DSLA: Dynamic Sampling Localization Algorithm Based on Virtual Anchor Node

Yanru Chen[1], Bingshu Yan[1], Liangxiong Wei[1], Min Guo[1], Feng Yin[2], Qian Luo[3], Wei Wang[1,4*] and Liangyin Chen[1,4*]

[1] School of Computer Science, Sichuan University,
Chengdu, China
[e-mail: yanruc1@student.unimelb.edu.au, weilx_scu@aliyun.com, guomin@stu.scu.edu.cn,
wang.david.wei@stu.scu.edu.cn, chenliangyin@scu.edu.cn]
[2] School of Computer Science and Technology,
Southwest Minzu University,
Chengdu, China
[e-mail: yf_eagle@swun.edu.cn]
[3] Second Research Institute, General Administration of Civil
Aviation of China, Chengdu, China
[e-mail: luoqian@caacetc.com]
[4] Institute for Industrial Internet Research, Sichuan University,
Chengdu, China
*Corresponding author: Wei Wang and Liangyin Chen
Yanru Chen and Bingshu Yan have equal contribution

Abstract

Compared with the localization methods in the static sensor networks, node localization in dynamic sensor networks is more complicated due to the mobility of the nodes. Dynamic Sampling Localization Algorithm Based on Virtual Anchor (DSLA) is proposed in this paper to localize the unknown nodes in dynamic sensor networks. Firstly, DSLA algorithm predicts the speed and movement direction of nodes to determine a sector sampling area. Secondly, a method of calculating the sampling quantity with the size of the sampling area dynamically changing is proposed in this paper. Lastly, the virtual anchor node, i.e., the unknown node that got the preliminary possible area (PLA), assists the other unknown nodes to reduce their PLAs. The last PLA is regarded as a filtering condition to filter out the conflicting sample points quickly. In this way, the filtered sample is close to its real coordinates. The simulation results show that the DSLA algorithm can greatly improve the positioning performance when ensuring the execution time is shorter and the localization coverage rate is higher. The localization error of the DSLA algorithm can be dropped to about 20%.

---

1. Introduction

The node localization in wireless sensor networks refers to how a node that does not know its own location in the network can compute its location coordinates via a localization algorithm. The node localization is the premise of location-based applications in wireless sensor networks (WSNs). Depending on whether it is needed to measure the physical distance or not, the technologies of node localization can be mainly divided into two categories, range-based and range-free localization algorithm. The range-based algorithms mainly include TOA localization [1], TDOA localization [2], AOA localization [3], RSSI localization [4] and so on [5-7]. The range-free algorithms mainly include Weighted Centroid localization [8], DV-hop localization [9-11], RSS-based localization [12] and so on [13]. In the literature [14], there are also some other localization algorithms about mobile anchor node assisted localization in static wireless sensor networks. At present, most research results of node localization are based on the static wireless sensor networks. However, if these results are applied to the dynamic wireless sensor networks, the mobile characteristics of nodes will affect the positioning performance of the algorithms. One of the serious challenges faced in the dynamic wireless sensor networks is how to implement the low-energy and high-accuracy node localization.

Some localization algorithms in dynamic sensor networks are based on the Monte Carlo Localization (MCL) theory, whose basic concept is that it makes full use of N samples to represent the posterior probability distribution of node localization. The MCL algorithms utilise such information to predict the movement speed of nodes and connectivity between nodes. However, there are still two limitations in MCL algorithms: 1) The sample area is too large, resulting in a high sampling rate, high computation complexity and long localization time. 2) It only uses the information of the one-hop and two-hop anchor nodes, instead of involving in the information of other unknown nodes. Therefore, the performance of node localization is worse in low-density area than normal. To overcome these two limitations, this paper proposes a Dynamic Sampling Localization Algorithm Based on Virtual Anchor Node (DSLA). The main contributions are as follows:

- We use Newton interpolation to predict the motion velocity and direction of the node at the next moment and obtain the sector area as the sampling area where the node is most likely to be located.

- We propose a calculation approach to change the sampling quantity with the change in the sampling area and the density of unknown node in this area. This approach can reduce the number of re-sampling and computation complexity.

- In the filtering phase, after using the unknown node whose preliminary possible location area is small as the virtual anchor node to narrow the possible area of other unknown nodes, we use the preliminary possible location area as the filtering condition to obtain the valid samples. Since the filtering conditions are stricter, the filtered samples are closer to the true position coordinates. Therefore, the positioning accuracy is significantly improved, and this accuracy is still good under the condition of lower anchor node density, which defines the ratio of the number of localizable nodes to the number of unknown nodes.

This paper is organized as follows. Section 2 describes the related works in this field. Section 3 gives the definitions and motion model. Section 4 presents the innovations of the

DSLA algorithm in details. In section 5, by simulation evaluation, we compare the advantages and disadvantages of our proposed methods with the existing algorithms. Section 6 summarizes this paper and gives our future directions.

2. Related Works

Under the dynamic network condition that both the anchor node and the unknown node would move, the famous algorithm for localization is the Monte Carlo Localization (MCL) Algorithm. The Monte Carlo method is actually an iterative Bayesian filter, whose basic idea is to use the valuable samples to represent the posterior probability density distribution of the states to be estimated in order to get the estimated solution. The Monte Carlo methods were originally applied to mobile robot positioning. Later, Hu and Evans applied it to the mobile sensor networks and proposed a Monte Carlo Localization (MCL) algorithm [15] in 2004. Also, Minhan Shon and Minho Jo conducted research on the iterative localization [27] [28] in 2012 and 2016. In the MCL algorithm, $L_t = \{l_t^1, l_t^2, ..., l_t^N\}$ is used to express the collection of the particles, which includes N position samples. The problem of node localization is described as follows. Denote t as a discrete time slot, l_t as the position distribution of the unknown node at time t and o_t as the anchor information observed at time t . The state transition equation is $p(l_t | l_{t-1})$, which indicates a prediction of the position of the moment t based on the previous moment t-1 . The observation equation is $p(l_t | o_t)$, which indicates the probability that the node is located at l_t, under the condition of o_t . This algorithm is implemented such that L_t is calculated by L_{t-1} and o_t . The implement steps of the MCL algorithm are divided into three steps: initialization, sample selection and location estimation. The sampling selection step includes prediction and filtering phase. Prediction means that the algorithm updates the sample collection using sample data in the previous moment. Filtering means that this algorithm filters out the unsatisfied constrain condition sample. The algorithm requires that the sample number is enough. Therefore, it needs to be predicted and filtered many times.

After the proposal of the MCL algorithm, some improvements were proposed as follows, MCB (Monte Carlo Localization Boxed) algorithm [16] proposed by Baggio Aline improves sampling area in the MCL algorithm. In the MCB algorithm, the unknown nodes utilize the information received from the neighbors of the anchor node to determinate the anchor box. Then this algorithm decides the sampling box by the samples and maximum motion speed in the previous moment. Lastly, it uses the interpolation area between the anchor box and the sampling box as sampling area. Compared with the MCL algorithm, the MCB algorithm shortens sampling time and improves localization precision. However, when the density of anchor nodes in the network is very low, the localization performance of the MCB algorithm is not good.

Similarily, in order to improve sampling efficiency, Enrique.S and Vijayanth.V proposed dual and mixture Monte Carlo Localization algorithm by exchanging the predicting and filtering phases in the MCL algorithm [17]. In literature [18], Hamid and Mehdi proposed a novel method to improve sampling efficiency, which uses uniform sampling in the sampling box instead of random sampling in the MCL algorithm.

Massomeli and Suprakash [19] firstly applied the neighbor of the unknown node for locating the unknown nodes to be located and proposed MSL and MSL* algorithm. These algorithms utilize the information of the one-hop and two-hop neighbor node for improving

the localization accuracy. However, these algorithms need a great amount of computing cost during the localization process, which lengthens the localization time of the node and affects the efficiency of these algorithms. The real-time characteristics of locating the unknown node are important for a mobile sensor networks, but these algorithms are limited in the practical scenario.

In literature [20], the author proposed the WMCL algorithm. Firstly, this algorithm utilizes the bounding-box [21] of the one-hop and two-hop anchor nodes to create the sampling box. Then, it makes full use of the communication radius of the two-hop neighbor anchor node and the unknown node error to estimate the coordinate that can narrow the sampling box. However, WMCL algorithm does not restrict filtering condition but affects the accuracy of valid samples and increases the localization error.

Range-based MCL algorithm [22] utilizes the physical distance and communication information between the unknown node and the anchor node, and in the filtering phase, it obtains more accurate samples to reduce the localization error. RSS-MCL algorithm [23] uses RSSI measuring distance technology and the received information indication that is received by the unknown node to improve the predicting and filtering process and increase the localization performance. OTMCL (Orientation Tracking-based Monte Carlo Localization) [24] algorithm introduces the direction angle information for better localization effect and can be applied in an environment where the sensor node can get its motion direction.

DSLA algorithm proposed in this paper is better than the existing dynamic localization algorithms in the aspect of sampling efficiency, localization time and localization accuracy.

3. Definitions and Motion Model

This section provides the definitions of several concepts that will be used in the following sections. At the same time, the suitable motion model plays an important role in analyzing the DSLA algorithm. Therefore, this section gives us the definitions and motion model in details.

3.1 Definitions

Definition 1. The Degree of Irregularity (DOI) of node radiation: The degree of irregularity of node radiation varies in all directions. Assuming there exists a maximum communication radius and a minimum communication radius, which the nodes outside the maximum communication circle cannot receive the information from the node, while the ones inside the minimum communication circle can receive the information certainly. In other words, the communication radius of the nodes is between the maximum radii and minimum radii.

Definition 2. The virtual anchor node: After the unknown nodes get primary possible area, this area can be used to assist other unknown nodes for localization. Then, the unknown node is called the virtual anchor node.

Definition 3. The inevitable communication circle: This is a circle that uses the coordinates of the nodes as the center of the circle and the minimum communication radius as the radius of the circle. So, the inevitable communication circle of the anchor node is a circle that uses the coordinate of the anchor node as the center of the circle; the inevitable communication circle of the unknown node uses the center of the circle that is used as the circumscribed circle of the possible localization area.

Definition 4. The virtual communication circle is a virtually existing circle. For example, if the actual communication radius of a node is R, and the node s is the 2 hop neighbor of the node d, then the virtual communication circle of the virtual node d relative to the node s is a

circle that uses the ordinates of the node d as the center and uses $2R$ as the radii of the circle.

3.2 Motion Model

In the field of current dynamic sensor networks, the most widely used motion model is the Random Waypoint Mobility Model (RWMM) [25]. The motion model is defined as randomly selecting two points D and S in the moving area and randomly selecting a velocity v in the range (v_{min}, v_{max}), assuming that a node moves from a point S (starting point) along a straight line to a point D (end point) at a speed v. The minimum moving speed is v_{min}, the maximum moving speed is v_{max}. A time t is randomly selected for a pause within a certain time range (t_{min}, t_{max}) of the end point; t_{min} is the minimum pause time, and t_{max} is the maximum pause time. Then, the end point D of this time is used as the starting point of the next stage and these operations are repeated. Although there are many hypotheses about the Random Waypoint Mobility Model, it is an idealized theoretical model and it has a good advantage as an analytic model. So, this model is selected as the motion model of nodes in the mobile wireless sensor networks.

4. DSLA algorithm

As a MCL algorithm, the DSLA algorithm obtains samples with weights by sampling and filtering. It uses samples to estimate the posterior probability distribution of node positions. DSLA algorithm mainly includes predicting, filtering, re-sampling and position estimation phases. The flowchart of the DSLA algorithm is shown in **Fig. 1**. The most important phases are the prediction phase and filtering phases. The prediction phase is meant to determine the sampling area and obtain samples. The filtering phase uses the filtering condition to filter the samples in order to obtain a set of valid samples.
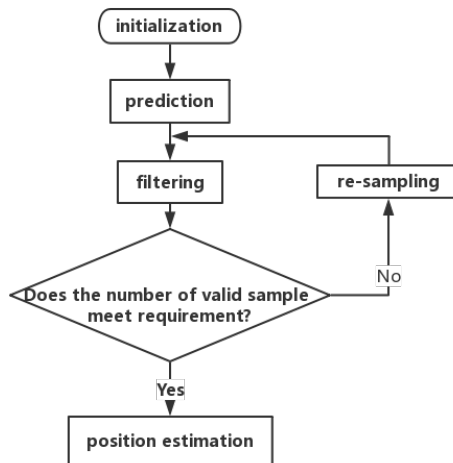

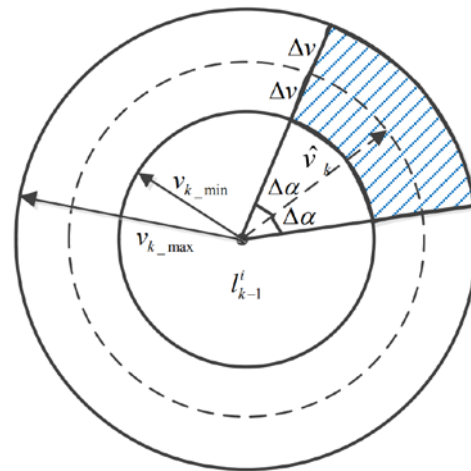**Fig. 1.** DSLA algorithm flowchart   **Fig. 2.** Sampling Area Diagram

In the prediction phase, DSLA algorithm predicts the motion velocity and direction of the node according to the motion trajectory of previous moments and uses the location area where the node is most likely to appear at the next moment as the sampling area of the node. After determining the sampling area, the required number of samples is calculated according to the

size of the area , the density of the node to be located and random samples in the sampling area. In the filtering phase, the unknown node collects the location information of the multi-hop anchor node neighbor as well as the unknown node neighbor and its hop count of the distance. The unknown node utilizes the inevitable communication circles and the virtual communication circles of the neighbor nodes. The possible location area is continuously segmented and used it as the filtering condition. If the number of samples satisfying the filter condition does not meet the requirements of the system, it should enter the re-sampling area and be re-sampled until the number of valid samples meets the requirements. In the position estimation phase, a weighted average calculation is executed on the valid samples to estimate the coordinates of the unknown nodes. The following is a detailed description of determining sampling area, calculation of the number of samples and acquisition of filtering conditions.

4.1 Sampling Area Determination

Firstly, this algorithm needs to determine the sampling area. In the sampling phase of the DSLA algorithm, the size and accuracy of the sampling area are the most important factors. The sampling area proposed in this paper is a sector segment area [26]. The algorithm considers the error distribution of the prediction method while considering the motion position prediction of the positioning node in the next moment and obtains an area where the node is most likely to be located as the sampling area. The node predicts the future possible position of the node by establishing a more accurate mobility model, which can effectively improve the positioning accuracy. By interpolating the positions of the previous moments of the node to obtain a rough motion trajectory in order to predict the motion direction of the node, the possible motion range of the node and the range of the sampling area can be effectively reduced. This paper introduces the Newton interpolation method for predicting the direction and speed of motion. The principle of the Newton interpolation method is as follows:

Suppose that the value of the function $f(t)$: $f(t_0)$, $f(t_1)$, $\cdots\cdots f(t_{k-1})$ at k different times t_0, $t_1 \cdots\cdots t_{k-1}$ are x_0, $x_1 \cdots\cdots x_{k-1}$. According to the Newton interpolation polynomial, the function values X_k of the previous k moment used to compute the function value X_k are as follows:

$$
\begin{aligned}
N_k(t) = f(t_0) &+ f[t_0,t_1](t-t_0) \\
&+ f[t_0,t_1,t_2](t-t_0)(t-t_1)+... \\
&+ f[t_0,t_1,...t_k](t-t_0)(t-t_1)...(t-t_{k-1})
\end{aligned}
\tag{1}
$$

According to the collections of particles $L_0, L_1,...,L_{k-1}$ at the previous k moments and motion model, the node can determine the collection of particles L_k. In this motion model, a node cannot determine motion speed and direction at the current moment, but can only know that the maximum speed is v_{max}. The Newton interpolation operation is performed on the node in the two directions of coordinate x and coordinate y, and the coordinates of the node at time k are obtained as (x_k, y_k). So, the velocity of the two directions, v_{kx} and v_{ky}, at time t are predicted as follows:

$$
v_{kx} = \frac{(x_k - x_{k-1})}{k}
\tag{2}
$$

$$
v_{ky} = \frac{(y_k - y_{k-1})}{k}
\tag{3}
$$

k is the size of every time slot. From the above, the velocity v_k and motion direction of the unknown node at time k can be obtained as follows:

$$\hat{v}_k = \sqrt{v_{kx}^2 + v_{ky}^2}, \hat{\alpha}_k = \arctan(\frac{y_k - y_{k-1}}{x_k - x_{k-1}}) \tag{4}$$

If the coordinates of all the previous moments are used to estimate the current position, Newton interpolation increases the amount of computing in addition to storing more data. Therefore, considering the theoretical and actual requirement, DSLA algorithm uses the latest 3 position coordinates to estimate the possible motion velocity v_k of the current moment. Because the motion velocity cannot be estimated at $k = 0$, $k = 1$ and $k = 2$, the motion velocity is the maximum velocity that the system sets, $\hat{v}_k = v_{\max}$.

As **Fig. 2** shows, the dash area is the sector sampling area in the DSLA algorithm. There is a certain error in the interpolation calculation method, and it meets the normal distribution. The nodes to be located may be located in the surrounding area of the four directions of the predicted position, so a speed parameter Δv and an angle parameter $\Delta \alpha$ are introduced in the algorithm. The obtained sector area is the most likely location of the node to be located at the next moment. First, the algorithm causes the nodes to increase or decrease the size of Δv in the estimation direction to obtain two radii, $v_{k\_\max}$ and $v_{k\_\min}$, and the estimated position at the last moment is used as the center of the circle to get two circles, as shown in **Fig. 2**. Since the motion trajectory of the node is relatively smooth, the motion direction of the node does not change abruptly. Therefore, the algorithm introduces an angle parameter $\Delta \alpha$ in the estimation direction. It expands the angle $\Delta \alpha$ of the clockwise and counterclockwise directions in the estimation direction of the node. Then, the shaded area shown in **Fig. 2** is obtained, which is the most possible area of the nodes at the current moment, system settings: $\Delta \alpha = 2\pi/9$, $\Delta v = 0.5 v_k$ (section 5.2 introduces how to obtain two parameters). In the motion model, the range of the node velocity is $(v_{\min}, v_{\max})$, and when $v_{k\_\min} > v_{\max}$, assume $v_{k\_\max} = v_{\max}$; when $v_{k\_\min} < v_{\max}$, assume $v_{k\_\min} = v_{\min}$. In the sampling area, n sample points are randomly sampled (n as actual sample number, as motion model, introduces how to obtain n). Therefore, with the shaded area as the sampling area, the mathematics expression is as follows:

$$L_t = \left\{ (x_k^i, y_k^i) \left| \begin{array}{l} v_{k\_\min} \le d[(x_k^i, y_k^i), x_{k-1}, y_{k-1}] \le v_{k\_\max} \\ \bigcap \left| \dfrac{y_k^i - y_{k-1}}{x_k^i - x_{k-1}} \right| \le \tan \Delta \alpha \end{array} \right. \right\} (0 \le i \le n) \tag{5}$$

4.2 Sampling Number Computation

The algorithm needs to confirm the number of sampling in the sampling phase of the DSLA algorithm. After confirming the sampling area, this paper proposes a method of determining the sampling number according to sample area size and node density change to increase sampling rate and reduce energy consumption. For wireless sensor network of limited energy, there is need to reduce the energy consumption of localization as much as possible, especially for dynamic wireless sensor network: every time slot needs to locate every unknown node and decrease sample times to save the energy of node. In literature [26], this algorithm dynamically adjusts the sample number according to the area size of the sampling area and

decreases the sampling times, save computation and decrease localization time.

When the anchor box created by the anchor node is quite small, only a few samples are needed to achieve a certain positioning accuracy. When the anchor box is large, a certain number of samples are required to achieve the required positioning accuracy. In this paper, the number of samples is calculated according to the size of the sampling area obtained in the motion prediction phase and the density of the node to be located, and the total number of sampling times is minimized on the basis of ensuring the positioning accuracy. The number of samples required for positioning is as follows:

$$n = \frac{S_{sample}}{R \times UD} \times N_U = \frac{S \times S_{sample}}{N_U \times \pi R^3} \times N$$

(6)

The density of the nodes in the networks is as follows: $UD = N_U \Big/ (\frac{S}{\pi(R)^2})$. n is the actual sampling numbers; R is the communication radius of the node; S_{sample} is the range of sampling area; N is the maximum sampling number set by the system; S is the range of the whole network; N_U is the number of the unknown nodes in the networks. In order to prevent the sampling area in the sampling phase from being too small, the number of samples is extremely small, and the minimum number of samples set by the system is N_{min}. If the actual number of samples calculated, n, is less than N_{min}, then N_{min}. In order to prevent excessive sampling, when the calculated actual number of samples, n, is greater than the maximum number of samples, N, that are set by the system, then $n = N$.

4.3 Obtaining Filtering Condition

After obtaining the DSLA algorithm samples at the sampling area, the filtering phase starts. The node to be located uses the received information of neighbor nodes to obtain filtering condition and filters out samples that do not meet the criteria to get valid samples for position estimation of the node to be located. In the DSLA algorithm, by using the idea of the Bounding-Box algorithm [15], the communication radius of the node is abstracted into a square. In the filtering phase, a preliminary possible location area is firstly determined by using the communication range of the box shape of the one-hop and two-hop anchor node neighbors of the node to be located; the shape is a rectangle. The unknown node which obtains preliminary possible location is called the virtual anchor node. The virtual anchor node can assist the other unknown nodes to be located, and it continuously narrows the possible area of other nodes to be located. Specifically, each node to be located continuously reduces its own possible location area by using its anchor node neighbor and the virtual anchor node neighbor with respect to its own possible virtual communication circle and the inevitable communication circle of its neighbor node.

The virtual communication circle of the anchor node is used to segment the possible location area of the positioning node, as shown in **Fig. 3**. The polygon ABCD is the preliminary possible location area obtained by the node $U1$ to be located, and the circle in the figure is the virtual communication circle of the anchor node $A1$. It is assumed that $U1$ is the two-hop neighbor of $A1$, and the radius of the virtual communication circle is $2R$. $U1$ is located in the virtual communication circle. So, the node $U1$ to be located can be located at a possible position outside the virtual communication circle and a segment out of this area. Firstly, the vertices of the possible localization areas outside the virtual communication circle are respectively connected, C and D, and the anchor node $A1$ are respectively connected and

two intersection points are formed by the two lines respectively intersecting the virtual communication circle. Then, the virtual communication circle is respectively made through the two intersection points. The two tangent lines divide the possible position area of the node $U1$ to be located into a shaded portion as shown in **Fig. 3**, as an updated possible position area of $U1$.
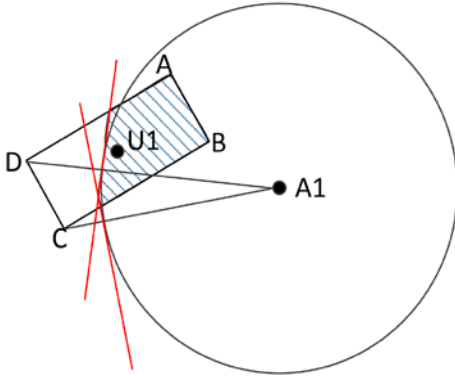

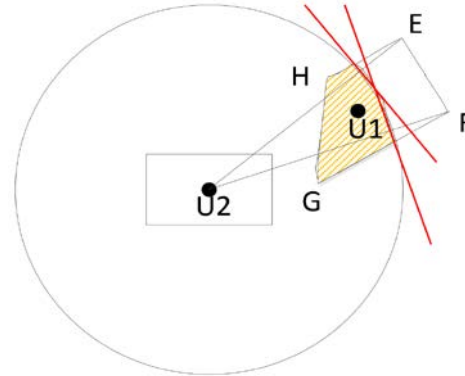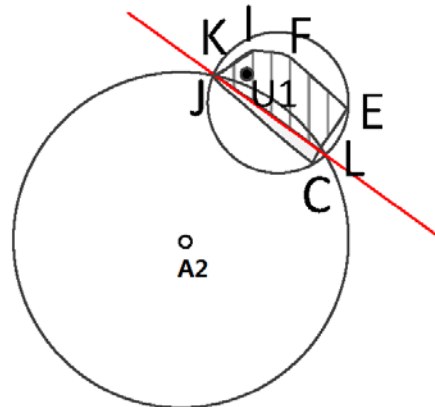**Fig. 3.** Virtual communication circle of node *A1*   **Fig. 4.** Virtual communication circle of node *U2*

Fig. 5. Utilization of the inevitable communication circle of node *A2*

The virtual communication circle of the virtual anchor node is used to segment the possible location areas of the positioning node, as shown in **Fig. 4**. The polygon EFGH is a possible location area of the node $U1$ to be located, and the circle in the figure is the virtual communication circle of the virtual anchor node $U2$. In the same way, as shown in **Fig. 3**, $U2$ is used to segment the possible location area of $U1$, and the shaded portion in **Fig. 4** is the updated possible location area of the node $U1$ to be located.

After the virtual communication circle is utilized, the positioning node is segmented by using the inevitable communication circle of the neighbor node. **Fig. 5** is a schematic diagram of the inevitable communication circle assisted positioning using the anchor node neighbor. It is assumed here that $U1$ is the two-hop neighbor of $A2$, the polygon CLEFIKJ is the current possible location area of the node $U1$ to be located, the circle on the left is the inevitable communication circle of the anchor node $A2$, and the circle on the right is the circumcircle of the possible location area of $U1$. As shown in the figure, the two circles intersect at two

intersection points, and the two intersection points are connected to obtain a straight line that cuts the possible position area of $U1$ into two parts. It should be known that the node $U1$ to be located is not located in the inevitable communication circle of $A2$, because if it is within the inevitable communication circle, $U1$ is a one-hop neighbor node of $A2$, which does not conform to the assumption. Therefore, the node $U1$ can segment the possible location area within the inevitable communication circle of $A2$, and the shaded portion in **Fig. 5** is the possible location area after the $U1$ update. Similarly, the use of the virtual communication node neighbor's inevitable communication circle to treat the position node is the same and will not be described in details here. After the node to be located obtains the final possible location area, as a filtering condition of the filtering phase, the samples that do not meet the conditions are removed. If the number of filtered samples is less than n, re-sampling is performed until the number satisfies n. At the same time, the average weighting calculation is performed on all the samples, and the coordinates of the node to be located are estimated.

5. Simulation and Analysis

This paper compared the various performances of MCL, MCB, WMCL and DSLA algorithms under the same network parameter settings, in order to compare the fairness of the different algorithms. Assume that 100 (quantity range $[50-150]$) anchor nodes and 640 (quantity range $[400-1300]$) unknown nodes are randomly distributed in a sensor network region of 1000m*1000m. Both the anchor and unknown nodes adopted the Random Path Point Motion Model (RWPM) and moved randomly within the network area. The moving node determines a target position according to the movement model in the network area and randomly selects a speed among $[v_{min}, v_{max}]$. After reaching the target position, the node pauses for a period of time t_{pause}. In the simulation, the pause time t_{pause} is set to 0 if the node is always in motion. The maximum communication radius of a node is set to 100m, and the communication DOI varies from $[0-0.3]$. The simulation results of these algorithms are the average value calculated after 20 simulations.

5.1 Performance Evaluation Parameters

1. Normalization average localization error:
It represents the average value of the average positioning error generated by the positioning algorithm in a certain scenario in the simulation (for the convenience of description, it is replaced by "average error"). In the simulation, this parameter is used to evaluate the positioning effect of various algorithms; the specific formula is as follows:

$$AError = \frac{\sum_{i=1}^{M}\sqrt{(x_{N_i} - \overline{x}_{N_i})^2 + (y_{N_i} - \overline{y}_{N_i})^2}}{\sum_{i=1}^{M} R_{max}^{N_i}}$$

(7)

$$AverageError = \frac{\sum_{t=1}^{T} AError_t}{T}$$

(8)

AverageError expresses the ratio of the error sum to the radii sum. The error sum is between the estimated coordinate and actual coordinate of all unknown nodes. The radii sum is the

maximum communication radiation radii of all unknown nodes. $(\overline{x}_{N_i}, \overline{y}_{N_i})$ express the estimated coordinate position of the unknown node; (x_{N_i}, y_{N_i}) is the actual position; $R_{\max}^{N_i}$ expresses the maximum communication radiation radius; *AError* indicates the average positioning error of the positioning algorithm running in a certain scenario; and T indicates the total number of times that the positioning algorithm runs in a certain scenario.

2. Re-sampling rate:

It expresses the ratio of the number of unknown nodes whose re-sample times are greater than 10000 to the number of total unknown nodes.

3. Average Calculation time:

This is the average time taken by all the nodes to execute one-time position at a time slot and obtain estimated coordinate.

4. Localization coverage ratio:

There are two cases in the algorithm when the unknown node cannot be located. First, it is difficult to obtain the estimated position of the unknown node in the algorithm that is at the edge of the network or does not receive enough information from the anchor node. The second case is when the sampling area is too large. If the number of re-sampling is more than 10000, the sample size of the system N is still not enough. At this time, "localization coverage" is defined as the ratio of the number of unknown nodes in the network that can be located and the obtained estimated position coordinates to the total number of unknown nodes in the entire network. The localization coverage rate, CoverRate , is expressed by the following specific calculation formula:

$$\text{CoverRate} = \frac{M - m}{M} \times 100\%$$

(9)

M represents the number of unknown nodes in the network, and m represents the number of unknown nodes that have not been successfully located.

5.2 Performance Analysis

1. Speed and angle parameter selection:

In the DSLA algorithm, the size and accuracy of the sampling area is related to the final positioning error. The algorithm needs to confirm a speed parameter Δv and an angle parameter $\Delta \alpha$, so that the positioning error of the algorithm is the lowest.
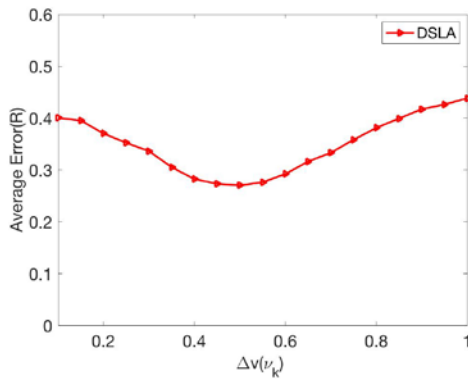

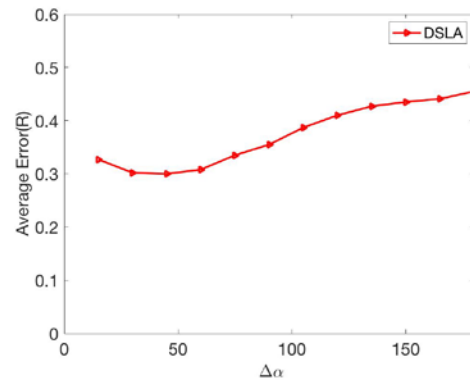**Fig. 6.** Impact of the speed parameter   **Fig. 7.** Impact of angle parameter

In the simulation experiment, since the error of Newton's interpolation computing meets the normal distribution, the angle parameter $\Delta\alpha$ is set to the random value in $[\pi/9, \pi/3]$, and the positioning error is observed with the speed parameter Δv , as shown in **Fig. 6**. After the speed parameter is determined, the case where the positioning error changes in the angle parameter will be compared. At this time, the experimental parameters use the default settings at the beginning of this section, and the speed parameters are set to $0.5v_k$. The change in the positioning error is shown in **Fig. 7**. According to the figure, $\Delta\alpha$ corresponds to less error at around 40 degrees. In summary, the speed parameter is set to $0.5v_k$ in the DSLA algorithm, and the angle parameter is $2\pi/9$ (40 degrees).

2. Performance analysis:

The performance analysis mainly analyzes the re-sampling rate, computing time, total coverage, positioning error and communication energy consumption of these algorithms. The specific analysis is as follows:
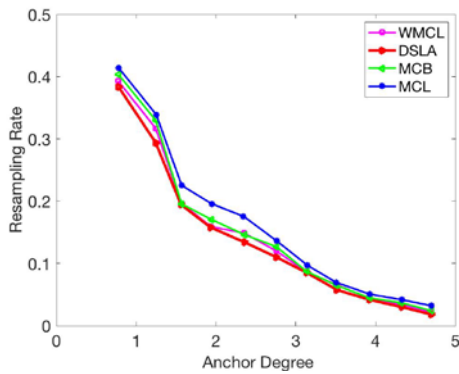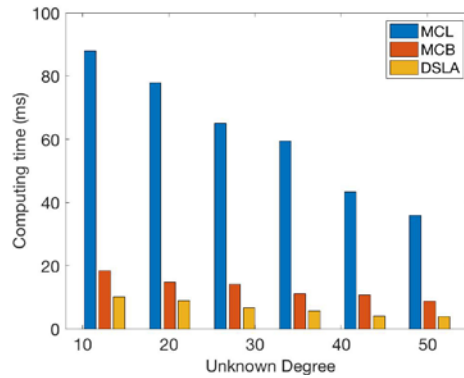

Fig. 8. Impact of the anchor on re-sampling

Fig. 9. Impact of the unknown nodes on average positioning time

Fig. 8 shows the change in re-sampling rate with change in anchor node density. It can be seen that the re-sampling rate of these algorithms decreases with increase in the density of the anchor node. The re-sampling rate decreases before the density of the anchor node is 3.5. Obviously, after the density of the anchor node exceeds 3.5, the change in the re-sampling rate is relatively flat. Because the number of anchor nodes increases, the information received by the node to be located is more accurate, and more samples in the sample set meet the filtering conditions. Also, the number of re-sampling decreases. Although the re-sampling rates of these algorithms are not much different, the DSLA algorithm is at a minimum. This is because the filtering phase in the DSLA algorithm utilizes the information of the multi-hop anchor node neighbor and the unknown node neighbor, and the obtained filtering condition is stricter than other algorithms. The valid samples that meet the conditions in the sample set obtained by each sampling are more than others. At the same time, the actual number of samples in the DSLA algorithm varies with the size of the sampling area, reducing many unnecessary re-sampling, so the re-sampling rate is low. The WMCL algorithm only reduces the sampling area in the sampling phase. The number of valid samples in the filtering phase is less than that of the DSLA algorithm, so more re-sampling is needed.

Fig. 9 shows the average positioning time as a function of the density of the unknown nodes. It can be seen that when the density of the unknown nodes increases, the computing time of several algorithms decreases to varying degrees. In **Fig. 9**, the MCL algorithm has the longest computing time, and the DSLA algorithm has the shortest computing time. The contrasting

effect is obvious. This is because, compared with other algorithms, the DSLA algorithm obtains a smaller sampling area after the prediction phase, and uses the information of the anchor node and the virtual anchor node neighbor in the filtering phase to obtain a more accurate range of possible positions of the node to be located, in addition to effectively filtering the sample for quick positioning. The two parameters of the algorithm's re-sampling rate and computing time reflect the sampling efficiency. As shown in **Fig. 8** and **Fig. 9**, the comparison of the computing time of several algorithms shows the sampling efficiency when the sampled values are not much different. In contrast, as shown in **Fig. 9**, the computing time of the DSLA algorithm is significantly lower than other algorithms, indicating that the sampling efficiency of the DSLA algorithm is higher than other algorithms.
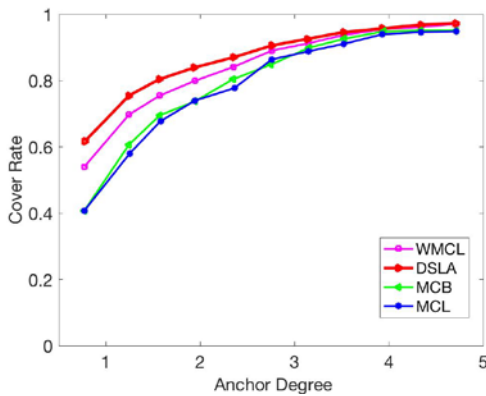

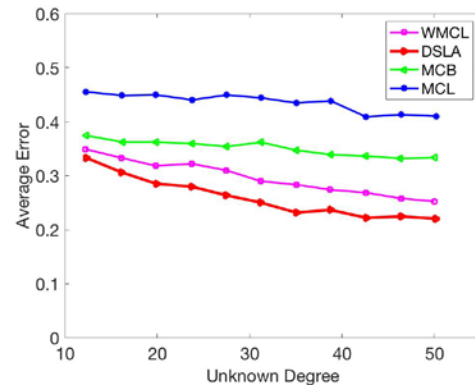**Fig. 10.** Impact of the anchor node on the coverage rate

Fig. 11. Impact of the unknown node density on average localization error

Fig. 10 shows the change in location coverage with change in the density of the anchor node, where the density of the unknown node remains at 20. It can be seen from the figure that these algorithms increase the positioning coverage rate when the number of anchor nodes increases. Among them, the coverage rate of DSLA algorithm is the highest; WMCL is second, and the coverage rate difference between MCL and MCB algorithms is low. The MCL and MCB algorithms only use the information of the one-hop and two-hop anchor nodes; where the anchor node density is low, some nodes to be located cannot receive the information of the neighbors of the one-hop or two-hop anchor nodes, which makes it impossible to locate. In the WMCL algorithm, the error of the position estimation of the unknown node is used to assist the determination of the sampling area, thereby increasing the number of positionable nodes and improving the positioning coverage. The DSLA algorithm uses the information of multi-hop neighbors in the positioning process. Even if the number of anchor nodes is small or the number of one-hop and two-hop neighbors is insufficient, the multi-hop neighbor nodes can be used to assist the positioning. Therefore, in theory, the unknown nodes in the network can be located.

In most Monte Carlo-based localization algorithms, these nodes usually transmit information to other nodes as neighbors of the other nodes. Therefore, when the unknown node density increases, the information of the anchor nodes (including one-hop and two-hop anchor nodes) that the unknown node can receive will increase, and the constraints that can be utilized when the unknown node constructs the sampling area and determines the filtering conditions increase, thereby increasing the accuracy. As shown in **Fig. 11**, as the number of unknown nodes increase, the average positioning error of these algorithms decreases. As the number of unknown nodes increases, the node distribution is relatively uniform, and the

connectivity between the nodes tends to be stable. Also, the average positioning error changes relatively smoothly. Among these algorithms, the average positioning error of the DSLA algorithm is the smallest. With increasing of the density of the anchor node, the DSLA algorithm is reduced by about 46% compared with the MCL algorithm and is reduced by about 33% compared with the MCB algorithm. It is seen that the average positioning error of the DSLA algorithm is greatly improved.
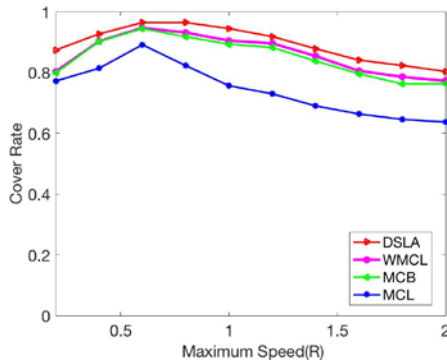

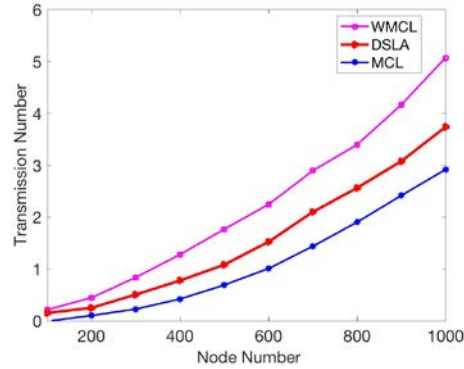**Fig. 12.** Impact of maximum motion speed on coverage rate

Fig. 13. Comparison of the communication consumption

Fig. 12 shows the positioning coverage as a function of the maximum speed of the node. It can be seen that when $v_{max} < 0.6R$, the coverage rate of the network gradually increases with the increase of the node's motion speed. When $v_{max} > 0.6R$, the coverage rate becomes smaller as the node speed increases. When $v_{max} = 0.6R$, the coverage of several algorithms is at the highest value. The DSLA algorithm has the highest positioning coverage, and the MCL algorithm has the lowest coverage. This is because in the MCL algorithm, the sampling area depends solely on the maximum motion speed of the node. When the speed increases, the size of the sampling area increases, the sampling rate decreases, and more samples do not meet the condition; therefore, the node to be located cannot be located. The MCB algorithm uses the information of the one-hop and two-hop anchor nodes in the sampling phase and combines the maximum motion speed of the node to determine the sampling area. Therefore, the motion speed of the node has little influence on the positioning of the algorithm. The DSLA algorithm estimates the current motion speed of the node. The maximum motion speed has little influence on the positioning of the algorithm, so the location coverage is the highest.

Since the energy carried by the node is limited, the energy consumption of the algorithm needs to be analyzed. As showed in **Fig. 13**, when the number of nodes in the network increases and the nodes to be located increases, the traffic of the network also increases. Moreover, the communication overhead of the DSLA algorithm is larger than that of the MCL algorithm; this is because the MCL algorithm only uses the one-hop and two-hop anchor node information, and the information sent by each node only needs to be forwarded twice to confirm the neighbor nodes in all two hops. Compared with the MCL algorithm, the DSLA algorithm utilizes multi-hop node neighbors in the filtering phase. The information sent by each node needs to be forwarded multiple times to determine the hop counts of its neighbor nodes from its own. The WMCL algorithm utilizes the position error of the node neighbors of the node to be located in the sampling phase. In the process of determining the position error, it is necessary to send information to all neighbor nodes several times, so the communication overhead of the network is larger than that of the DSLA algorithm.

6. Conclusion

In this paper, we proposed a DSLA algorithm used for node localization in the dynamic sensor network. The improvements include the determination of the sampling area, sampling number and filtering condition. After determining the sampling area and sampling number, we made full usage of the unknown node in the preliminary possible localization area as the filtering condition. The experimental results show that DSLA algorithm effectively shortens positioning time and improves positioning accuracy. Because of the availability of many other mobility models, analyzing the effect of the proposed algorithm in other mobility models is suggested for the future works.

Acknowledgement

References

[1] L. Taponecco, A. D'amico and U., Mengali, "Joint TOA and AOA estimation for UWB localization," *IEEE Transaction on Wireless Communications*, vol. 10, no. 7, pp. 2207-2217, 2011. Article (CrossRef Link)

[2] B. Xu, G. Sun and R. Yu, "High-Accuracy TDOA-Based Localization without Time Synchronization," *IEEE Transaction on Parallel and Distributed Systems*, vol. 24, no. 8, pp. 1567-1576, 2011. Article (CrossRef Link).

[3] P. Kulakowski, J. Vales-Alonso, E. Egea-Lopez, W. Ludwin and J. García-Haro, "Angle-of-arrival localization based on antenna arrays for wireless sensor networks," *Computers & Electrical Engineering*, vol. 36, no. 6, pp. 1181-1186, 2010. Article (CrossRef Link).

[4] G. Adewumi, K. Djouani and M. Kurien, "RSSI based indoor and outdoor distance estimation for localization in WSN," *IEEE Transaction on Parallel and Distributed Systems*, pp. 1534-1539, 2013. Article (CrossRef Link).

[5] X. Liu, S. Zhang and K. Bu, "A locality-based range-free localization algorithm for anisotropic wireless sensor networks," *Telecommunication Systems,* vol. 62, no. 1, pp. 3-13, 2016. Article (CrossRef Link).

[6] F. Shahzad, T. Shaltami and E. Shakshukhi, "DV-maxHop: A Fast and Accurate Range-Free Localization Algorithm for Anisotropic Wireless Networks," *IEEE Transactions on Mobile Computing,* vol. 16, no. 9, pp. 2494-2505, 2017. Article (CrossRef Link).

[7] J. Liu, Z. Wang, M. Yao and Z. Qiu, "VN-APIT: virtual nodes-based range-free APIT localization scheme for WSN," *Wireless Networks,* vol. 22, no. 3, pp. 867-878, 2016. Article (CrossRef Link).

[8] S. Liang, L. Liao and Y. Lee, "Localization Algorithm based on Improved Weighted Centroid in Wireless Sensor Networks," *Journal of Networks*, pp. 1534-1539, 2013.

[9] C. Fu, Z. Qian and G. Ji, "An Improved DV-Hop Localization Algorithm in Wireless Sensor Network," in *Proc. of 2013 International Conference on Information Technology and Applications*, pp.13-16, 2013. Article (CrossRef Link).

[10] Y. Liu, J. Chen and Z. Xu, "Improved DV-Hop Localization Algorithm Based on Bat Algorithm in Wireless Sensor Networks," *KSII Transactions on Internet and Information Systems,* vol. 11, no. 1, pp. 215-236, 2017. Article (CrossRef Link).

[11] W. Wang, Y. Yang, L. Wang and W. Lu, "PMDV-hop: An effective range-free 3D localization scheme based on the particle swarm optimization in wireless sensor network," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 1, pp. 61-80, 2018. Article (CrossRef Link).

[12] S. Tomic, M. Beko and R. Dinis, "Distributed RSS-based localization in wireless sensor networks using convex relaxation," in *Proc. of 2014 International Conference on Computing, Networking and Communications,* pp. 853-857, 2014. Article (CrossRef Link).

[13] S. Singh and S. Sharma, "Range Free Localization Techniques in Wireless Sensor Networks: A Review," *Procedia Computer Science*, vol. 57, pp. 7-16, 2015. Article (CrossRef Link).

[14] G. Han, J. Jiang, C. Zhang and et al, "A Survey on Mobile Anchor Node Assisted Localization in Wireless Sensor Networks," *IEEE Communications Surveys & Tutorials,* vol. 18, no. 3, pp. 2220-2243, 2016. Article (CrossRef Link).

[15] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. of the 10th International Conference on Mobile Computing and Networking*, pp. 45-57, 2004. Article (CrossRef Link).

[16] A. Baggio and K. Langendoen, "Monte-Carlo localization for mobile wireless sensor networks," *Journal of Ad Hoc Networks*, vol. 6, no. 5, pp. 718-733, 2008. Article (CrossRef Link).

[17] S. Enrique, V. Vijayanth and W. Vincent, "Dual and Mixture Monte Carlo Localization Algorithms for Mobile Wireless Sensor Networks," in *Proc. of IEEE WCNC2007*, pp. 4024-4025, 2007. Article (CrossRef Link).

[18] H. Mirebrahim and M. Dehghanand, "Monte Carlo Localization of Mobile Sensor Networks Using the Position Information of Neighbor Nodes," in *Proc. of 8th International Conference on Ad-Hoc, Mobile and Wireless Networks*, pp. 270-283, 2009.

[19] M. Rudafshani and S. Datta, "Localization in Wireless Sensor Networks," in *Proc. of 6th International Symposium on Information Processing in Sensor Networks*, pp. 51-60, 2007. Article (CrossRef Link).

[20] S. Zhang, J. Cao and C. Li, "Accurate and energy-efficient range-free localization for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 897-910, 2010. Article (CrossRef Link).

[21] S. Simic and S. Sastry, "Distributed localization in wireless adhoc networks," *Journal of Ad Hoc Networks*, 2001.

[22] B. Dil, S. Dulman and P. Havinga, "Range-based localization in mobile sensor networks," *Wireless Sensor Networks*, pp. 164-179, 2006.

[23] W. Wang and Q. Zhu, "RSS-based Monte Carlo localisation for mobile sensor networks," *Communications,* vol. 2, no. 5, pp. 673-681, 2008. Article (CrossRef Link).

[24] M. Martins, H. Chen and K. Sezaki, "OTMCL: Orientation tracking-based Monte Carlo localization for mobile sensor networks," in *Proc. of the 6th International Conference on Networked Sensing Systems (INSS),* pp. 1-8, 2009. Article (CrossRef Link).

[25] T. Camp, J. Boleng and J. Davies, "A survey of mobility models for ad hoc network research," *Wireless communications and mobile computing,* vol. 2, no. 5, pp. 483-502, 2002. Article (CrossRef Link).

[26] L. Chen, M. Li, K. Liu, J. Zhang, T. Peng, Y. Liu, Y. Xu, Q. Luo and T. He, "Distributed range-free localisation algorithm for wireless sensor networks," *ELECTRONICS LETTERS,* vol. 50, no. 12, pp. 894-896, 2014. Article (CrossRef Link).

[27] M. Shon, M. Jo and H. Choo, "An interactive cluster-based MDS localization scheme for multimedia information in wireless sensor networks," *Computer Communications*, vol. 35, no. 15, pp. 1921-1929, 2012. Article (CrossRef Link).

[28] T. Vinh, N. Thu and M. Jo, "A Lateration-localizing Algorithm for Energy-efficient Target Tracking in Wireless Sensor Networks," *Ad Hoc & Sensor Wireless Networks,* vol. 34, no. 1-4, pp. 191-220, 2016.

Yanru Chen holds a Master's degree of Management (Finance) from the University of Melbourne, Australia. Now, she is studying for her Ph.D. degree in the Embedded Systems and Big Data Management Laboratory and the Internet of Things Laboratory of the School of Computer Science from Sichuan University under the guidance of Professor Bing Guo. Her research interests include wireless sensor network, cognitive psychology, personal financial big data, computational finance, and financial data space, etc.

Bingshu Yan holds a Master's degree from Sichuan University, Chengdu, China. Her research interests include Wireless Sensor Network, Embedded Systems and Computer Network.

Liangxiong Wei is studying for his PhD at School of Computer Science, Sichuan University, Chengdu, China. His research interests include Wireless Sensor Network and Cloud Computing.

Min Guo is studying for her PhD at School of Computer Science, Sichuan University, Chengdu, China. Her current research interests include Industrial Internet and Indoor localization.

Feng Yin holds a PhD degree from Sichuan University, Chengdu, China. He is currently a professor at the School of Computer Science and Technology, Southwest Minzu University. His research interests include data mining, middleware, distributed computing and software test.

Qian Luo holds a PhD degree from Sichuan University, Chengdu, China. He is currently working at the Second Research Institute of Civil Aviation Administration of China (CAAC). He is an expert in the field of information technology and has accomplished several key national projects on aviation area. He is a researcher and has published many academic papers.

Wei Wang is studying for his PhD at School of Computer Science, Sichuan University, Chengdu, China. His research interests include edge computing, indoor localization and situation forecasting.

Liangyin Chen holds a PhD degree from the School of Computer Science from Sichuan University, in 2008. He is currently a professor at the School of Computer Science, Sichuan University. He was a visiting researcher at the University of Minnesota from August 2009 to September 2010 under the guidance of Professor Tian He. He is the author and co-author of more than 30 papers, many of which were published in premier network journals and conferences. He received the award of SCU-Lixin Tang distinguished teacher in 2012. His research interests include wireless sensor network, embedded systems, computer network, distributed systems, big data analysis, natural language processing, Industrial Internet and Internet of Thing.