frac{\sum_{i=1}^n y_i \times \frac{1}{Dy_i}}{\sum_{i=1}^n \frac{1}{Dy_i}}, \bar{z} = \frac{\sum_{i=1}^n z_i \times \frac{1}{Dz_i}}{\sum_{i=1}^n \frac{1}{Dz_i}} \quad (21)$$

Wherein,  $(\bar{x}, \bar{y}, \bar{z})$  represents final location coordinate of unknown node,  $(x_i, y_i, z_i)$  represents unknown node location coordinate obtained by the selected  $i$  th anchor node in line with requirements.  $n$  represents total location quantity of the obtained unknown node (namely total selected anchor node groups in line with requirements),  $\frac{1}{Dx_i}, \frac{1}{Dy_i}, \frac{1}{Dz_i}$  represents the weight of the  $i$  th coordinate. When the

value in  $(D_x, D_y, D_z)$  is zero, the coordinate value of the unknown node corresponding to the error value on the coordinate axis is selected as the coordinate value of the unknown node on the coordinate value. For example, when  $D_x = 0$ ,  $x_i$  is regarded as the corresponding coordinate value of the unknown node on  $x$  axis. When the value is smaller than zero, the absolute value is obtained.

## Simulation analysis of improved 3D location algorithm

Our designed simulation scenario design in MATLAB is as follows in order to verify expansion correctness of our algorithm: 500 sensor nodes are randomly distributed within the space 100 meters long, 100 meters wide and 50 meters high, including 40 temporary anchor nodes. The communication radius is provisionally set as 35 meters. Two thresholds should be set, namely hop count threshold and threshold for determining whether the sensor nodes are within the tetrahedron or not, and it is referred to as 'internal threshold'. When the hop count threshold is 6, and internal threshold is 2, location node quantity is 458, average location error is 10.5421, and the precision is 0.3012. (The data is regarded as average value of 20 experiments under the same condition, the selection threshold of the anchor node group is set as 10000 times, which are the same as the following data).

Table 1. Influence of hop count threshold

Hop count threshold	Location node quantity		Average location error		Location precision	
	Improved 3D algorithm	DV-hop 3D DV-hop	Improved 3D algorithm	DV-hop 3D DV-hop	Improved 3D algorithm	DV-hop 3D DV-hop
7	200	480	13.2863	24.7521	0.4429	0.8251
8	273	480	15.8623	24.7521	0.5287	0.8251
9	377	480	18.9947	24.7521	0.6332	0.8251

Note: communication radius: 30m; anchor node quantity:20; threshold in the tetrahedron: 2

Table 1 shows the influence of hop count threshold on location precision and location node quantity, location node quantity of the newly-designed 3D location algorithm is correspondingly increased with increase of hop count threshold, however the location precision is decreasing, and there is no influence on DV-hop 3D expansion.

Table 2. Influence of threshold in the tetrahedron

Hop count threshold	Location node quantity		Average location error		Location precision	
	Improved 3D algorithm	DV-hop 3D DV-hop	Improved 3D algorithm	DV-hop 3D DV-hop	Improved 3D algorithm	DV-hop 3D DV-hop
1	192	480	12.4203	24.7521	0.4140	0.8251
2	200	480	13.2871	24.7521	0.4429	0.8251
3	205	480	13.8679	24.7521	0.4623	0.8251
4	231	480	14.2237	24.7521	0.4741	0.8251

Note: communication radius: 30m; anchor node quantity: 20; hop count threshold: 6;

Table 2 shows the influence of threshold in the tetrahedron on location precision and location node quantity, and location node quantity of the newly-designed 3D location algorithm is correspondingly increased with increase of internal threshold. However, the location precision is decreasing, and there is no influence on DV-hop 3D expansion.

Table 3. Influence of anchor node quantity

Hop count threshold	Location node quantity		Average location error		Location precision	
	Improved 3D algorithm	DV-hop 3D DV-hop	Improved 3D algorithm	DV-hop 3D DV-hop	Improved 3D algorithm	DV-hop 3D DV-hop
20	200	480	13.2871	24.7521	0.4140	0.8251
25	253	480	13.2871	24.0573	0.4429	0.8019
30	330	480	13.8679	23.8623	0.4623	0.7954
35	395	480	14.2237	22.9922	0.4741	0.7764

Note: communication radius:30m; hop count threshold: 6; threshold in the tetrahedron: 2

Table 3 shows the influence of anchor node quantity on location precision and location node quantity, location node quantity and location precision of the newly-designed 3D location algorithm is

correspondingly increased with increase of threshold in the tetrahedron. 3D expansion location precision of DV-hop is also improved with improvement of anchor node quantity.

Table 4. Influence of communication radius

Communication radius	Location node quantity		Average location error		Location precision	
	Improved 3D algorithm	DV-hop 3D DV-hop	Improved 3D algorithm	DV-hop 3D DV-hop	Improved 3D algorithm	DV-hop 3D DV-hop
20	30	480	8.8861	19.3662	0.4443	0.9683
25	98	480	11.0925	23.6752	0.4437	0.9470
30	200	480	13.2871	24.7521	0.4629	0.8251
35	298	480	15.1165	25.0635	0.4319	0.7161

Note: anchor node quantity: 20; hop count threshold: 6; threshold in the tetrahedron: 2;

Table 4 shows the influence of communication radius on location precision and location node quantity, location node quantity of the newly-designed 3D location algorithm is prominently increased, and the location precision is slightly increased with increase of internal threshold. 3D expansion location precision of DV-hop is also improved with improvement of anchor node quantity.

Table 1, 2, 3 and 4 show that the newly-design 3D location algorithm is compared with 3D DV-Hop algorithm, the newly-design 3D location algorithm has prominent advantage in location precision. However, the newly-design 3D location algorithm is jointly affected by hop count threshold, threshold in the tetrahedron, anchor node quantity and communication radius in the location node quantity, which is greatly affected by anchor node quantity, hop count threshold and communication radius. However, high location precision and coverage scope can be obtained through proper selection of all parameters.

## Conclusion

DV-hop algorithm is a distributed range-free location algorithm. It has advantages of simplicity and none dependence on the fact of measurement error. However, the algorithm has the disadvantage of low localization accuracy. In the paper, DV-hop location algorithm is improved. An improved DV-hop algorithm is proposed, namely a method of improving average hop distance calculation by introducing weight. It can be more rational. Topology relationship is utilized for selecting anchor node. Finally, weighted centroid algorithm is used for estimating location. The algorithm can reduce the algorithm complexity and location error.

## Acknowledgements

This work was financially supported by the PhD Start-up Fund of Natural Science Foundation of Guangdong Province of China (Grant No.2014A030310286) and the National Natural Science Foundation of China (Grant No. 61001092); National Natural Science Foundation of China (Grant No. 61371102).

## References

- [1] Kumar S, Lobiyal D K. An Advanced DV-Hop Localization Algorithm for Wireless Sensor Networks[J]. Wireless Personal Communications An International Journal, 2015, 57(2):1365-1385.
- [2] Gui L, Val T, Wei A, et al. Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks[J]. Ad Hoc Networks, 2015, 24:55-73.
- [3] Wu J, Chen H, Lou W, et al. Label-Based DV-Hop Localization Against Wormhole Attacks in Wireless Sensor Networks. Pervasive and Mobile Computing, 2014, 16:22-35.
- [4] Han G, Chao J, Zhang C, et al. The impacts of mobility models on DV-hop based localization in Mobile Wireless Sensor Networks. Journal of Network & Computer Applications, 2014, 42(6):70-79.



- [5] Zhang Y, Xiang S, Fu W, et al. Improved Normalized Collinearity DV-Hop Algorithm for Node Localization in Wireless Sensor Network. *International Journal of Distributed Sensor Networks*, 2014, 2014(2):1-14.
- [6] Wang Y, Qian K. An improved DV-HOP positioning algorithm based on Hop distance correction// *International Computer Conference on Wavelet Active Media Technology and Information Processing*. IEEE, 2014:415 - 418.
- [7] Honglong Chen, Wei Lou, Zhi Wang, et al. Securing DV-Hop localization against wormhole attacks in wireless sensor networks. *Pervasive and Mobile Computing*, 2014, 16:22–35.
- [8] Li X, Yan L, Pan W, et al. Optimization of DV-hop localization algorithm in hybrid optical wireless sensor networks[J]. *Journal of Heuristics*, 2014, 21(2):177-195.
- [9] Gayan S, Dias D. Improved DV-Hop algorithm through anchor position re-estimation// *IEEE Asia Pacific Conference on Wireless and Mobile*. 2014:126-131.
- [10] Wang Y, Fang Z, Chen L. A New Type of Weighted DV-Hop Algorithm Based on Correction Factor in WSNs. *Journal of Communications*, 2014.