Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks — Source link ↗

Linqing Gui, Thierry Val, Anne Wei, Rejane Dalce

Institutions: University of Toulouse, Conservatoire national des arts et métiers

Related papers:

- DV Based Positioning in Ad Hoc Networks

- GPS-less low-cost outdoor localization for very small devices

- Range-free localization schemes for large scale sensor networks

- Two novel DV-Hop localization algorithms for randomly deployed wireless sensor networks

- Improvements of DV-Hop localization algorithm for wireless sensor networks

Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks

Linqing Gui, Thierry Val, Anne Wei, Réjane Dalce

Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks

Linqing Gui [a,b,*], Thierry Val [b], Anne Wei [c], Réjane Dalce [b]

[a] Nanjing University of Science and Technology, 200 Xiaolingwei Street, 210094 Nanjing, China
[b] University of Toulouse, IRIT, 1 Place Georges Brassens, 31703 Toulouse, France
[c] CNAM, CEDRIC, 292 rue St Martin, 75003 Paris, France

ABSTRACT

Localization is a fundamental issue for many applications in wireless sensor networks. Without the need of additional ranging devices, the range-free localization technology is a cost-effective solution for low-cost indoor and outdoor wireless sensor networks. Among range-free algorithms, DV-hop (Distance Vector-hop) has the advantage to localize the mobile nodes which has less than three neighbour anchors. Based on the original DV-hop algorithm, this paper presents two improved algorithms (Checkout DV-hop and Selective 3-Anchor DV-hop). Checkout DV-hop algorithm estimates the mobile node position by using the nearest anchor, while Selective 3-Anchor DV-hop algorithm chooses the best 3 anchors to improve localization accuracy. Then, in order to implement these DV-hop based algorithms in network scenarios, a novel DV-hop localization protocol is proposed. This new protocol is presented in detail in this paper, including the format of data payloads, the improved collision reduction method E-CSMA/CA, as well as parameters used in deciding the end of each DV-hop step. Finally, using our localization protocol, we investigate the performance of typical DV-hop based algorithms in terms of localization accuracy, mobility, synchronization and overhead. Simulation results prove that Selective 3-Anchor DV-hop algorithm offers the best performance compared to Checkout DV-hop and the original DV-hop algorithm.

1. Introduction

In recent years, wireless sensor networks have attracted worldwide research and industrial interest. They are typically composed of resource-constrained sensor nodes which can communicate with each other and cooperatively collect information from the environment. Wireless sensor networks can be deployed in various applications. For example, they can serve for parking space detection [1], security surveillance [2], indoor object tracking [3,4], or monitoring services [5–7]. It is crucial for sensor data to be combined with position information in many applications [1,3–5]. The position of sensors can also help to facilitate routing as well as determining the quality of coverage and achieving load balancing. Therefore, localization has become a fundamental element in wireless sensor networks study [1–9].

The existing localization techniques can be generally categorized into two types: range-based and range-free. Range-based schemes [10–14] need first precisely measure the range information (the distance or the angle) between concerned equipments, and then calculate the desired position based on trilateration or triangulation approaches. The ranging methods typically use Received Signal

Strength Indicator (RSSI) [11], Time of Arrival (TOA) [12], Time Difference of Arrival (TDOA) [13], and Angle of Arrival (AOA) [14]. GPS (Global Positioning System) [10] is the most well-known range-based technique using TOA or TDOA. However, the GPS devices not only consume lots of energy, but also fail to work indoors. An alternative technique is GSM (Global System for Mobile communications), using RSSI and AOA methods. Note that GPS and GSM support localization by using complex and expensive systems. Another technology is UWB (Ultra Wide Band) which can be used to measure time of flight with high precision [15]. The range-based techniques have two major drawbacks. First, the range information is very easily affected by multipath fading, noise and environment variations. Second, usually, additional ranging devices are needed, which consume more energy and increase the overall cost.

While the range-based scheme uses the distance or angle between nodes, the range-free scheme uses connectivity information between nodes. In this scheme, the nodes that are aware of their positions are called anchors, while others are called normal nodes. Anchors are fixed, while normal nodes are usually mobile. Normal nodes first gather the connectivity information as well as the positions of anchors, and then calculate their own positions. Since no ranging information is needed, the range-free scheme can be implemented on low-cost wireless sensor networks. Another advantage of range-free scheme is its robustness; the connectivity information between nodes is not easily affected by the environment. As a result, we focus our research on the range-free scheme.

The typical range-free algorithms include Centroid [16], CPE (Convex Position Estimation) [17], and DV-hop (Distance Vector-hop) [18]. Centroid and CPE are simple, having low complexity, but they require a normal node to have at least three neighbouring anchors. DV-hop algorithm can handle the case where a normal node has less than three neighbour anchors. Considering this interesting advantage of the DV-hop algorithm, we focus this paper on DV-hop based localization algorithms.

Many algorithms based on DV-hop have been proposed these past years [19–21]. In [19], the author proposes a DDV-hop (Differential DV-hop) algorithm, using an average distance per hop to estimate the mobile node's position. Unlike the original DV-hop algorithm, this new average distance per hop is calculated based on the differential error of each anchor's distance-per-hop. In [20], a self-adaptive DV-hop algorithm is proposed, which obtains a weighted average distance per hop for each normal node based on its hop counts to anchors. The work in [21] presents a robust DV-hop algorithm, where each normal node calculates a weighted average distance per hop based on its topology relationship with any two anchors. However, these algorithms have not provided sufficient accuracy. In order to improve localization performance, we have introduced two new algorithms: Checkout DV-hop and Selective 3-Anchor DV-hop [22,23]. In this paper, we present these algorithms in detail.

During the verification process of our two new algorithms, we noted that most of the existing algorithms were only studied using tools like MATLAB which neglect the possible problems of a real network. In fact, since the

principle of DV-hop based algorithms is the broadcast of position related information through the network, some problems such as collisions and link congestion must be solved by a localization protocol. Having found no such protocol, we propose in this paper a DV-hop localization protocol in real network scenarios. This protocol is based on the IEEE 802.15.4 standard, with the chosen medium access method being non-slotted CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). The network topology is assumed as ad-hoc.

In the following, we list three main contributions of this paper.

(1) We introduce our two improved algorithms Checkout DV-hop and Selective 3-Anchor DV-hop. Checkout DV-hop adjusts the position of a normal node based on its distance to the nearest anchor, while Selective 3-Anchor DV-hop chooses the best 3 anchors based on connectivity parameters.
(2) We present a new DV-hop localization protocol. The new protocol covers the format of data payload, the improved collision reduction method E-CSMA/CA, several parameters for the end of each DV-hop step, and the complete frame exchange procedure. Note that our protocol can be used in both synchronized and unsynchronized networks.
(3) Based on our DV-hop protocol, we simulate original DV-hop, Checkout DV-hop and Selective 3-Anchor DV-hop by using the network simulator WSNet [24,25]. The comparative network simulation results are presented and analyzed in terms of accuracy, overhead, mobility and synchronization.

The rest of this paper is organized as follows. Section 2 introduces a few typical DV-hop based algorithms, as well as our two improved algorithms (Checkout DV-hop and Selective 3-Anchor DV-hop). Section 3 presents our new protocol for the implementation of the DV-hop algorithm. In Section 4, the simulation results and analysis are given and localization performances are discussed. Finally we give our conclusion and prospects in Section 5.

2. DV-hop based algorithms

In this section, we first introduce the original DV-hop as well as some typical DV-hop based localization algorithms. Then, we present our two algorithms, Checkout DV-hop and Selective 3-Anchor DV-hop.

2.1. The original DV-hop algorithm

The DV-hop localization algorithm was proposed by Niculescu [18]. It is a suitable solution for normal nodes having three or less neighbour anchors. As shown in Fig. 1, although the normal node Nx has only one neighbour or reachable anchor $A1$, Nx can use the DV-hop algorithm for localization. The algorithm consists of the following three steps.

First, each anchor A_i broadcasts through the network a message containing the position of A_i and a hop count field
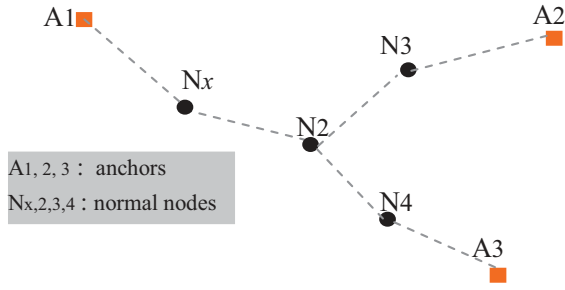
Fig. 1. Example of DV-hop.

set to 0. This hop count value will increase with each hop during the broadcast of the message. This means, as soon as this message is received by a node, the hop count value in the message will be incremented. On the first reception of the message, every node N (either anchor or normal node) records the position of A_i and initializes hop_i as the hop count value in the message. Here, hop_i is the minimum hop count between N and A_i. If the same message is received again, N maintains hop_i: if the received message contains a lower hop count value than hop_i, N will update hop_i with that lower hop count value and relay the message; otherwise, N will ignore the message. Through this mechanism, all nodes can get the minimum hop count to each anchor.

Second, when an anchor A_i receives the positions of other anchors as well as the minimum hop counts to other anchors, A_i can calculate its average distance per hop, denoted as dph_i. The detailed calculation of dph_i can be found in [18]. Once dph_i is calculated, it will be broadcasted by A_i.

Third, when receiving dph_i, the normal node Nx multiplies $hop_{i,Nx}$ (its hop count to A_i) by dph_i, so that Nx obtains its distance to each anchor A_i, denoted as $d_{i,Nx}$. Here, $i \in \{1,2,\dots m\}$, if we assume that there are totally m anchors. Then each normal node Nx can calculate its estimated position $N_{DV\text{-}hop}$ by trilateration. The detail of the calculations of $N_{DV\text{-}hop}$ can be found in [18].

Although DV-hop algorithm can localize the normal nodes with less than 3 neighbour anchors, there is still much room for improvement regarding its localization accuracy. Thus, many algorithms have been proposed in recent years. In the following, several typical algorithms will be analyzed.

2.2. Typical DV-hop Based Algorithms

In this section we describe a few DV-hop based localization algorithms such as DDV-hop (Differential DV-hop), Self-adaptive DV-hop, and Robust DV-hop.

(i) *DDV-hop:* in [19], the author proposes a DDV-hop (Differential DV-hop) algorithm. This algorithm changes Step 2 and Step 3 of the original DV-hop algorithm. In Step 2 of DDV-hop, each anchor A_i not only broadcasts its distance-per-hop dph_i through the network, but also broadcasts the differential error of dph_i to the entire network. The

definition and calculation of this differential error can be found in [19]. In Step 3, DDV-hop and DV-hop differ on the calculation of the estimated distance between a normal node Nx and each anchor A_i. In the original DV-hop algorithm, when a normal node Nx receives the distance-per-hop value of A_i, Nx immediately calculates its estimated distance to A_i as $dph_i \times hop_{i,Nx}$. But in DDV-hop algorithm, Nx uses its own distance-per-hop value denoted as dph_{Nx} to replace the anchors' distance-per-hop. dph_{Nx} is obtained as the weighted sum of all anchors' distance-per-hop. The weighting coefficients are decided by the differential error of anchors' distance-per-hop. The details on the calculation of dph_{Nx} can be found in [19].

(ii) *Self-Adaptive DV-hop:* in [20], a DV-hop based Self-Adaptive Positioning algorithm is proposed. This algorithm is composed of two methods. Since the second method requires RSSI information, we only consider the first method of this self-adaptive algorithm. This algorithm has the same network overhead as the original DV-hop but slightly changes Step #3. That is, when a normal node Nx calculates its estimated distance to A_i, Nx also uses its own distance-per-hop value denoted as dph_{adp} to replace the anchors' distance-per-hop. dph_{adp} is also obtained as the weighted sum of anchors' distance-per-hop. But compared to DDV-hop algorithm, this self-adaptive algorithm has a different way to decide the weighing coefficients for dph_{adp}. In this algorithm, when calculating dph_{adp}, the weighting coefficient of dph_i (each anchor A_i's distance-per-hop) is decided based on Nx's hop counts to A_i. The more hops between Nx and A_i, the smaller the value assigned to the weighting coefficient of dph_i. The details on the calculation of dph_{Nx} can be found in [20].

(iii) *Robust DV-hop:* a Robust DV-hop (RDV-hop) algorithm is proposed in [21]. Similar to the above two algorithms, RDV-hop uses a weighted distance-per-hop value dph_{rdv} for each normal node Nx. However, this time, dph_{rdv} is not the weighted sum of each anchor's distance-per-hop, but is the weighted sum of the distance-per-hop values between any two anchors. In the calculation of dph_{rdv}, the weighing coefficient of $dph_{i,k}$ (distance-per-hop between two anchors A_i and A_k) will have the maximum value, if Nx is one node on the shortest path between A_i and A_k. The details of the calculation of the weighting coefficients and dph_{Nx} can be found in [21].

All these typical DV-hop based algorithms use a weighting method to determine a weighted distance-per-hop value for each normal node. However, in order to get a more accurate weighted distance-per-hop value, sometimes additional information is necessary, such as differential error in [19], or network topology in [21]. Broadcasting this additional information increases the network traffic. We should also note that, the simulation results of the above algorithms are not so convincing, because the distributions of sensor nodes are particularly designed rather than randomly obtained. For example, in [20], the

simulation scenario is very special: anchors are distributed at the corners of the simulation area, while normal nodes are regularly distributed inside the area. In order to obtain a better accuracy without increasing the network overhead, we will present in the following our two algorithms, Checkout DV-hop and Selective 3-Anchor DV-hop.

2.3. Our Checkout DV-hop algorithm

In order to improve localization accuracy, we have proposed Checkout DV-hop algorithm. Making best use of the nearest anchor, Checkout DV-hop adds a low-complexity calculation step to DV-hop. In the following, we will introduce its principle in detail.

The key issue of DV-hop is calculating the approximate distance between the normal node Nx and each anchor Ai, by multiplying the hop count by the average distance per hop. This means:

$$d_{i,Nx} = hop_{i,Nx} \times dph_i, i = 1, 2 \ldots m \quad (1)$$

where $d_{i,Nx}$ is the approximate distance between N_x and A_i, $hop_{i,Nx}$ is the minimal *hop* number between N_x and A_i, and dph_i is the approximate average distance per hop of A_i. The calculation of dph_i is shown as:

$$dph_i = \left(\sum_{k(k \neq i)} d_{i,k} \right) \Big/ \left(\sum_{k(k \neq i)} hop_{i,k} \right) \quad (2)$$

where $d_{i,k}$ is the distance between A_i and A_k, $hop_{i,k}$ is the minimal hop count between A_i and A_k.

Since $d_{i,Nx}$ is an important element for calculating the position of the normal node N_x [18], it has a considerable influence on the accuracy of DV-hop. We denote the true distance from N_x to A_i as $d_{i,NxTrue}$, and the difference between $d_{i,NxTrue}$ and $d_{i,Nx}$ as $\Delta d_{i,Nx}$, where obviously $\Delta d_{i,Nx}$ is one reason for the inaccuracy of DV-hop. If we denote Δdph_i as the average difference between dhp_i and its true value, then from Eq. (1), we have:

$$\Delta d_{i,Nx} = hop_{i,Nx} \times \Delta dph_i \quad (3)$$

Here, we should mention that the Eq. (3) functions in an average manner. That means, the equation may be unfit for a few special cases, for example, when Δdph_i becomes too small. The special cases will be investigated in future work.

Eq. (3) indicates that, when $hop_{i,Nx}$ increases, on average, $\Delta d_{i,Nx}$ also increases, and the accuracy of DV-hop decreases. If A_{near} is the nearest anchor to Nx among all anchors $A_1\ A_2\ \ldots\ A_m$, then correspondingly $hop_{near,Nx}$ is the smallest, so that $\Delta d_{near,Nx}$ is the smallest position error. We can conclude that, compared to other anchors, the distance from the normal node N_x to its nearest anchor A_{near}, denoted as $d_{near,Nx}$, has the highest reliability in terms of precision. Our proposed Checkout DV-hop method makes best use of this concept by correcting the results using the most reliable information available.

Now we illustrate the principle of our algorithm, which adds a checkout step to DV-hop algorithm, shown in Fig. 2. For the purpose of comparison, Fig. 2(a) shows the result of original DV-hop, while Fig. 2(b) shows our checkout step.

As shown in Fig. 2(a), the normal node Nx uses DV-hop to obtain its estimated position at N_{DV-hop} with its coordinates denoted as (x', y'). It then calculates the distance between N_{DV-hop} and its nearest anchor A_{near} (here A_{near} is A_1), denoted as d_{DV-hop}. Note that Nx has used Eq. (1) to evaluate its approximate distance to A_{near}, denoted as $d_{near,Nx}$.

The purpose of the checkout step is to change the estimated position from N_{DV-hop} (see Fig. 2(b)) to a new one called $N_{checkout}$, whose distance to A_{near} is $d_{near,Nx}$. To achieve this, the easiest and quickest way is to change the position along the line connecting N_{DV-hop} and A_{near}. $N_{checkout}$ is on the line from N_{DV-hop} to A_{near}, and the distance between $N_{checkout}$ and A_{near} is $d_{near,Nx}$. The position of A_{near} is (x_{Anear}, y_{Anear}) and N_{DV-hop} is located at (x', y'), therefore the position of $N_{checkout}$, denoted as $(x_{checkout}, y_{checkout})$ can be derived as follows. $N_{checkout}$ is chosen as our node estimated position.

$$\begin{cases} X_{checkout} = x' - \left(\frac{d_{DV-hop} - d_{near,Nx}}{d_{DV-hop}} \right) \times (x' - x_{Anear}) \\ X_{checkout} = x' - \left(\frac{d_{DV-hop} - d_{near,Nx}}{d_{DV-hop}} \right) \times (y' - y_{Anear}) \end{cases} \quad (4)$$

2.4. Our Selective 3-Anchor DV-hop algorithm

Since the accuracy improvement by Checkout DV-hop is limited [22], we have proposed Selective 3-Anchor DV-hop algorithm. First, this algorithm generates a group of candidates. Then, from this pool, it chooses one based on its connectivity vector.

In order to facilitate our presentation of this algorithm, we first introduce two basic elements: 3-anchor group and 3-anchor estimated position. Then, the principle of our algorithm is presented.

2.4.1. 3-Anchor Groups and 3-Anchor estimated positions

Let's consider a network with m anchors $A_1\ A_2\ \ldots\ A_m$. Through the first two steps of DV-hop, a normal node N_x can obtain $hop_{i,Nx}$, which is its minimum hop count to each anchor A_i, as well as $d_{i,Nx}$, which is the estimated distance between N_x and A_i. Then, N_x can calculate its estimated position N_{DV-hop} by trilateration based on the m estimated distance values $d_{1,Nx}\ d_{2,Nx}\ \ldots\ d_{m,Nx}$. So, the quality of these estimates has a great influence on the accuracy of DV-hop.

In fact, instead of using all m estimates, three estimated distance values to three different anchors are sufficient for N_x to calculate its position. For example, we use $d_{i,Nx}, d_{j,Nx}, d_{k,Nx}$, which are the three estimated distance values from N_x to the three corresponding anchors A_i, A_j, A_k. If we denote the true position of N_x as (x, y), and the positions of $A_i\ A_j\ A_k$ respectively as $(x_i, y_i), (x_j, y_j), (x_k, y_k)$, then we can write the following equations:

$$\begin{cases} (x - x_i)^2 + (y - y_i)^2 = d_{i,Nx}^2 \\ (x - x_j)^2 + (y - y_j)^2 = d_{j,Nx}^2 \\ (x - x_k)^2 + (y - y_k)^2 = d_{k,Nx}^2 \end{cases} \quad (5)$$

Solving (5) by trilateration, we can get a 3-anchor estimated position of N_x, denoted as $N_{\langle i,j,k \rangle}$ $(x_{\langle i,j,k \rangle}, y_{\langle i,j,k \rangle})$. It is calculated as:
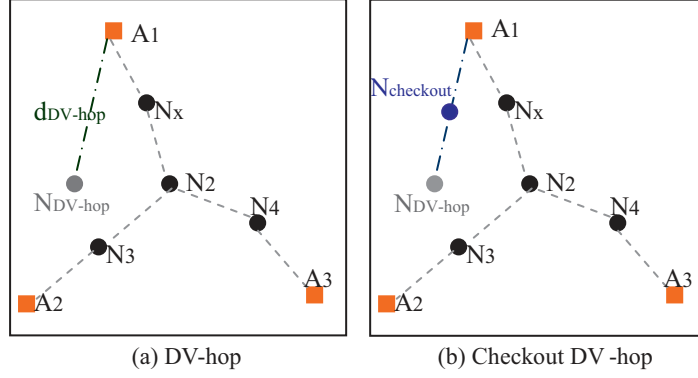
Fig. 2. Principle of Checkout DV-hop.

$$N<i,j,k>: \begin{bmatrix} x<i,j,k> \\ y<i,j,k> \end{bmatrix} = C^{-1}B, \text{ and}$$

$$C = -2 \times \begin{bmatrix} x_i - x_k & y_i - y_k \\ x_j - x_k & y_j - y_k \end{bmatrix}, \quad (6)$$

$$B = \begin{bmatrix} d_{i,Nx}^2 - d_{k,Nx}^2 - x_i^2 - y_i^2 + x_k^2 + y_k^2 \\ d_{j,Nx}^2 - d_{k,Nx}^2 - x_j^2 - y_j^2 + x_k^2 + y_k^2 \end{bmatrix}$$

where the dimension of matrix C is 2 by 2, and that of matrix B is 2 by 1. Here, it should be mentioned that the three anchors $A_i A_j A_k$ cannot be colinear. Otherwise, matrix C will be singular.

Among the m available anchors, if we select any three anchors to form a 3-anchor group, then there are in total C_m^3 groups. Using (6), based on each group, N_x can generate a 3-anchor estimated position. Totally N_x can have C_m^3 3-anchor estimated positions. They are all candidate positions for N_x.

Some 3-anchor estimated positions of N_x have much higher accuracy than $N_{DV\text{-}hop}$, and some others are not so accurate. In order to present this phenomenon, we use a typical example of network topology as shown in Fig. 3. The network occupies a 50 * 50 m^2 area, with a total of 10 nodes randomly distributed inside. The maximum communication range of all the nodes is set to 20 m. Among the
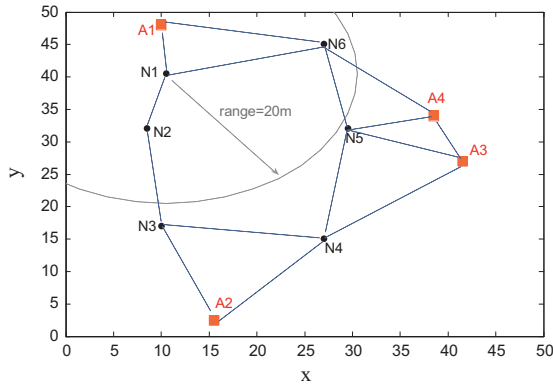

Fig. 3. Example of nodes distribution.

10 nodes, 4 are anchors $A_1 A_2 A_3 A_4$ who already know their positions. The remaining units are normal nodes $N_1 N_2 \dots N_6$. These normal nodes do not know their positions. The dashed lines indicate that the two linked nodes are in each other's communication range.

In this example, based on Group$\langle A_1, A_2, A_3 \rangle$, Group$\langle A_1, A_2, A_4 \rangle$, and Group$\langle A_2, A_3, A_4 \rangle$, N_x (which corresponds to N_1) can get its 3-anchor estimated positions respectively $N_{1\langle 1,2,3 \rangle}$, $N_{1\langle 1,2,4 \rangle}$, and $N_{1\langle 2,3,4 \rangle}$. Table 1 lists these estimated positions and their corresponding location errors. We can note that $N_{1\langle 1,2,3 \rangle}$ is much more accurate than other estimated positions.

Here, the location error is defined as the Euclidean distance between a normal node's estimated position and its real position. N_1's real position is (10.50, 40.50).

Our selective 3-anchor DV-hop algorithm will select the most accurate 3-anchor estimated position and regard it as the final estimated position.

2.4.2. Position vs. connectivity

Range-free localization schemes are based on two kinds of information: anchors' positions, and the connectivity between nodes. In DV-hop, the connectivity of N_x is specified as the minimum hop counts between N_x and each anchor. Since this paper focuses on the algorithms based on DV-hop, the connectivity mentioned in this paper will be considered as an array which contains the minimum hop counts to anchors. For example, if there are totally m anchors and the minimum hop count from N_x to each anchor A_i is hop_i, then the connectivity of N_x is the array $[hop_1, hop_2 \dots hop_m]$.

The connectivity of a normal node can identify its position. For example, from Fig. 3, the connectivity of each normal node can be observed. The results are summarized

Table 1
Examples of 3-anchor estimated positions for N_1.

3-Anchor estimated positions (m)	Location error (m)
$N_{1\langle 1,2,3 \rangle}$ (7.77, 44.82)	5.11
$N_{1\langle 1,2,4 \rangle}$ (18.44, 46.11)	9.72
$N_{1\langle 2,3,4 \rangle}$ (0, 73.92)	35.03
$N_{1\langle 1,3,4 \rangle}$ (45.90, 102.02)	70.98
DV-hop estimated position	10.23
$N_{1, \text{DV-hop}}$ (17.30, 48.14)	

Table 2
Connectivities of normal nodes.

Normal node	Connectivity
N_1	[1,3,3,2]
N_2	[1–3,3]
N_3	[2,1–3]
N_4	[3,1,1,2]
N_5	[2,2,1,1]
N_6	[1,3,2,1]

in Table 2. From this table, we can find that each normal node has a unique connectivity, which allows us to identify its position.

Since the connectivity of a normal node can represent its position, if two normal nodes have similar connectivities, then they must have similar positions. That is, they are very near to each other. Therefore, we can deduce the relationship between connectivity difference and the distance: smaller connectivity difference between two normal nodes will result in smaller distance between them.

Then, we utilize the sum of absolute difference to quantify the connectivity difference. For example, from Table 2, the connectivity difference between N_1 and N_2 can be calculated as $|1 - 1| + |3 - 2| + |3 - 3| + |2 - 3| = 2$. This small connectivity difference indicates a small distance between N_1 and N_2, which then can be observed from Fig. 3.

To give a better understanding of this concept, we investigate the relationship between N_x (N_1 in Fig. 3) and any other normal node. From Table 2, we can calculate the connectivity difference between N_1 and all the other normal nodes. The results are listed in Table 3. In this table, we also give the distance value between N_1 and any other normal node. Comparing the last two lines, we can find that larger connectivity difference always reflects the longer distance between two normal nodes. For example, the connectivity difference between N_3 and N_1 is bigger than that between N_2 and N_1. Correspondingly, N_3 is further from N_1 than N_2.

This relationship between the distance and connectivity difference can be used to find the most accurate 3-anchor estimated position. The basic principle of our selective 3-anchor DV-hop algorithm is to choose the 3-anchor estimated position which has the smallest connectivity difference to N_x.

However, the connectivity of each 3-anchor estimated position is still unknown. That is, the hop count from $N_{\langle i,j,k \rangle}$ to each anchor is unknown. We need to know how to calculate the hop count between $N_{\langle i,j,k \rangle}$ and each anchor.

2.4.3. Hop count for 3-anchor estimated position

Through the first two steps of DV-hop, N_x can obtain the anchors' positions as well as its minimum hop counts to all

Table 3
Connectivity difference and distance to N_1.

Normal node	N_2	N_6	N_5	N_3	N_4
Connectivity difference to N_1	2	2	5	5	6
Distance to N_1 (m)	8.73	17.10	20.36	23.51	30.37

anchors. Therefore, N_x can calculate the distances between $N_{\langle i,j,k \rangle}$ and each anchor A_t, denoted as $d_{\langle i,j,k \rangle,t}$. Then the problem of calculating the hop count between $N_{\langle i,j,k \rangle}$ and A_t becomes the problem of calculating the distance per hop. Because if N_x knows the distance per hop between $N_{\langle i,j,k \rangle}$ and A_t, denoted as $dph_{\langle i,j,k \rangle,t}$, then N_x can calculate the hop count between $N_{\langle i,j,k \rangle}$ and A_t as:

$$hop_{\langle i,j,k \rangle,t} = \frac{d_{\langle i,j,k \rangle,t}}{dph_{\langle i,j,k \rangle,t}} \tag{7}$$

We must then find a method to estimate the value of $dph_{\langle i,j,k \rangle,t}$. But all the distance-per-hop information that N_x has obtained are anchors' distance-per-hop values: dph_1, $dph_2, \ldots, dph_m$, including the distance per hop of A_t denoted as dph_t. Hence, we need to estimate $dph_{\langle i,j,k \rangle,t}$ based on the anchors' distance-per-hop values.

In order to get an approximate value of $dph_{\langle i,j,k \rangle,t}$, three kinds of relative positions between $N_{\langle i,j,k \rangle}$ and its nearest anchor A_{near} are considered, based on the distance between $N_{\langle i,j,k \rangle}$ and A_{near}. In the first case, the distance between $N_{\langle i,j,k \rangle}$ and A_{near} is so small that we can use the distance-per-hop value of A_{near} (denoted as dph_{near}) as an approximate value of $dph_{\langle i,j,k \rangle,t}$. Here, as an example, we can set the distance threshold as half of the radio range of nodes. Of course, the best value of the threshold can be determined by simulations. The second case is the opposite: the distance between $N_{\langle i,j,k \rangle}$ and A_{near} is so large that we can only use dph_t as an approximate value of $dph_{\langle i,j,k \rangle,t}$. Here, also as example, the threshold of distance is set as the radio range of nodes. Since the third case is between the above two cases, the value of $dph_{\langle i,j,k \rangle,t}$ in the third case can be set as the average of dph_{near} and dph_t. These three cases are shown in Fig. 4.

In Fig. 4, N_p and N_q are two other normal nodes which connect N_x and A_t. Summarizing the three cases, we can estimate the value of $dph_{\langle i,j,k \rangle,t}$ as follow, where d_{near} is the distance between $N_{\langle i,j,k \rangle}$ and A_{near}, dph_{near} is the distance per hop of A_{near}.

$$dph_{\langle i,j,k \rangle,t} \approx \begin{cases} dph_{near}, \text{when } d_{near} < range/2 \\ dph_t, \text{when } d_{near} > range \\ (dph_{near} + dph_t)/2, \text{others} \end{cases} \tag{8}$$

Using (7) and (8), N_x can obtain $hop_{\langle i,j,k \rangle,t}$, which is the estimated hop count between $N_{\langle i,j,k \rangle}$ and each anchor A_t. Then, the connectivity difference between $N_{\langle i,j,k \rangle}$ and N_x can be calculated as $\sum_{t=1}^{m} |hop_{\langle i,j,k \rangle,t} - hop_t|$.

The procedure of our Selective 3-Anchor DV-hop algorithm is summarized as follows. The first and second steps are the same as DV-hop algorithm. In the third step, a normal node N_x selects any three non colinear anchors to form a 3-anchor group, and correspondingly generates a 3-anchor estimated position. Then, based on (7) and (8), N_x calculates the connectivity of each 3-anchor estimated position. Finally, N_x chooses the 3-anchor estimated position which has the smallest connectivity difference to N_x.

We should mention an exceptional case concerning the very low ratio of anchors. For example, let's consider a network with 100 nodes, with only 5 of them being anchors. In this case, two different normal nodes may have the same connectivity. That means, the number of anchors m is not

(a) $d_{near}<range/2$ (b) $d_{near}>range$ (c) $range/2<d_{near}<range$
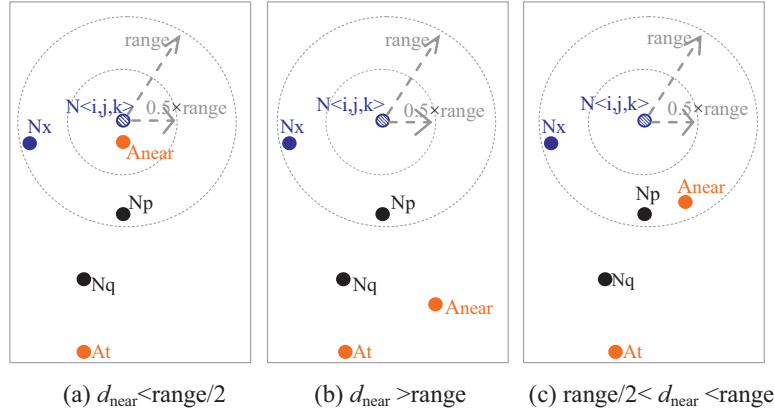
Fig. 4. Three kinds of relative positions.

large enough to ensure that the connectivity vector can identify one unique position. In this case, since our Selective 3-Anchor DV-hop algorithm does not perform well, we suggest Checkout or original DV-hop algorithm be utilized.

The simulation results by MATLAB in [23] prove that, when the ratio of anchors is more than 0.1, our Selective 3-Anchor DV-hop algorithm achieves much better precision than the existing algorithms [18–22]. The improvement of precision ranges from 20% to 57%, comparing with different existing algorithms and with different ratios of anchors.

2.5. Computational complexity estimation

The complexity of an algorithm is commonly expressed using "O" notation, which suppresses multiplicative constants and lower order terms [24]. In this subsection, we use the "O" notation to compare the computational complexity of the three algorithms (the original DV-hop, Checkout DV-hop, and Selective 3-Anchor DV-hop).

As for the original DV-hop algorithm, most of the calculations take place at Step 3. Let's assume that there are m anchors in the network. At Step 3, through the trilateration method [18], each normal node calculates its estimated position based on its estimated distance to all m anchors. So, the computational complexity for the original DV-hop algorithm is O(m).

The Checkout DV-hop algorithm adds a simple calculation to the original DV-hop algorithm, shown as Eq. (4). The computational complexity for Checkout DV-hop algorithm is still O(m).

Nevertheless, the Selective 3-Anchor DV-hop algorithm adds much more computation. It generates all the possible 3-anchor estimated positions in order to select the best candidate. The maximum number being C_m^3, the computational complexity is O(m^3).

In conclusion, we can see that, Selective 3-Anchor DV-hop algorithm has a much higher complexity than the other two algorithms.

3. Our DV-hop localization protocol

To the best of our knowledge, most of DV-hop based algorithms are simulated using MATLAB [18–23]. They all

neglect the issues inherent to a real network, such as collisions, mobility and synchronization. We noted that these problems can significantly influence the localization accuracy. As a result, it is important to estimate the performance of a localization algorithm from a networking point of view. However, IEEE 802.15.4 standard does not define a localization protocol suitable for DV-hop. Hence, we decided to implement a DV-hop localization protocol in order to evaluate the original DV-hop, Checkout DV-hop and Selective 3-Anchor DV-hop algorithms.

Our DV-hop localization protocol is implemented in the WSNet network simulator [25,26]. In the following subsections, we will introduce our DV-hop localization protocol, including the format of the data payload, the improved collision reduction methods and the procedure of the protocol.

3.1. Proposed formats of data payload in each step of DV-hop algorithm

Like DV-hop algorithm, our protocol consists of 3 steps. At Step #1, anchors need to broadcast their positions throughout the network. At Step #2, anchors also need to diffuse their distance-per-hop values. So we must define the frame formats for the message exchange at the first two steps.

Conforming to the general frame format specified in IEEE standard 802.15.4–2009 [27], the frames in DV-hop protocol consist of three basic fields: MHR (MAC header), MAC payload and MFR (MAC footer). Shown in Table 4, MHR is composed of frame control, sequence number, destination address and source address. The detailed information of frame control and sequence number can be found in the IEEE standard. Here, destination and source addresses use 16-bit short format. Since the frames in DV-hop protocol are all to be broadcasted, the destination address should be 0xFFFF.

Data payload carries information from a certain anchor. The information could be the position of the anchor or its distance-per-hop value. The detailed formats of data payload will be given later on. MFR contains the FCS (Frame Check Sequence), that is a 16-bit ITU-T CRC [27].

Two formats of data payload are proposed for the first two steps of DV-hop protocol.

MHR				Data Payload	MFR
Frame Control (16 bits)	Sequence Number (8 bits)	Destination Address (16 bits)	Source Address (16 bits)	(variable length)	FCS (16 bits)

MHR	Data payload Data Type (1 bit) (in total 8 bits)	HopCount (7 bits)	*xi* (32 bits)	*yi*(32 bits)	MFR

MHR	Data payload Data Type (1 bit) In total 32 bits	dphi (31 bits)	MFR

At Step #1, each anchor A_i broadcasts through the network a position frame "*frame_pos$_i$*", so that all nodes (including anchors and normal nodes) can know the position of A_i and the minimum hop count to A_i. The format of *frame_pos$_i$* is shown in Table 5. The data payload is composed of four parts: "Data Type", "x_i", "y_i" and "HopCount".

Data Type identifies the type of information carried by the frame. In DV-hop algorithm, each anchor A_i only need to broadcast two types of information: its position at Step #1 and its distance-per-hop at Step #2. So, we define that Data Type (1 bit) is "0" if this is a position frame, or "1" if this is a distance-per-hop frame. "x_i" and "y_i" represents A_i's coordinates. "x_i", as well as "y_i", is a 32-bit single precision float-point value [28].

"HopCount" is the hop count value initialized to "0" by the initial sender A_i. This hop count value will increase with each retransmission during the flooding of the network. Here, HopCount is limited to 7 bits: the maximum value, 127 has been deemed sufficient for the network.

At Step #2, A_i provides normal nodes with its dph_i by broadcasting a distance-per-hop frame "*frame_dph$_i$*". The format of *frame_dph$_i$* is shown in Table 6. The data payload of *frame_dph$_i$* consists of Data Type and dph_i. The value of Data Type is 1. "dph_i" is a single precision float-point value. Normally, the length of "dph_i" should be 32 bits. However, considering hardwares always process data in bytes (8 bits) and "Data Type" has only 1 bit, we assume that the first bit of the float-point value is used for "Data Type" and the other 31 bits are used for dph_i. However, when a node retrieves the value of dph_i, it should automatically add one bit "0" to the end of dph_i, so that a 32-bits float-point format can be obtained. Since the "0" is the last bit at right end, its influence to the value of dph_i is very little.

3.2. Proposed Enhanced CSMA/CA (E-CSMA/CA) access method

The IEEE standard 802.15.4–2009 defines several channel access methods that can help reduce collisions, for example, slotted CSMA/CA and non-slotted CSMA/CA. Slotted CSMA/CA requires a network coordinator which at regular intervals sends beacon messages for synchronization and network association. On the other hand, non-slotted CSMA/CA does not require the transmission of beacons, thus it can serve for not only star or tree networks but also ad-hoc networks. Due to this simplicity and flexibility, non-slotted CSMA/CA is a popular method for low-cost sensor networks. Therefore, in this paper, we mainly focus on non-slotted CSMA/CA.

The original DV-hop algorithm has not considered the problem of frame collisions, which however frequently happen during the broadcasts of position frames and distance-per-hop frames. Even if the non-slotted CSMA/CA in IEEE 802.15.4 is used as the MAC layer protocol, it cannot effectively reduce collisions in DV-hop. That is because in point-to-point communication, the CSMA/CA scheme normally generates the ACK (acknowledgement) signal to ensure a final successful transmission. However, as for DV-hop protocol, since all the communications are broadcasts, no ACK signal is sent, thus it becomes non-slotted CSMA/CA without ACK, which cannot ensure successful transmissions if collisions exist. In the following, we first analyze how the collisions take place, and then introduce our solution E-CSMA/CA (non-slotted Enhanced CSMA/CA without ACK).

The collisions may happen when anchors simultaneously broadcast their position frames or distance-per-hop frames. At the beginning of Step #1, it is assumed that anchors are simultaneously ready to broadcast their position frames. According to the principle of CSMA/CA without ACK, each anchor first waits for a short random period, and then if the channel is still free, the position frame is sent immediately. Here, the short random period is randomly chosen among 8 values which are 0, t_{bo}, $2 \times t_{bo}$, ..., $7 \times t_{bo}$ [27], where t_{bo} is the back-off period. According to the standard IEEE 802.15.4–2009, if the data rate is 250 kbps, then t_{bo} is 320 μs, and the maximum value of this random period is 7×320 μs = 2.24 ms. With such a short random waiting period, when anchors simultaneously broadcast position frames throughout the network, collisions easily occur. The same phenomenon could also happen at Step #2 of DV-hop when anchors send their distance-per-hop frames simultaneously.

The solution that we use to reduce collisions is to make the senders (nodes ready for sending frames) wait for

another longer random duration before they perform CSMA/CA. So the probability of collision is reduced. The details about this longer waiting period are described in the following.

At the beginning of Step #1, each anchor A_i first waits for a random duration denoted as t_{wpi}. Then, A_i performs CSMA/CA and sends its position frame. Similarly, at the beginning of Step #2 of DV-hop, after each anchor A_i has calculated its distance per hop denoted as dph_i, it waits for a random duration denoted as t_{wdi}. Then, A_i performs CSMA/CA before sending its distance-per-hop frame $frame\_dph_i$.

The following two figures show how collisions happen and how our access method E-CSMA/CA works. In Fig. 5, it is assumed that three anchors A_1 A_2 A_3 start their first step simultaneously: at T_0 they perform the non-slotted CSMA/CA without ACK. A_1 and A_2 happen to choose the same period $2 \times t_{bo}$, while A_2 wait for a longer period $5 \times t_{bo}$ before broadcasting its position frame. Since A_1 and A_2 send out their position frames at the same time, the two frames will arrive simultaneously at the common neighbour node of both A_1 and A_2, thus a collision occurs at Step #1. The same phenomenon could take place at Step #2, with A_2 and A_3 choosing the same waiting period $1 \times t_{bo}$.

Fig. 6 shows an example of our collision reduction method, using the same scenario of Fig. 5. Comparing these two figures, we can see that our method adds an extra random duration before the beginning of the CSMA/CA procedure at each anchor. Thus, the probability of simultaneous emissions is reduced.

In fact, our collision reduction method E-CSMA/CA should also be applied to the relay nodes. These relay nodes, either anchors or normal nodes, help relay the position frame or distance-per-hop frame by broadcast. According to our method, every time a relay node is ready to perform CSMA/CA, this node needs to wait for a supplementary random duration t_{wr}.

3.3. Parameters for the end of each step

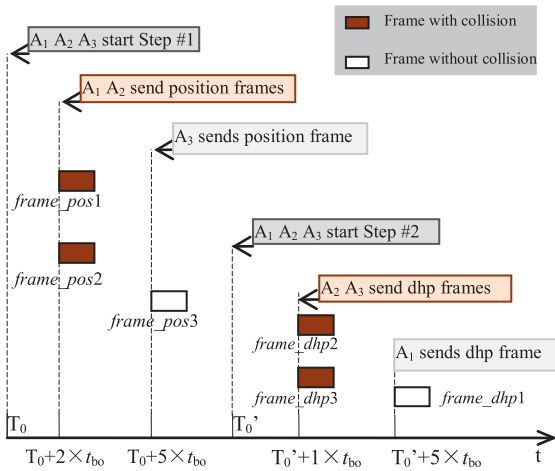As for DV-hop algorithm, the first step ends as soon as every node in the network has received all anchors'
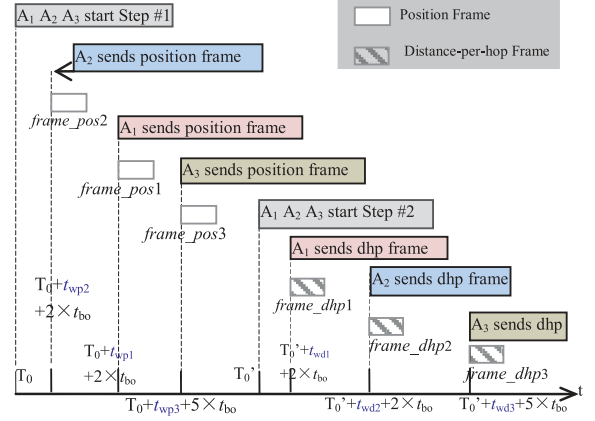

Fig. 6. Example of our access method E-CSMA/CA.

position frames, while the second step ends on condition that all anchors' distance-per-hop frames have been received. These two ending conditions can be fulfilled in an ideal scenario by a mathematic simulator such as MATLAB. However, in practical network scenarios, the ending conditions cannot be reached because the algorithm will encounter two problems. Solving the problems, we propose several parameters to control the end of the first two steps of DV-hop.

As for the first problem, it is unnecessary for nodes to receive all anchors' positions, especially when the total number of anchors is very large. Because mobile normal nodes need to calculate their positions as quickly as possible, it could take too much time for them to collect all anchors' positions. Therefore, each node can set a maximum number of anchors whose information they take into account: the node will then wait until it has identified this number of distinct anchors. This maximum number of anchors can be denoted as 'num_wait_pos'. Then, as long as a normal node has received num_wait_pos anchors' positions, it can stop relaying position frames and end Step #1 of DV-hop algorithm. As for anchors, when an anchor has received num_wait_pos-1 anchors' positions, it can end Step #1. (Here, it is 'num_wait_pos-1' instead of 'num_wait_pos', because the number 'num_wait_pos' includes A_i). Similarly, if a normal node has received num_wait_dph anchors' distance-per-hop, it can end Step #2. Normally, num_wait_pos is no less than num_wait_dph.

The second problem occurs when collisions happen or the total number of anchors is less than 'num_wait_pos' or 'num_wait_dph'. When collisions occur during the first two steps of DV-hop algorithm, a few nodes may miss some anchors' position frames as well as distance-per-hop frames. As a result, these nodes might never receive as many as 'num_wait_pos' anchors positions as expected, neither num_wait_dph anchors' distance-per-hop. Of course, this phenomenon could also happen if the total number of anchors is less than 'num_wait_pos' or 'num_wait_dph'.

Timers will be used to solve the second problem. To end Step #1, we need to set a timer for each node N_i at the time instant $T^0_i + t_{s1}$. Since all nodes periodically execute

Fig. 5. Collisions occur at Step #1 and Step #2.

DV-hop localization protocol, T^0_i is the beginning time of N_i's localization period. All nodes could have the same beginning time if the network is well synchronized. If this is not the case, each node might begin its period at a different instant. t_{s1} is the maximum duration of Step #1 and is configured and shared by all nodes. Before the expiration of $T^0_i + t_{s1}$, those anchors who have already received as many as 'num_wait_pos-1' anchors' positions must immediately end Step #1. At $T^0_i + t_{s1}$, all anchors must end Step #1 regardless of the amount of data collected.

In order to end Step #2, we can set a timer at the time instant $T^0_i + t_{s1} + t_{s2}$. Here, t_{s2} is the maximum duration of Step #2, which is shared by all normal nodes. In fact, Step #3 of DV-hop algorithm is designed for normal nodes to calculate their positions. Hence, the timer for ending Step #2 is specific to normal nodes. Before $T^0_i + t_{s1} + t_{s2}$, those normal nodes who have already received as many as 'num_wait_dph' anchors' distance-per-hop frames and 'num_wait_pos' anchors' position frames, could immediately end Step #2 and start Step #3. At time '$T^0_i + t_{s1} + t_{s2}$', normal nodes that have not yet received the specified amount of data need to nevertheless start Step #3.

In DV-hop algorithm, all broadcasts of frames are included at Step #1 and Step #2, while Step #3 only includes the position calculation. Since broadcasts normally take much more time than calculation, the total duration of Step #1 and Step #2 is very close to the entire period of localization. That is, $t_{s1} + t_{s2} \approx t_p$. Here, t_p is the duration of a localization period. Besides, since Step #1 and Step #2 both broadcast frames, their duration should be similar. That is $t_{s1} \approx t_{s2}$. For example, t_{s1} could be set as $t_p/2$, while t_{s2} could be set as $t_p * (3/8)$. Then, the time left is devoted to Step #3, that is: $t_p - t_{s1} - t_{s2} = t_p/8$.

3.4. Procedure of our DV-hop localization protocol

The execution of our DV-hop localization protocol is shown in the following two figures. One figure shows the procedure for anchors and another illustrates the procedure for normal nodes.

Fig. 7 shows the procedure followed by each anchor A_i. The duration of the localization period is t_p, and A_i begins its period at the time T^0_i. Then, according to our collision avoidance method, A_i first waits for a random duration t_{wpi}, and then broadcasts through the network its position frame which has been defined in Section 3.1. Meanwhile, A_i also receives and relays the positions frames of other anchors. When A_i has received 'num_wait_pos-1' anchors' position frames, it will immediately end Step #1 and enter Step #2. This time instant is denoted as Tr_i. However, if A_i could not receive as many as 'num_wait_pos-1' anchors' position frames until the time instant $T^0_i + t_{s1}$, it will still end Step #1 when it reaches $T^0_i + t_{s1}$. So A_i ends Step #1 at the time instant Tr_i or $T^0_i + t_{s1}$. A_i begins Step #2 by calculating its distance-per-hop. Then, according to our collision reduction method, A_i waits for a random duration t_{wdi}, and then broadcasts through the network its distance-per-hop frame. Meanwhile, A_i also helps relay the distance-per-hop frames of other anchors. When A_i ends Step #2, it also ends one localization period, because only normal nodes participate in the third step.
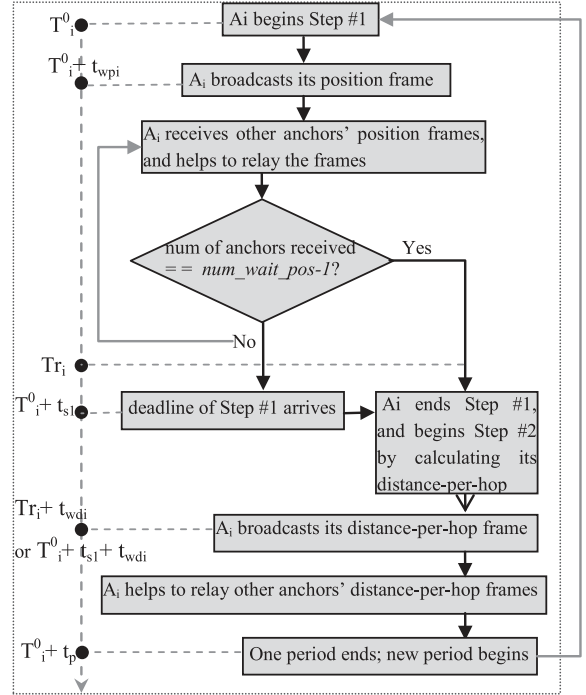

Fig. 7. Procedure for each anchor A_i.

Fig. 8 shows the procedure for each normal node N_j. N_j begins its period at the time T^0_j. During the first two steps, N_j receives and relays anchors' frames. When N_j has
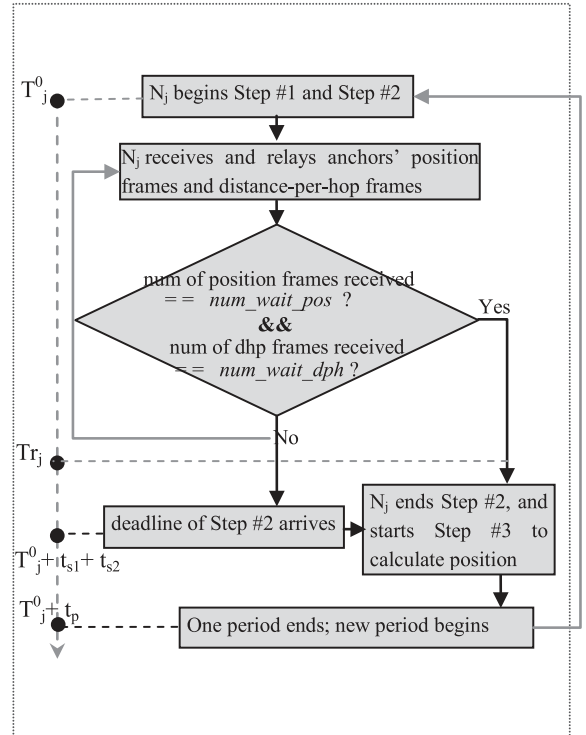

Fig. 8. Procedure for each normal node N_j.

received as many as *num_wait_pos* anchors' position frames and as many as *num_wait_dph* anchors' distance-per-hop frames, it will immediately end the first two steps. This time instant is denoted as Tr_j. However, if N_j could not receive as many as *num_wait_dph* distance-per-hop frames until the time $T^0_j + t_{s1} + t_{s2}$, it still ends Step #2 anyway. Since t_p is the duration of the period, at the time $T^0_j + t_p$, N_j will end the current period.

In this section about our DV-hop localization protocol, we have presented the frame structure, the improved collision reduction method, several parameters to end each step and finally the procedure of the protocol. Using this protocol, DV-hop based algorithms can be implemented in network scenarios.

4. Performance evaluation of DV-hop based algorithms

In this section, based on the implementation of our DV-hop protocol, we evaluate the performance of the original DV-hop, Checkout DV-hop, and Selective 3-Anchor DV-hop algorithms. First, we assign values to the parameters of our DV-hop protocol and also configure simulation scenarios. Second, through network simulations, we investigate the specific performance of the original DV-hop algorithm. Finally, in terms of mobility, synchronization and network overhead, we present comparative evaluation of the concerned DV-hop based algorithms.

4.1. Parameters quantization and scenario configuration

The simulator we use is WSNet, which is an event-driven simulator designed by three researchers from INRIA [25]. Compared to others such as NS-2 and OPNET, WSNet not only facilitates the development of new models, but also supplies sufficient modules at each layer [26]. Using WSNet, we have implemented our DV-hop localization protocol as a model in C language.

In the previous section, we have proposed several important parameters of DV-hop localization protocol. When we implement the protocol, we need first quantize these parameters.

As introduced in Section 3.2, t_{wpi} is A_i's random waiting time before performing CSMA/CA to broadcast its position frame, while t_{wdi} is the random duration that preceded the broadcast of its distance-per-hop frame. As for the range of t_{wpi} or t_{wdi}, as an example, we can set their minimum value as 0. Their maximum value cannot be too small; otherwise different anchors might easily send frames at the same time, making collisions happen. 0.5 s is assumed to be big enough for this maximum value, considering an example of just 2.24 ms given in Section 3.2. Thus, t_{wpi} and t_{wdi} are uniform-random values between 0 and 0.5 s.

Also proposed in Section 3.2, t_{wr} is any relay node's random waiting time before it resends position frames or distance-per-hop frames. The maximum value of t_{wr} should not be too big because mobile nodes cannot wait too long to receive the positions or distance-per-hops from the far-away anchors. In our simulation, the maximum of t_{wr} is set as 10 ms and its minimum is 0.

Our simulation scenario takes place within a 100×100 m^2 area. Inside this area, 100 nodes including
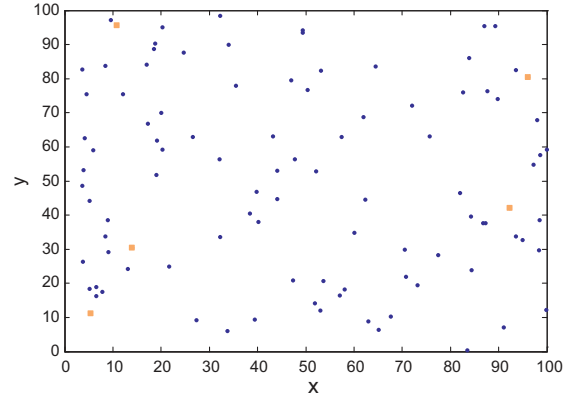

Fig. 9. Example of nodes distribution.

anchors and normal nodes are randomly placed according to a uniform distribution. An example of distribution is shown in Fig. 9. In this example, 5 of the 100 nodes are anchors which are represented as squares, while others are normal nodes. This illustrates a 5% ratio of anchors, which is defined as the ratio of the number of anchors to the total number of nodes.

The scenario parameters and their values are listed in Table 7. The last 5 parameters marked by '*' have different values in different scenarios, while other parameters are constant over the scenarios.

We use a log-distance pathloss radio propagation model, which is usually applied in indoor scenarios [29]. Note that the problem of interference from other technologies is not studied in our scenarios.

Since low-cost sensor nodes have limited memory, we assume that, each node can receive at most 30 anchors' positions at Step #1, and at most 20 anchors' distance-per-hop at Step #2. That is to say, *num_wait_pos* and *num_wait_dph* proposed in Section 3.3 are respectively 30 and 20.

The network can be synchronized (all nodes can simultaneously begin their localization period) or unsynchronized (nodes start time will be different). As for mobility, anchors are static, while normal nodes may be static or mobile. All these scenarios are considered and their simulation results will be presented in the following subsections. We will first investigate the performance of original DV-hop algorithm, and then compare it with Checkout DV-hop and Selective 3-Anchor DV-hop.

4.2. Simulations and evaluations on original DV-hop algorithm

In the following, based on our DV-hop localization protocol, we will present 6 scenarios for the original DV-hop algorithm (including 3 particular static scenarios, 1 general static scenario, 1 mobile synchronized scenario and 1 mobile unsynchronized scenario). As for the first three static scenarios, we aim to obtain specific performance of DV-hop algorithm without influence of node movement. But from the fourth static scenario and other 2 mobile scenarios, we aim to know general performance. Thus, for the first three static scenarios, we set network simulation

Table 7
Senario parameters.

Radio range of nodes	20 m
Physical Data rate	250 kbps
Radio propagation	Log-distance pathloss propagation model
Interference	none
Physic layer protocol	IEEE 802.15.4, 2.4 GHz, OQPSK
MAC layer protocol	IEEE 802.15.4 non-slotted CSMA/CA
Localization period t_p	6s
A_i's waiting time before sending: t_{wpi} and t_{wdi}	Both randomly selected between 0 and 0.5 s
Maximum duration of Step #1: t_{s1}	$1/2 * t_p = 3$ s
Maximum duration of Step #2: t_{s2}	$3/8 * t_p = 2.25$ s
Maximum waiting number: num_wait_pos	30
Maximum waiting number: num_wait_dph	20
Network synchronized or not[a]	To be decided in specific scenario
Ratio of anchors[a]	To be decided in specific scenario
Nodes mobility[a]	To be decided in specific scenario
Network simulation time[a]	To be decided in specific scenario

[a] Parameters having different values in different scenarios.

Table 8
Particular parameters of static scenario 1.

Network synchronized or not	Synchronized and all nodes start at the same time
Ratio of anchors	5%
Nodes mobility	Static (distribution as Fig. 9)
Network simulation time	18 s (3 localization periods)

time as only 18 s (equal as 3 localization periods) to get 3 particular cases for each scenario. As for general static scenario and mobile scenarios, simulation time is set as 3000 s (equal as 500 periods) to obtain average performance.

4.2.1. Static Scenario 1

Since the parameters have already been listed in Table 7, here, we assign values to the parameters marked with an asterisk. Table 8 lists these parameters.

Since the simulation runs for 3 localization periods, we can obtain 3 particular results, as shown in Table 9. The results are examined using two criteria, location error and number of transmitted frames. In Table 9, location error (in meters) is the average of all distances between each normal node's estimated position and its real position. The location error can be used to evaluate the accuracy of DV-hop algorithm. A smaller location error indicates better accuracy performance. Another parameter is the number of transmitted frames, which is the number of frames transmitted by all nodes during one localization period of DV-hop protocol. The number of transmitted frames can be used for evaluating the network overhead. A higher figure indicates higher network overhead.

From Table 9, we can reach the following conclusions:

(1) Even if the same scenario is applied, in each period, we could obtain different results. This is caused by the random nature of some parameters, for example, t_{wpi} and t_{wdi} in Table 7. Consequently, in each period, the collisions might happen between different nodes and at different times. As a result, the performance will be different for each result.

(2) In the scenario, all three cases use the same distribution of nodes, but the accuracy could be quite different from a run to the other. For example, the location error of Result 1 is much higher than that of Result 3. This indicates that there is strong relationship between the accuracy of DV-hop algorithm and the performance of DV-hop protocol.

(3) Network overhead is studied. In DV-hop protocol, the network traffic exists only during the first two steps. At Step #1, each anchor broadcasts its position frame throughout the network. In order to make all nodes be aware of this frame, every node in the network needs to relay this frame once. If the total number of nodes is num, the number of anchors is $num \times$ 'ratio of anchors', then the number of transmitted frames at Step #1 is at least $num \times (num \times$ 'ratio of anchors'$) = num^2 \times$ 'ratio of anchors'. The same result can be obtained for Step 2. Thus, the number of transmitted frames for DV-hop protocol is about $2 \times num^2 \times$ 'ratio of anchors'. To verify this, for example in this scenario, the number of transmitted frames is at least $2 \times 100^2 \times 5\% = 1000$, which can be supported by the results in Table 9.

Table 9
Performance results of static scenario 1.

Result 1		Result 2		Result 3	
Location error (% radio range)	Number of transmitted frames	Location error (% radio range)	Number of transmitted frames	Location error (% radio range)	Number of transmitted frames
17.60/20 = 88%	1071	12.03/20 = 60%	1223	10.78/20 = 54%	1063

Table 10
Performance results of static scenario 2.

Result 1		Result 2		Result 3	
Location error (% radio range)	Number of transmitted frames	Location error (% radio range)	Number of transmitted frames	Location error (% radio range)	Number of transmitted frames
14.97/20 = 75%	6783	10.01/20 = 50%	7001	16.89/20 = 84%	6780

Table 11
Performance results of static scenario 3.

Result 1		Result 2		Result 3	
Location error (% radio range)	Number of transmitted frames	Location error (% radio range)	Number of transmitted frames	Location error (% radio range)	Number of transmitted frames
15.75/20 = 79%	12,072	17.87/20 = 89%	11,895	20.02/20 = 100%	11,981

(4) The average location error of the three results is $(17.60 + 12.03 + 10.78)/3 = 13.47$ meters (that is 67% in percentage of radio range), while the average number of transmitted frames is 1119. These average results can be finally regarded as the average performance under Static Scenario 1.

4.2.2. Static Scenario 2

From Static Scenario 1 to Static Scenario 2, only the ratio of anchors changes from 5% to 40%.

We can also obtain 3 particular results, as shown in Table 10.

From Table 10, we can deduce the following:

(1) When there are more anchors in the network, the network overhead of DV-hop protocol will increase. In this scenario, according to the previous estimate on network overhead, the number of transmitted frames should be at least $2 \times 100^2 \times 30\% = 6000$ (Here, it is 30% rather than 40%, because we set maximum waiting number '*num_wait_pos*' to be 30, shown in Table 7). This large amount of transmitted frames brings heavy traffic to the network.

(2) An increase in the number of anchors does not necessarily improve localization accuracy of DV-hop algorithm. This conclusion can be obtained by comparing Tables 9 and 10. The location errors in Table 10 (with 40 anchors) are a little higher than those in Table 9 (with 5 anchors). One reason is that when the anchor population is large, the traffic in the network becomes heavy, which leads to more collisions. This in turn prevents normal nodes from receiving the right position frames which have the smallest hop count values.

4.2.3. Static Scenario 3

From Static Scenario 2 to Static Scenario 3, the ratio of anchors changes from 40% to 80%.

We can also obtain 3 particular results, as shown in Table 11.

From Tables 10 and 11, we can reach the following conclusion: if there are too many anchors, the network traffic of DV-hop protocol will be too heavy, generating excessive collisions and causing the localization accuracy to decline. As a result, when the ratio of anchors is greater than or equal to 40%, instead of using DV-hop algorithm, we need to use other low-traffic localization solutions, such as Centroid and CPE.

4.2.4. General static scenario and mobile scenarios

From the above three static scenarios, we have found that DV-hop protocol is not suitable for scenarios with large number of anchors. From now on, we will configure the scenarios with no more than 30 anchors (the total number of nodes still being 100). In the following, we present three scenarios, including general static scenario, synchronized mobile scenario and unsynchronized mobile scenario. First, we list the particular parameters for each scenario (the common parameters are the same as Table 7). Then, their simulation results are presented together.

4.2.4.1. Particular parameters of general static scenario. The particular parameters of general static scenario are listed in Table 12. In order to obtain more general results than the previous three static scenarios, we increase the simulation duration to 5000 s which allows for 500 localization periods.

4.2.4.2. Particular parameters of synchronized mobile scenario. The particular parameters of the synchronized mobile scenario are listed in Table 13. Anchors remain static, while normal nodes move in billiard mode. That means, when a normal node reaches the edge of the $100 \times 100 \text{ m}^2$ simulation area, this node will bounce back like a billiard ball. The speed is fixed as 0.5 m/s, which corresponds to low-speed human movement.

4.2.4.3. Particular parameters of unsynchronized mobile scenario. The particular parameters of unsynchronized mobile scenario are the same as those in Table 13, except the synchronization. Here, nodes will start at different time. Some nodes might start very late, while others start earlier. This means that when late nodes begin Step #1, some early nodes might have already finished their Step #2. For example, as shown in Fig. 10, anchor A_i starts its

Table 12
Particular parameters of general static scenario.

Network synchronized or not	Synchronized and all nodes start at the same time
Ratio of anchors	5, 10, 15, 17, 19, 20, 25, 30/100
Nodes mobility	Static (distribution as Fig. 9)
Network simulation time	3000 s (500 localization periods)

Table 13
Particular parameters of synchronized mobile scenarios.

Network synchronized or not	Synchronized and all nodes start at the same time
Ratio of anchors	5, 10, 15, 17, 19, 20, 25, 30/100
Nodes mobility	Anchors are static, normal nodes move at a speed of 0.5 m/s in billiard mode
Network simulation time	3000 s (500 localization periods)

Step #1 so late that anchor A_k has already ended its Step #2.

However, this kind of unsynchronized situations has been considered by our DV-hop protocol. In the protocol, when a normal node is working at Step #2, it can receive both distance-per-hop and position frames. Therefore, no matter how late an anchor begins Step #1, its position frame and distance-per-hop frame will sooner or later be received by all nodes.

4.2.4.4. Simulation results of general static scenario and mobile scenarios. The simulation results of our three scenarios (general static, synchronized mobile, and unsynchronized mobile) using the original DV-hop algorithm are presented in Figs. 11 and 12. The data is collected on a per anchor ratio basis. Fig. 11 shows the average location error per node per localization period, expressed as a percentage of the radio range. Fig. 12 presents the average number of transmitted frames per localization period.

From Fig. 11, we can see that, for all scenarios, as the number of anchors increases, the location error declines, which means the localization accuracy improves. As expected, the location error increases when the number of anchors goes over 20 or 25. This is caused by the increase in frame collisions. As there are many anchors, a large number of frames are broadcasted through the network, thus the collisions can easily occur.
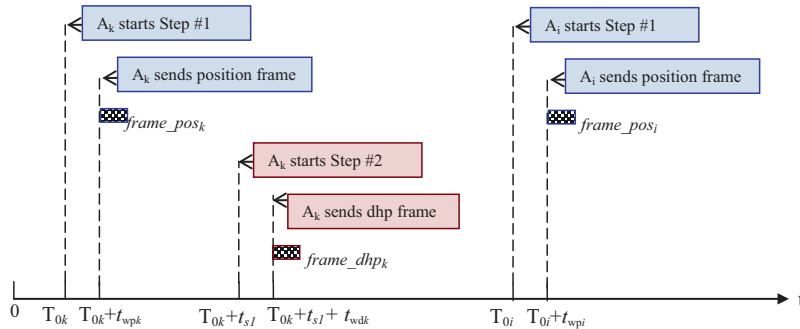
Comparing the location error between general static and synchronized mobile scenarios in Fig. 11, we can see the influence of node mobility. The location error of synchronized mobile scenario is normally a little bigger than that of general static scenario. The reason may be that we have not used any position prediction method. Therefore, when nodes are mobile, their estimated positions do not match their latest positions.

From Fig. 11, we can also notice that, although lacking a position prediction mechanism, the unsynchronized mobile scenario generally has the best accuracy. That is because, in the unsynchronized scenario, nodes generally start their localization period at different times. Hence, compared with the synchronous scenario, the anchors have less chance to broadcast their positions simultaneously, resulting in fewer collisions.

We notice that the accuracy performance of DV-hop is not very satisfying. Its minimum location error corresponds to half the radio range. These results will nevertheless serve as a benchmark in the evaluation of Checkout DV-hop and Selective 3-Anchor DV-hop algorithms. Their simulation results will be presented in the next section.

The three scenarios have almost the same simulation results regarding the number of transmitted frames, as shown in Fig. 12. We can see that when the number of anchors is less than 20, the transmitted frames number increases linearly with the number of anchors. But this linearity ends when the number of anchors exceeds 20. That is because, according to the settings (Table 7), each node is supposed to keep at most 20 anchors' distance-per-hop values at Step #2. That means, when a node has obtained as many as 20 anchors' distance-per-hop, its memory for distance-per-hop is supposed to be completely occupied. If this node receives another distance-per-hop frame in the future, it has to discard this frame. However, in a scenario with less than 20 anchors, since the memory for distance-per-hop can never be completely occupied, new anchors' distance-per-hop frames are always recorded and then transmitted instead of being discarded.

4.3. Comparative evaluation of DV-hop, Checkout DV-hop and Selective 3-Anchors DV-hop algorithms

Checkout DV-hop and Selective 3-Anchor DV-hop algorithms both share the same Step #1 and Step #2 with DV-hop algorithm. The difference between these 3 algorithms lies
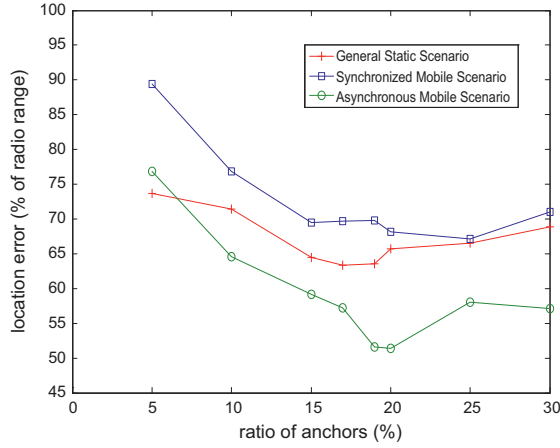

Fig. 10. Example of unsynchronized scenario.

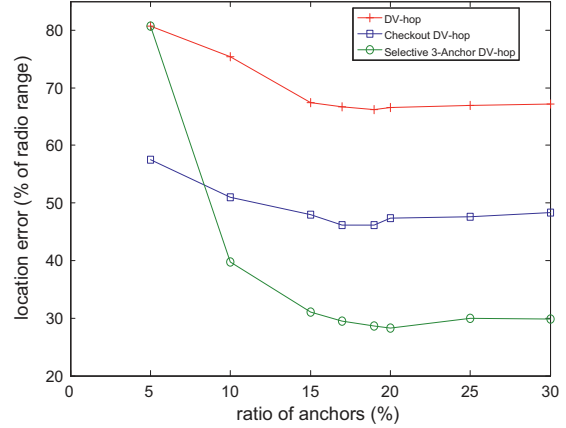**Fig. 11.** Location error in three scenarios.

Fig. 13. Location error (static scenarios, range 20 m).

in the computation phase which is Step #3. Therefore, Checkout DV-hop and Selective 3-Anchor DV-hop can use the same DV-hop protocol as the one used for original DV-hop algorithm. The following sections will present the comparison of the simulation results of these 3 algorithms.

4.3.1. Comparison under static scenarios

The static scenarios we use here are the same as those in Section 4.2.4.1. The simulation results about the number of transmitted frames remain the same as Fig. 12 in Section 4.2.4.4. The results on location error are shown in Fig. 13. This figure indicates that, in general, the localization accuracy of Checkout DV-hop is about 25% better than that of original DV-hop. When the anchor ratio is larger than 5%, Selective 3-Anchor DV-hop has better accuracy. The improvement is about 30% when considering Checkout DV-hop and about 55% compared to DV-hop.

It should be mentioned that, when the ratio of anchors is as low as 5%, many normal nodes will have the same connectivity. Thus, Selective 3-Anchor DV-hop algorithm cannot identify the unique solution. It then temporarily

Fig. 12. Number of transmitted frames in three scenarios.

utilizes DV-hop algorithm. That is why in Fig. 13 Selective 3-Anchor DV-hop and the original DV-hop both start from the same point.

In order to investigate the radio range's influence on accuracy, we change the node radio range from 20 m to 15 m. Meanwhile, all other scenario parameters remain the same. Fig. 14 illustrates the results with a radio range of 15 m.

Fig. 14 shows that, in general, the accuracy of Selective 3-Anchor DV-hop is 25% better than Checkout DV-hop's and about 50% better than the original DV-hop algorithm. Comparing Figs. 13 and 14, the accuracy improvement is similar when the radio range passes from 20 m to 15 m. The reason can be that when the radio range decreases, there are fewer neighbour nodes around each normal node, thus less connectivity information can be obtained; but at the same time, there are fewer collisions in the network.
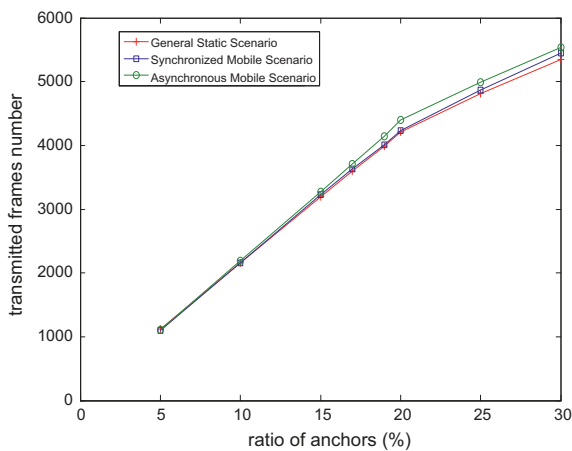
4.3.2. Comparison in synchronized mobile scenarios

The scenarios here are the same as those in Section 4.2.4.2. The number of transmitted frames during the execution of the 3 algorithms remains the same as described in Fig. 12. The simulation results in terms of location error are presented in Fig. 15.

Fig. 15 presents the relationship between accuracy and anchor ratio for DV-hop, Checkout DV-hop and Selective 3-Achor DV-hop in synchronized mobile scenarios. The accuracy improvement of Checkout DV-hop over DV-hop is between 20% and 25%. When the number of anchors is larger than 5, the improvement of Selective 3-Anchor DV-hop over Checkout DV-hop ranges from 18% to 32%, and is between 37% and 48% compared to DV-Hop.

In order to investigate the accuracy with a different radio range, we reduced the radio range to 15 meters. The other parameters remain the same. The results are shown in Fig. 16. Comparing Figs. 15 and 16, we noticed that the accuracy for the proposed protocols is not affected by the change in the radio range. Selective 3-Anchor DV-hop's accuracy is about 20% better than Checkout DV-hop algorithm and about 50% better than the original DV-hop algorithm.
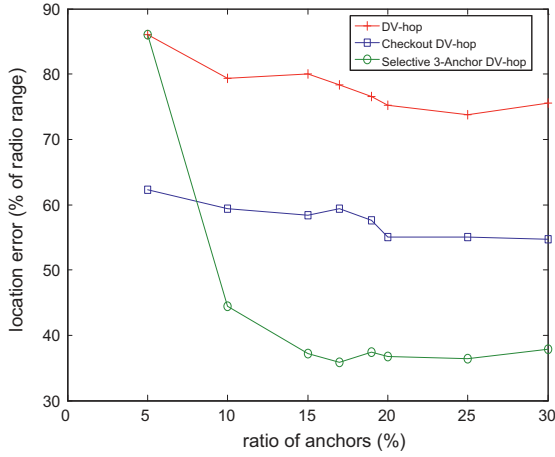
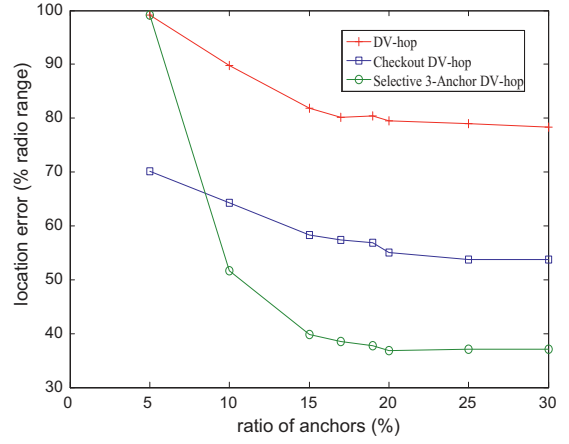**Fig. 14.** Location error (static scenarios, range 15 m).

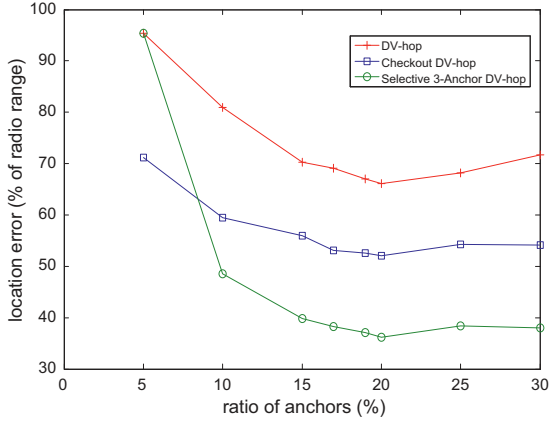**Fig. 16.** Location error (range 15 m).

Fig. 15. Location error (sync mobile scenarios, range 20 m).
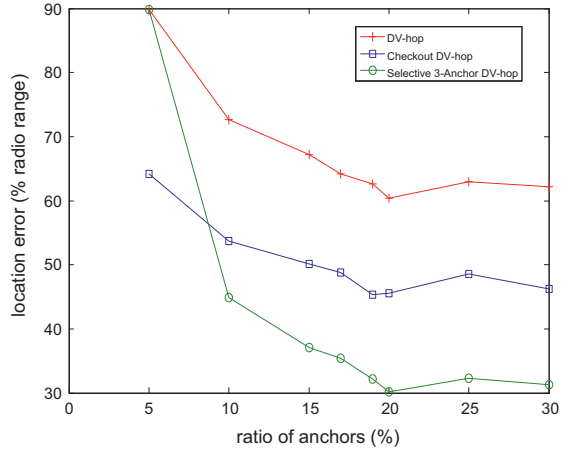

Fig. 17. Location error (unsync mobile scenarios, range 20 m).

4.3.3. Comparison in unsynchronized mobile scenarios

The scenarios of this section are the same as those in Section 4.3.4.3. The number of transmitted frames when executing the 3 algorithms remains the same as illustrated by Fig. 12. The simulation results in terms of location error are presented in Fig. 17.

Fig. 17 shows that the accuracy improves by 10–20% when using Checkout DV-hop instead of DV-hop. When the number of anchors is larger than 5, the improvement of Selective 3-Anchor DV-hop over Checkout DV-hop is between 20% and 34%, and when compared to DV-hop, it is between 32% and 45%.

We also change the radio range from 20 m to 15 m, while all other scenario parameters remain the same. The simulation results for the three algorithms under unsynchronized mobile scenarios with the radio range set to 15 m are shown in Fig. 18.

Fig. 18 indicates that, in general, the accuracy of Selective 3-Anchor DV-hop is 30% better than Checkout DV-hop and about 45% better than the original DV-hop algorithm. We can conclude that the change in radio range had minimal impact on the performance.

4.3.4. Influence of node placement

In the simulation scenarios of previous sections, nodes are randomly distributed because we wanted to obtain average precision of DV-hop based algorithms under all distributions. But it is also interesting to know which kind of distribution of nodes can give the best precision. Therefore, in this subsection, we investigate the influence of node distribution on the precision of DV-hop based algorithms.

During the previous simulations, 100 nodes are randomly distributed inside the 100×100 m^2 square area. In total, we simulated 500 different (random) distributions of nodes. As for each distribution, anchors are randomly chosen from all the 100 nodes, and we also simulated 2000 random placements of anchors for each distribution. The location errors mentioned in the previous sections were the average location error of all these distributions.

However, since DV-hop algorithm is based on the connectivity between normal node and each anchor, the variation of placements of nodes, especially the placement of anchors, can influence the accuracy of DV-hop algorithm.
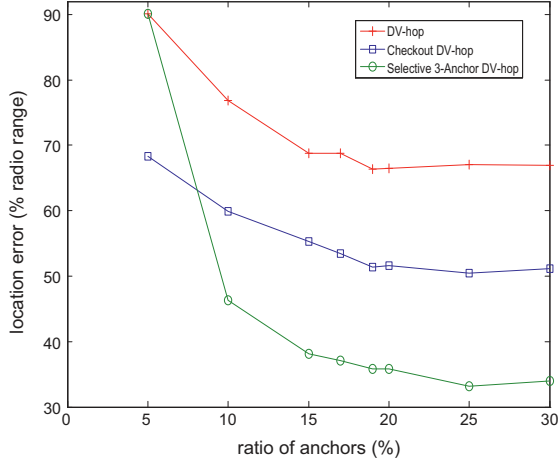
Fig. 18. Location error (same scenario, range 15 m).

That means, when there is a change in the distribution of nodes, the average location error of normal nodes may also change.

In this subsection, through simulations on different distributions of nodes, we give the best placement of nodes that can provide the smallest average location error of normal nodes. The result is shown in Fig. 19. Among 100 nodes, 15 are anchors marked as circles in Fig. 19, thus in this example the ratio of anchor is 15% (Results of other ratios of anchors will be given in our future work). Among 500 random distributions of nodes and 2000 random placements of anchors for each distribution, we found the best distribution for DV-hop algorithm shown in Fig. 19(a), and the best distribution for Selective 3-Anchor DV-hop algorithm shown in Fig. 19(b). While the average location error (percentage of radio range) of all distributions for DV-hop is about 44.9%, the location error of its best distribution is only 25.4%. As for Selective 3-Anchor DV-hop, its average location error of all distributions is about 32.0%, but that of the best distribution is just 19.1%. This encourages us in our future work to design a topology control algorithm to transform a random distribution of nodes to the best placement.
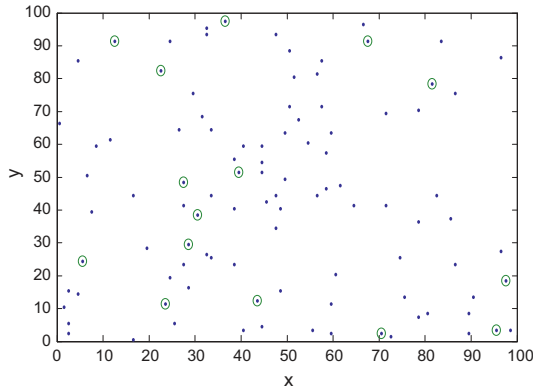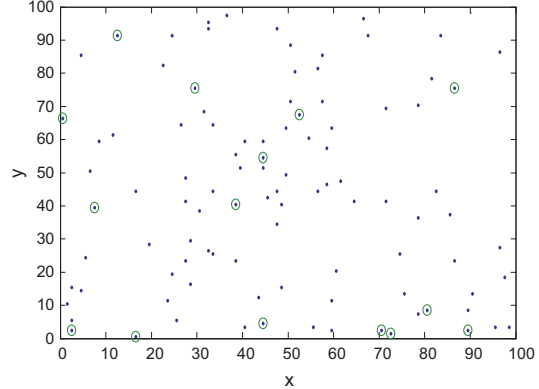
From Fig. 19, we can find one common thing in the best distributions of nodes for the both two algorithms: most anchors are placed near to the border of the region, 1 anchor is placed near to the center of the region, and other anchors are between the center and the border. For example, in Figs. 19(a) and 11 out of 15 anchors are near to the four edges of the region, 1 anchor is near to the center, and other 3 anchors are between the center and the edges. More clearly, we can find the same phenomenon from Fig. 19(b). The deeper research on this best placement of nodes will be included in our future work.

4.3.5. Summary of analysis

As the conclusion of Section 4.4, we review the results of the simulation of the three algorithms, the original DV-hop, the Checkout DV-hop and the Selective 3-Anchor DV-hop.

The original DV-hop and the Checkout DV-hop have the same requirement regarding the minimum number of anchors. They need at least 3 anchors in any network. But the Selective 3-Anchor DV-hop generally requires more anchors. As shown in previous sections, its interest appears when there are at least 10 anchors in a network with 100 nodes in total.

As for accuracy, on average, the original DV-hop's location error is about 70% of the radio range. When using Checkout DV-hop, this error is about 55% of the radio range. Selective 3-Anchor DV-hop is the best choice as the location error drops to 35% on average.

Finally, since these three algorithms share the same communication procedure (Step #1 and Step #2) and only differ on the position calculation (Step #3), they have the same network overhead.

As for calculation time, since the position evaluation is restricted to Step #3, the calculation time does not exceed the duration of Step #3. In our simulation, the duration of Step #3 is set as $1/8 * t_p = 0.75$ s. Therefore, all the three algorithms spend no more than 0.75 s calculating the position.

The three algorithms report more accurate results in unsynchronized scenarios. That is because in synchronous scenarios, all nodes are configured to start their

(a) best for DV-hop (location error 25.4%)

(b) best for Selective 3-Anchor DV-hop (location error 19.1%)

Fig. 19. Distribution of nodes that has smallest location error.

Table 14
Brief comparison on the three algorithms under all scenarios.

	DV-hop	Checkout DV-hop	Selective 3-Anchor DV-hop
Accuracy	Modest scenarios: unsynchronized mobile > static > synchronized mobile	Better: 20% > DV-hop	Best:50% > DV-hop
Calculation complexity	$O(m)$	$O(m)$	$O(m^3)$
Network overhead	share the same network overhead		

Note: in this table, " > " means "better accuracy than".

localization period simultaneously, which leads to more collisions. But in unsynchronized scenarios, anchors generally have fewer chances to broadcast their positions simultaneously. Therefore, synchronization is not a necessary condition for the use of DV-Hop based solutions.

The last parameter is mobility. With the speed as low as 0.5 m/s, the accuracy of the three algorithms in synchronized mobile scenarios is about 10% lower than in static scenarios. However, the accuracy in unsynchronized mobile scenarios is about 10% better than in static scenarios. This suggests that, in the context of low-speed mobility, the influence of synchronization becomes more noticeable.

The following table gives a brief comparison of the three algorithms (see Table 14).

5. Conclusion

In the context of low-cost wireless ad-hoc sensor networks, the range-free localization scheme is not only more cost-effective than range-base scheme, but also more robust. Among typical range-free localization algorithms, DV-hop algorithm shows its particular advantage for localizing the normal nodes that have less than three neighbour anchors. Focusing on DV-hop, we have proposed two improved algorithms, Checkout DV-hop and Selective 3-Anchor DV-hop. Checkout DV-hop has better accuracy than original DV-hop algorithm because it adjusts the estimated position of a normal node based on its distance to the nearest anchor. Generally, the nearest anchor to a normal node has the most accurate estimated distance compared with other anchors. Selective 3-Anchor DV-hop algorithm outperforms Checkout DV-hop and the original DV-hop because many more candidate positions for each normal node are created, and the best candidate is chosen based on its hop counts difference with the normal node.

In order to implement and evaluate the proposed algorithms, a new DV-hop localization protocol is presented. Based on IEEE standard 802.15.4–2009, we design new data payload formats, a new access method E-CSMA/CA to improve the performance of non-slotted CSMA/CA. In addition, several parameters such as timers and maximum number of received anchors are proposed to end each step of DV-hop. Finally, using this protocol, we investigate the performance of DV-hop, Checkout DV-hop and Selective 3-Anchor DV-hop in terms of mobility, synchronization, overhead and accuracy. The simulation results prove that:

(1) The number of anchors has an important role in localization accuracy: normally the accuracy gets better when there are more anchors, however, if the number of anchors is too big (for example, anchor ratio is more than 40%), the accuracy of all three algorithms will decline because of the heavy traffic and increased collisions.

(2) Our protocol can work not only in synchronized scenarios, but also in unsynchronized scenarios. As a matter of fact, the accuracy is better in unsynchronized scenarios compared to synchronized scenarios.

(3) The accuracy of the three algorithms in synchronized mobile scenarios is about 10% lower than in static scenarios. However, the accuracy in unsynchronized mobile scenarios is about 10% better than in static scenarios. This shows that in the context of low-speed mobility, the influence of synchronization becomes more noticeable.

(4) All three algorithms have the same network overhead represented by "number of transmitted frames" being about $2 \times num^2 \times$ 'ratio of anchors', but they differ on computational complexity. The original DV-hop and Checkout DV-hop algorithms have the same complexity, while the Selective 3-Anchor DV-hop algorithm has higher complexity.

(5) Compared to the original DV-hop algorithm, the accuracy is improved by the other two algorithms, Selective 3-Anchor DV-hop being the best choice.

In the future, we will be interested in implementing our algorithms and protocols on a prototype. This will allow us to compare real results with those simulation results.

References

[1] R. Vishnubhotla, P.S. Rao, A. Ladha, S. Kadiyala, A. Narmada, B. Ronanki, S. Illapakurthi, ZigBee based multi-level parking vacancy monitoring system, in: 2010 IEEE International Conference on Electro/Information Technology, May 2010, pp. 1–4.

[2] J. Zhang, G. Song, G. Qiao, T. Meng, H. Sun, An indoor security system with a jumping robot as the surveillance terminal, IEEE Trans. Consum. Electron. 57 (4) (November 2011) 1774–1781.

[3] B. Lee, W.Y. Chung, Multitarget three-dimensional indoor navigation on a PDA in a wireless sensor network, IEEE Sensors J. 11 (3) (March 2011) 799–807.

[4] Y. Gu, A. Lo, I. Niemegeers, A survey of indoor positioning systems for wireless personal networks, IEEE Commun. Surv. Tutorials 11 (1) (2009) 13–32 (First Quarter).

[5] G. López, V. Custodio, J.I. Moreno, LOBIN: E-textile and wireless-sensor-network-based platform for healthcare monitoring in future hospital environments, IEEE Trans. Inf. Technol. Biomed. 14 (6) (2010) 1446–1458. November.

[6] O.A. Postolache, J.M.D. Pereira, P.M.B.S. Girao, Smart sensors network for air quality monitoring applications, IEEE Trans. Instrum. Measur. 58 (9) (September 2009) 3253–3262.

[7] L. Tang, K.C. Wang, Y. Huang, F. Gu, Channel characterization and link quality assessment of IEEE 802.15.4-compliant radio for factory environments, IEEE Trans. Ind. Inf. 3 (2) (May 2007) 99–110.

[8] H. Dai, A.G. Chen, X.F. Gu, L. He, Localisation algorithm for large-scale and low-density wireless sensor networks, Electron. Lett. 47 (15) (2011) 881–883.

[9] E. Guerrero, H.G. Xiong, Q. Gao, A distributed range-free localization algorithm for wireless sensor networks based on a mobile robot, in: International Conference on Advanced Technologies for Communications, Hai Phong, Vietnam, October 2009, pp. 93–98.

[10] R. Ouyang, A. Wong, K. Woo, GPS localization accuracy improvement by fusing terrestrial TOA measurements, in: IEEE International Conference on Communications (ICC), May 2010, pp. 1–5.

[11] P. Kumar, L. Reddy, S. Varma, Distance measurement and error estimation scheme for RSSI based localization in wireless sensor networks, in: Fifth IEEE Conference on Wireless Communication and Sensor Networks (WCSN), Allahabad, India, December 2009, pp. 1–4.

[12] P.J. Voltz, D. Hernandez, Maximum likelihood time of arrival estimation for real-time physical location tracking of 802.11a/g mobile stations in indoor environments, in: Position Location and Navigation Symposium (PLANS), California, USA, April 2004, pp. 585–591.

[13] L. Kovavisaruch, K.C. Ho, Alternate source and receiver location estimation using TDOA with receiver position uncertainties, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), Pennsylvania, USA, March 2005, pp. iv/1065–iv/1068.

[14] P. Rong, M. Sichitiu, Angle of arrival localization for wireless sensor networks, Ann. IEEE Commun. Soc. Sensor Ad Hoc Commun. Networks, USA 1 (2006) 374–382.

[15] Z. Li, W. Dehaene, G. Gielen, A 3-tier UWB-based indoor localization system for ultra-low-power sensor networks, IEEE Trans. Wireless Commun. 8 (6) (June 2009) 2813–2818.

[16] R. Patro, Localization in wireless sensor network with mobile beacons, in: Proceedings of 23rd IEEE Convention of Electrical and Electronics Engineers in Israel, September 2004, pp. 22–24.

[17] L. Doherty, K. Pister, L. Ghaoui, Convex position estimation in wireless sensor networks, in: Proceedings of IEEE INFOCOM '01, Alaska, USA, vol. 3, April 2001, pp. 1655–1663.

[18] D. Niculescu, B. Nath, DV based Positioning in ad hoc networks, J. Telecommun. Syst. 22 (1) (2003) 267–280.

[19] S. Hou, X. Zhou, X. Liu, A novel DV-hop localization algorithm for asymmetry distributed wireless sensor networks, in: 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 4, 2010, pp. 243–248.

[20] Z. Zhang, G. Xu, Y. Li, S. Huang, DV-hop based self-adaptive positioning in wireless sensor networks, in: IEEE Conference on Wireless Communications Networking and Mobile Computing, 2009, pp. 1–4.

[21] J. Lee, W. Chung, E. Kim, I. Hong, Robust DV-hop algorithm for localization in wireless sensor network, in: International Conference on Control Automation and Systems, 2010, pp. 2506–2509.

[22] L. Gui, A. Wei, T. Val, A two-level range-free localization algorithm for wireless sensor networks, in: IEEE Conference on Wireless Communications Networking and Mobile Computing, 2010, pp. 1–4.

[23] L. Gui, A. Wei, T. Val, Improving localization accuracy using selective 3-anchor DV-hop algorithm, IN: IEEE Vehicular Technology Conference (VTC 2011-fall), September 2011, pp. 1–5.

[24] A. Sanjeev, B. Boaz, Computational Complexity: A Modern Approach. Cambridge, 2009.

[25] http://wsnet.gforge.inria.fr/tutorial.html.

[26] E. Hamida, G. Chelius, J. Gorce, On the complexity of an accurate and precise performance evaluation of wireless networks using simulations, in: Proceedings of the 11th ACM-IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2008), October 2008.

[27] IEEE standard 802.15.4-2009.

[28] IEEE standard 754.

[29] J. Choi, N. Kang, Y. Sung, S. Kim, Empirical ultra wide band path loss model in office environments, in: IEEE 63rd Vehicular Technology Conference, vol. 4, May 2006, pp. 1956–1960.

Linqing Gui obtained his Ph.D. in information science at INSA-Toulouse, France, in 2013. Now he is a teacher in Nanjing University of Science and Technology, Nanjing, China. In the last four years, he has published several articles on localization in wireless sensor networks. His research interests include wireless communication, sensor networks, and network simulations.

Thierry Val obtained his Ph.D. in computer science at Blaise Pascal University, Clermont–Ferrand, France, in 1993. In 1994, he became a lecturer at the University of Toulouse, where he currently teaches networks and computing systems. He obtained his HDR in 2002.He is now a professor for the University of Toulouse at the Blagnac Institute of Technology. He was sub-manager of the LATTIS laboratory, where he managed a research activity on wireless local networks and related protocols. He is now a member of IRIT-CNRS laboratory of Toulouse in IRT team.

Anne Wei graduated from the Department of Electronic Engineering of the Shanghai University in 1986 and completed her PhD in 1999 at Institute National des Telecommunications, France. After two years spent working for STERIA a computing system company and eight years as an Assistant professor at the University of Paris XII and two years as Professor at the University of Toulouse II, she is now a Professor at the CNAM Paris where she teaches computing science and networks. Her research interests include computer networks, 4G mobile systems and especially security, QoS and mobility in wireless network communications.

Rejane Dalce obtained her Ph.D. in information science at INSA-Toulouse, France, in 2013. Now she is a teacher in University of Toulouse, Toulouse, France. In the last four years, she has published several articles on localization in wireless sensor networks. Her research areas include wireless communication, sensor networks, and simulations.