



**HAL**  
open science

# An Adaptive Range-free Localization Protocol in Wireless Sensor Networks

Linqing Gui, Thierry Val, Anne Wei, Sami Taktak

► **To cite this version:**

Linqing Gui, Thierry Val, Anne Wei, Sami Taktak. An Adaptive Range-free Localization Protocol in Wireless Sensor Networks. IJAHUC - International Journal of Ad Hoc and Ubiquitous Computing, 2014, 15 (1-3), pp.1-20. 10.1504/IJAHUC.2014.059906 . hal-01120257

**HAL Id: hal-01120257**

**<https://hal.science/hal-01120257>**

Submitted on 25 Feb 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>  
Eprints ID : 12635

**To link to this article** : DOI :10.1504/IJAHUC.2014.059906  
URL : <http://dx.doi.org/10.1504/IJAHUC.2014.059906>

**To cite this version** : Gui, Linqing and Val, Thierry and Wei, Anne and Taktak, Sami *An Adaptive Range-free Localization Protocol in Wireless Sensor Networks*. (2014) International Journal of Ad Hoc and Ubiquitous Computing, vol. 15 (n° 1-2-3). ISSN 1743-8225

Any correspondence concerning this service should be sent to the repository administrator: [staff-oatao@listes-diff.inp-toulouse.fr](mailto:staff-oatao@listes-diff.inp-toulouse.fr)

---

# An Adaptive Range-free Localisation Protocol in wireless sensor networks

---

Linqing Gui\* and Thierry Val

NanJing University of Science and Technology,  
200140, NanJing, China

and

Toulouse University,  
CNRS-IRIT-IRT, IUT Blagnac, UT2, BP 60073,  
31703 Blagnac cedex, France

E-mail: guilingqing@gmail.com

E-mail: val@irit.fr

\*Corresponding author

Anne Wei and Sami Taktak

CNAM CEDRIC,  
292 Rue St Martin, 75141 Paris Cedex 03, France

E-mail: anne.wei@cnam.fr

E-mail: sami.taktak@cnam.fr

**Abstract:** It is well known that localisation is a fundamental issue for many wireless network applications. Without the need of additional ranging devices, the range-free localisation technology is a cost-effective solution for low-cost indoor and outdoor wireless sensor networks. However, we noted that most existing algorithms were only studied using tools like MATLAB neglecting possible problems in a real wireless network context such as frame collision and node synchronisation. Thus, we propose an Adaptive Range-free Localisation Protocol (ALP) based on IEEE 802.15.4 standard, which can evaluate localisation algorithms. Using our localisation protocol, we investigate and compare the performance of our new approach to some existing range-free algorithms in terms of localisation accuracy, mobility, synchronisation and overhead. Our results show that our Mid-perpendicular, Checkout DV-hop and Selective 3-Anchor DV-hop algorithms support robust and dynamic localisation in the context of our adaptive localisation protocol.

**Keywords:** wireless sensor networks; localisation; range-free; algorithm; protocol.

**Reference** to this paper should be made as follows: Gui, L., Val, T., Wei, A. and Taktak, S. (xxxx) 'An Adaptive Range-free Localisation Protocol in wireless sensor networks', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. x, No. x, pp.xxx-xxx.

**Biographical notes:** Linqing Gui obtained his PhD in Information science at University of Toulouse, France, in 2013. He is now a lecturer candidate of university in China. His research interests cover PHY layer and MAC layer key techniques in wireless sensor networks.

Thierry Val obtained his PhD 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.

Sami Taktak obtained his PhD in Computer Science at Pierre and Marie-Curie University, Paris, France, in 2009. In 2012, he became a Lecturer at the Conservatoire National des Arts et Métiers, where he teaches computer sciences. He is currently a member of the Sempia Team of the Cédric laboratory where he pursues his work on formals methods.

## 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 be used for parking space detection (Vishnubhotla et al., 2010), security surveillance (Zhang et al., 2011), indoor object tracking (Lee and Chung, 2011) or monitoring services (López et al., 2010; Postolache et al., 2009; Tang et al., 2007). Note that it is important for sensor data to be combined with position information in many applications (Vishnubhotla et al., 2010; Lee and Chung, 2011; Gu et al., 2009; López et al., 2010). The position of sensors can also help to facilitate routing as well as determining the quality of coverage and achieving load balancing. Therefore, localisation has become a fundamental element in wireless sensor networks (Vishnubhotla et al., 2010; Zhang et al., 2011; Lee and Chung, 2011; Gu et al., 2009; López et al., 2010; Postolache et al., 2009; Tang et al., 2007; Dai et al., 2011; Guerrero et al., 2009).

The existing localisation techniques can be generally categorised into two types: range-based and range-free. Range-based schemes (Ouyang et al., 2010; Kumar et al., 2009; Voltz and Hernandez, 2004; Kovavisaruch and Ho, 2005; Rong and Sichitiu, 2006) need first to precisely measure the range information (the distance or the angle) between concerned sensor nodes, and then to calculate the desired position based on trilateration or triangulation approaches. The ranging methods typically use Received Signal Strength Indicator (RSSI) (Kumar et al., 2009), Time of Arrival (TOA) (Voltz and Hernandez, 2004), Time Difference of Arrival (TDOA) (Kovavisaruch and Ho, 2005) and Angle of Arrival (AOA) (Rong and Sichitiu, 2006). Global Positioning System (GPS) (Ouyang et al., 2010) is the most well-known range-based technique using TOA or TDOA. However, 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 localisation 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 (Li et al., 2009). The range-based techniques have two major drawbacks. First, the range information is very easily affected by multipath fading, noise and environmental variations. Second, additional ranging devices are usually needed which consume more energy and increase the overall cost.

While range-based scheme uses the distance or angle between nodes, the range-free scheme uses connectivity information between nodes. In this scheme, the nodes aware of their positions are called anchors, while other nodes are called normal nodes. Anchors are fixed, while normal nodes are usually mobile. Normal nodes first gather their connectivity information as well as positions of

anchors. Then they calculate their own positions. Since no range information is needed, range-free scheme can be implemented on low-cost wireless sensor networks. Another advantage of range-free scheme is its robustness; connectivity information is not easily affected by the environment. As a result, we focus on the range-free scheme.

The typical range-free algorithms include Centroid (Patro, 2004), CPE (Convex Position Estimation) (Doherty et al., 2001), and DV-hop (Distance Vector-hop) (Niculescu and Nath, 2003). Centroid and CPE have low complexity, but they require a normal node to have at least three neighbour anchors. The DV-hop algorithm can handle the case where a normal node has less than three neighbour anchors. It should be noted that, these localisation algorithms are not accurate enough. Then, in order to improve localisation accuracy, we have proposed several new algorithms such as Mid-perpendicular (Gui et al., 2010), Checkout DV-hop (Gui et al., 2010) and Selective 3-Anchor DV-hop (Gui et al., 2011).

However a common problem of range-free localisation algorithms has been neglected. That is, performances of these existing algorithms are studied by using tools like MATLAB which doesn't consider possible problems in a real wireless network context such as frame collisions and node synchronisation. Thus, we propose in this paper an ALP based on IEEE 802.15.4 standard, which can be used to evaluate localisation algorithms.

The main contributions of this paper are:

- We propose ALP-3 protocol which is used to implement DV-hop based algorithms. In this novel protocol we design new data payload formats, and 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 based algorithms.
- We also propose ALP+3 protocol to implement other algorithms such as Centroid, CPE and Mid-perpendicular. In this protocol normal nodes broadcast their localisation request to neighbour nodes. Then their neighbour anchors respond by sending back anchors' positions. Here, our E-CSMA/CA method is also used to reduce frame collisions.
- We suggest an adaptive approach named as ALP protocol which combines ALP-3 protocol and ALP+3 protocol. The basic principle of this adaptive protocol is as follows: given the ratio of anchors, the network administrator can estimate the network overhead of both ALP-3 and ALP+3 protocols. Thus, the maximum acceptable network overhead has its corresponding maximum ratio of anchors which is defined as the threshold of ratio of anchors ' $RA_{thres}$ '. When the ratio of anchors is lower than  $RA_{thres}$ , *ALP-3 protocol* needs to be used; but when the ratio of anchors is higher than  $RA_{thres}$ , in order to avoid a large number of network traffic, *ALP+3 protocol* should be used.

- Based on our protocols, using the network simulator WSNnet, we simulate the concerned range-free localisation algorithms in the IEEE 802.15.4 wireless network. Comparative network simulation results are presented and analysed in terms of localisation accuracy, overhead, node mobility and node synchronisation.

The rest of this paper is organised as follows. Section 2 introduces existing typical range-free localisation algorithms. Section 3 presents our improved algorithms such as Mid-perpendicular, Checkout DV-hop and Selective 3-Anchor DV-hop. In Section 4, our ALP is introduced. In Section 5, simulation results and analysis are given. Finally we give our conclusion and perspective in Section 6.

## 2 Related works

In this section we introduce and analyse the typical range-free localisation algorithms. Among them some algorithms that we mark as A+3 algorithms require a normal node to have at least three neighbour anchors. But others that we denote by A-3 algorithms do not have this constraint.

### 2.1 A+3 algorithms: Range-free algorithms with at least 3 neighbour anchors

When some range-free algorithms like Centroid and CPE localise a normal node, it is necessary for the normal node to have at least 3 anchors at range.

Centroid algorithm was proposed by Bulusu (Patro, 2004). The basic principle is to look at the centroid point of neighbour anchors as the estimated position of the normal node. The author chooses a simple radio propagation model, which fits quite well for outdoor environment. In this model there are two assumptions: the first is perfect spherical radio propagation, and the second is identical transmission range for all radios.

Suppose that in the network there are  $m$  anchors situated at known positions  $A_1(x_1, y_1), A_2(x_2, y_2) \dots A_m(x_m, y_m)$ . All these anchors have the same communication range denoted by  $R$ . Their transmission areas have an overlap. Inside the overlap locates the normal node  $N_x$ . That means all these  $m$  anchors are the neighbour anchors of  $N_x$ . Then  $N_x$  localises itself at the centroid of these  $m$  anchors:

$$\begin{cases} x_{cen} = (x_1 + x_2 + \dots + x_m) / m \\ y_{cen} = (y_1 + y_2 + \dots + y_m) / m \end{cases} \quad (1)$$

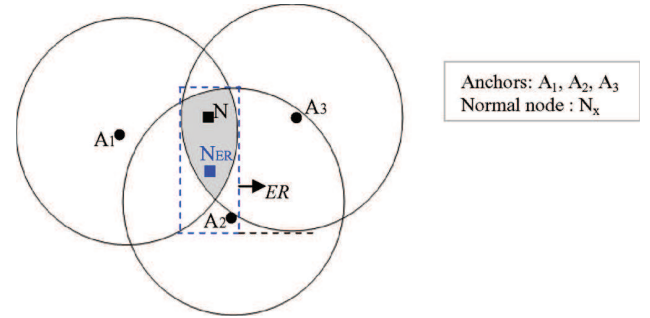
The CPE (Convex Position Estimation) algorithm was proposed by Doherty (Doherty et al., 2001). The authors of CPE algorithm first provide an optimisation concept and then they propose to estimate positions of normal nodes using the results of a joint optimisation problem. The CPE algorithm is a centralised localisation scheme because the resource-limited normal node is unable to do numerous and

complex calculations required by optimisation process. Thus, the original CPE algorithm scales poorly when the network is large.

However, a simplified and distributed version of CPE algorithm has been proposed by some researchers (Sheu et al., 2006, 2008). The simplified CPE algorithm defines an Estimated Rectangle (ER) which bounds the communication range of anchors, as shown in Figure 1. Its centre point, denoted by  $N_{ER}$ , is the estimated position of the simplified CPE algorithm, calculated as:

$$x_{ER} = \frac{\min_i x_i + \max_i x_i}{2}, \quad y_{ER} = \frac{\min_i y_i + \max_i y_i}{2}.$$

**Figure 1** Example of a simplified CPE algorithm (see online version for colours)



Centroid and the simplified CPE algorithms have both low network overhead and low calculation complexity. But their accuracy performance is not so good. It should be noted that the above algorithms work under the condition that a normal node has at least 3 neighbour anchors. However if the density of anchors is not very high in a network, some normal nodes may have less than 3 neighbour anchors. In this case DV-hop based algorithms should be used.

### 2.2 A-3 algorithms: Range-free algorithms with less than 3 neighbour anchors

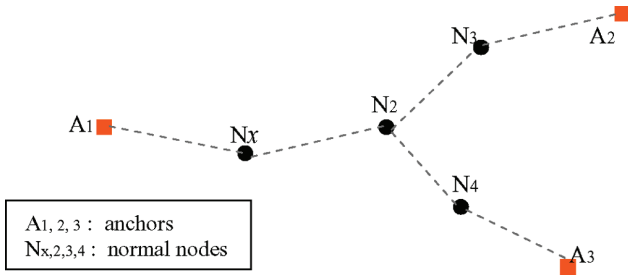
A-3 algorithms do not require a normal node to have at least 3 neighbour anchors. Here, they refer to DV-hop based algorithms. In this subsection we first introduce the original DV-hop algorithm and then some typical DV-hop based algorithms such as DDV-hop (Differential DV-hop), Self-adaptive DV-hop and Robust DV-hop.

#### 2.2.1 DV-hop algorithm

The DV-hop algorithm was proposed by Niculescu (Niculescu and Nath, 2003). It is a suitable solution for normal nodes having less than 3 neighbour anchors. As shown in Figure 2, although the normal node  $N_x$  has only one neighbour anchor  $A_1$ ,  $N_x$  can use DV-hop algorithm for localisation. It means that  $N_x$  first gathers at least three anchors' positions via other normal nodes, then based on the hop counts and distances between  $N_x$  and anchors,  $N_x$  estimates its position. The algorithm consists of the following three steps.



**Figure 2** Example of topology in DV-hop algorithm (see online version for colours)



*Step 1:* each anchor  $A_i$  broadcasts through the network a message containing the position of  $A_i$  and a hop count field initialised as 0. This hop count value will increase with the number of hops during the broadcast of the message in the network. On the first reception of the message, every node  $N$  (either anchor or normal node) records the position of  $A_i$ , and initialises  $hop_i$  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,  $N$  updates  $hop_i$ . 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 the nodes in the network can get the minimum hop count to each anchor.

*Step 2:* 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  $dph_i$ . Once  $dph_i$  is calculated, it will be broadcasted by  $A_i$ .

*Step 3:* when receiving  $dph_i$ , the normal node  $N_x$  multiplies  $hop_{i,N_x}$  (its hop count to  $A_i$ ) by  $dph_i$ , so that  $N_x$  obtains its distance to each anchor  $A_i$ , denoted as  $d_{i,N_x}$ . Here,  $i \in \{1, 2, \dots, m_d\}$ , if we assume that there are  $m_d$  anchors. Then each normal node  $N_x$  can calculate its estimated position  $N_{DV-hop}$  by trilateration. The detail of the calculations of  $N_{DV-hop}$  can be found in Niculescu and Nath (2003).

Although DV-hop algorithm can localise normal nodes which have less than three neighbour anchors, its localisation accuracy needs to be improved (Hou et al., 2010; Zhang et al., 2009; Lee et al., 2010). Thus, many algorithms have been proposed in recent years. In the following, several typical algorithms will be analysed.

### 2.2.1 Typical DV-hop based algorithms

In this section we describe a few DV-hop based localisation algorithms such as DDV-hop (Differential DV-hop) (Hou et al., 2010), Self-adaptive DV-hop (Zhang et al., 2009), and Robust DV-hop (Lee et al., 2010).

*DDV-hop:* This algorithm modifies 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

Hou et al. (2010). In Step 3, DDV-hop and DV-hop differ on the calculation of the estimated distance between a normal node  $N_x$  and each anchor  $A_i$ . That is,  $N_x$  uses its own distance-per-hop value denoted as  $dph_{N_x}$  to replace the anchors' distance-per-hop  $dph_i$ . Here,  $dph_{N_x}$  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.

*Self-Adaptive DV-hop:* This algorithm is composed of two methods. Because the second method needs 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 modifies Step 3. At Step 3, when a normal node  $N_x$  calculates its estimated distance to  $A_i$ ,  $N_x$  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. 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  $N_x$ 's hop count to  $A_i$ . The more hops between  $N_x$  and  $A_i$ , the smaller value assigned to the weighting coefficient of  $dph_i$ .

*Robust DV-hop:* a Robust DV-hop (RDV-hop) algorithm is proposed in Lee et al. (2010). Different from the above two algorithms, in order to replace  $dph_i$  (the average distance per hop of  $A_i$ ), RDV-hop algorithm defines a distance-per-hop value between  $N_x$  and  $A_i$ , denoted by  $dph_{N_x,i}$ . And  $dph_{N_x,i}$  is calculated as the weighted sum of the distance-per-hop values between  $A_i$  and every other anchor  $A_k$ . Here the distance-per-hop between  $A_i$  and  $A_k$  is denoted as  $dph_{i,k}$ . In the calculation of  $dph_{N_x,i}$ , the weighing coefficient of  $dph_{i,k}$  will have the maximum value, if  $N_x$  is one node on the shortest path between  $A_i$  and  $A_k$ .

These typical DV-hop based algorithms use weighing methods to determine weighted distance-per-hop values for each normal node. However, in order to get a more accurate distance-per-hop value, sometimes additional information is demanded such as differential error in Hou et al. (2010) and hop counts between anchors in Lee et al. (2010). Broadcasting this additional information always 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 specifically designed rather than randomly obtained. For example, in Zhang et al. (2009) the anchors are distributed at the corners of the simulation area and the normal nodes are regularly distributed inside the area. Thus, in order to obtain a better accuracy without increasing the network overhead, we are motivated to provide improved methods which will be presented in the next section.

## 3 Improved range-free localisation algorithms

According to the previous analysis on the typical range-free algorithms, when a normal node has at least 3 neighbour

anchors, it can localise itself using algorithms such as Centroid, CPE (in the paper from now on, it refers to simplified version of CPE). On the contrary, when a normal node has less than 3 neighbour anchors, the available localisation algorithms are only the DV-hop based algorithms.

This encourages us to categorise normal nodes into two types according to the number of neighbour anchors: the normal nodes having at least 3 neighbour anchors are noted as ‘A+3’ nodes, while others are ‘A–3’ nodes. For each type of normal nodes, we will present our improved localisation methods in the following.

### 3.1 Improved range-free algorithm for A+3 nodes

For A+3 normal nodes, Centroid and CPE are popular algorithms because of their low communication and computation cost, regardless of their inaccuracy. Our aim is to propose a new algorithm which can achieve a higher accuracy, at the cost of higher calculation complexity. The algorithm we have proposed is Mid-perpendicular (Gui et al., 2010). The basic principle of this algorithm is to find the centre of anchors communication overlap by perpendicular lines and consider this centre as the estimated position.

We first investigate the case when a normal node has only 3 neighbour anchors as shown in Figure 3(a). ‘Line1’ is the mid-perpendicular of the line connecting the anchors  $A_2$  and  $A_3$ . That means Line1 passes the middle point between  $A_2$  and  $A_3$  and crosses the line (which connects  $A_2$  and  $A_3$ ) at a right angle. According to the symmetry, Line1 goes through the centre of the overlap region. In the same manner, Line2 is the mid-perpendicular of the line connecting  $A_1$  and  $A_3$ , while Line3 is the mid-perpendicular of the line connecting  $A_1$  and  $A_2$ . Both Line2 and Line3 go through the centre of the overlap region. Thus, the cross point of the three mid-perpendiculars (Line1, Line2 and Line3) can be regarded as the centre of overlap. This cross point is denoted as  $N_{mid}$  which is also the estimated position of Mid-perpendicular algorithm.

If the coordinates of the three anchors ( $A_1$ ,  $A_2$  and  $A_3$ ) are respectively  $(x_1, y_1)$ ,  $(x_2, y_2)$ , and  $(x_3, y_3)$ , the cross point of the above two mid-perpendiculars denoted by  $N_{mid}$  ( $x_{mid}, y_{mid}$ ) can then be calculated as:

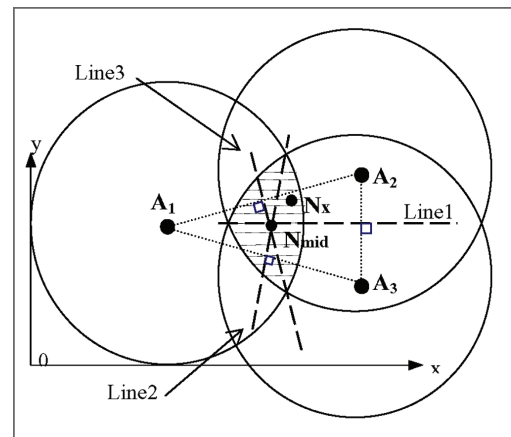
$$\begin{cases} x_{mid} = \frac{(x_1^2 - x_2^2)(y_3 - y_1) + (x_1^2 - x_3^2)(y_1 - y_2) + (y_1 - y_2)(y_2 - y_3)(y_3 - y_1)}{2 [y_1(x_2 - x_3) + y_2(x_3 - x_1) + y_3(x_1 - x_2)]} \\ y_{mid} = \frac{(y_1^2 - y_2^2)(x_3 - x_1) + (y_1^2 - y_3^2)(x_1 - x_2) + (x_1 - x_2)(x_2 - x_3)(x_3 - x_1)}{2 [x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)]} \end{cases} \quad (2)$$

It should be noted that there is one condition for the above derivation:  $N_x$ 's 3 neighbour anchors ( $A_1$ ,  $A_2$  and  $A_3$ ) form an acute triangle where all the angles are less than 90 degrees. However, if the 3 neighbour anchors form a right triangle or an obtuse triangle, then the calculation of  $N_{mid}$  will be much simpler: in this case,  $N_{mid}$  is the middle point of the longest side in the triangle.

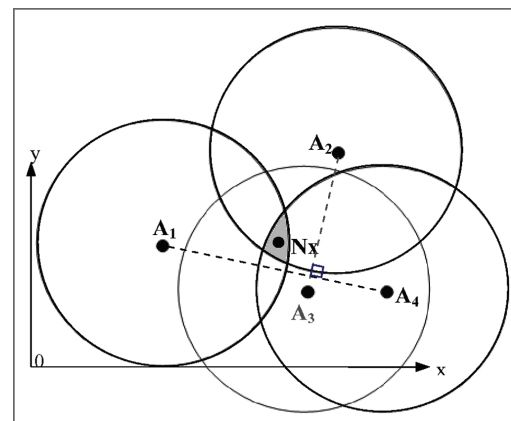
Then we investigate the case when a normal node has more than 3 neighbour anchors. Suppose that there are  $m$  neighbour anchors for the normal node  $N_x$ , with  $m > 3$ . We have found that the overlap communication region of all the  $m$  anchors is contributed mainly by three anchors. In Figure 3 (b), the example has 4 neighbour anchors. From the figure, we can see that the overlap region formed by all the 4 anchors is actually the overlap of the three anchors  $A_1$ ,  $A_2$ , and  $A_4$ . These three anchors have the following characteristics:

- Two of them have the longest distance compared with distances between any two of the entire anchors. That is because the two most distant anchors have the smallest overlap. In the example, the two most distant anchors are  $A_1$  and  $A_4$ .
- The third anchor is farthest to the line connecting the two most distant anchors. In this example, since the two most distant anchors are  $A_1$  and  $A_4$ , the anchors except them are  $A_2$  and  $A_3$ . Compared with  $A_3$ ,  $A_2$  has a longer distance to the line connecting  $A_1$  and  $A_4$ . Thus, the third anchor is  $A_2$ .

**Figure 3** Mid-perpendicular in different cases: (a) case with 3 neighbour anchors (b) case with more than 3 neighbour anchors (see online version for colours)



(a)



(b)

Thus we know how to estimate the position of  $N_x$ . First,  $N_x$  calculates the distance between any two anchors. Since there are  $m$  neighbour anchors, there will be  $C_m^2$  distances in total. Comparing these distances,  $N_x$  can find out the two farthest anchors, denoted by  $A_i$  and  $A_k$ . Then, among all other anchors except  $A_i$  and  $A_k$ ,  $N_x$  finds out the anchor which has the longest distance to the line connecting  $A_i$  and  $A_k$ . This anchor is denoted as  $A_j$ . Thus,  $A_i$ ,  $A_k$  and  $A_j$  are the three anchors which contribute the overlap of all  $m$  anchors. Finally, using equation (2),  $N_x$  calculates the centre of the overlap formed by  $A_i$ ,  $A_k$  and  $A_j$ , thus obtaining the final estimated position.

The simulation results in Gui et al. (2010) show that, with nodes randomly distributed, our Mid-perpendicular algorithm has a better localisation accuracy of about 15% than Centroid and CPE algorithms.

### 3.2 Improved range-free algorithms for A-3 nodes

In the previous section, we introduce our improved method to localise A+3 normal nodes. Now, we would like to focus on the case with A-3 normal nodes. Usually, in a network there are always a few anchors and much more normal nodes. As a result, most normal nodes will be A-3 nodes, having less than 3 neighbour anchors. DV-hop algorithm is frequently used to localise A-3 nodes. However, its accuracy should be improved. Thus, we have proposed Checkout DV-hop and Selective 3-Anchor DV-hop algorithms. In the following, we will introduce the principle of these two algorithms.

#### 3.2.1 Checkout DV-hop algorithm

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

$$d_{i,N_x} = \text{hop}_{i,N_x} \times \text{dph}_i, \quad i = 1, 2, \dots, m \quad (3)$$

where  $d_{i,N_x}$  is the approximate distance between  $N_x$  and  $A_i$ ,  $\text{hop}_{i,N_x}$  is the minimal hop number between  $N_x$  and  $A_i$ , and  $\text{dph}_i$  is the approximate average distance per hop of  $A_i$ .

Since  $d_{i,N_x}$  is an important element for calculating the position of the normal node  $N_x$  (Niculescu and Nath, 2003), it has a considerable influence on the accuracy of DV-hop. We denote the true distance from  $N_x$  to  $A_i$  by  $d_{i,N_x\text{True}}$ , and the difference between  $d_{i,N_x\text{True}}$  and  $d_{i,N_x}$  by  $\Delta d_{i,N_x}$ , where obviously  $\Delta d_{i,N_x}$  is one reason for the inaccuracy of DV-hop. If we denote  $\Delta \text{dph}_i$  by the difference between  $\text{dph}_i$  and its true value, then from equation (3) we have:

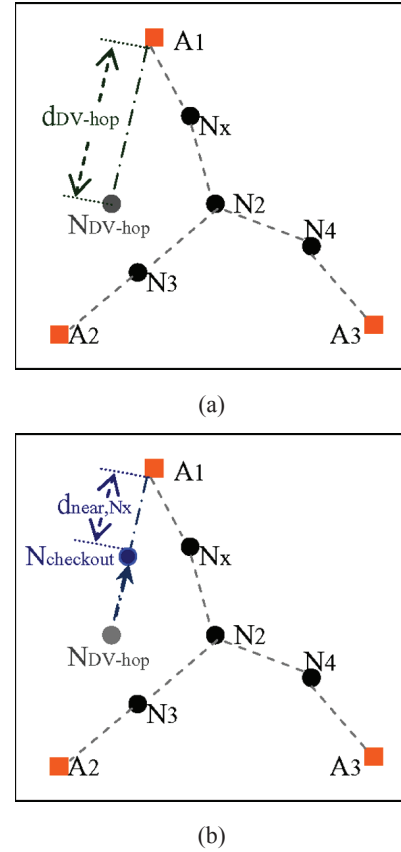
$$\Delta d_{i,N_x} = \text{hop}_{i,N_x} \times \Delta \text{dph}_i \quad (4)$$

Equation (4) indicates that when  $\text{hop}_{i,N_x}$  increases,  $\Delta d_{i,N_x}$  also increases, and the accuracy of DV-hop decreases. If  $A_{\text{near}}$  is the nearest anchor to  $N_x$  among all anchors  $A_1 A_2 \dots A_m$ , then correspondingly  $\text{hop}_{\text{near},N_x}$  is the smallest, so that  $\Delta d_{\text{near},N_x}$  is the smallest position error. So we can conclude that, compared to other anchors, the distance from the normal

node  $N_x$  to its nearest anchor  $A_{\text{near}}$ , denoted by  $d_{\text{near},N_x}$ , has the highest reliability in terms of precision. Checkout DV-hop algorithm will make best use of this concept to improve the accuracy using the most reliable information available.

Now we illustrate the principle of Checkout DV-hop algorithm. It adds a checkout step to DV-hop algorithm, as shown in Figure 4. For the purpose of comparison, Figure 4(a) shows the result of DV-hop without ‘checkout’, while Figure 4(b) shows the impact of our checkout step. As shown in Figure 4(a), the normal node  $N_x$  uses DV-hop to obtain its estimated position at  $N_{\text{DV-hop}}$  with its coordinates denoted by  $(x', y')$ . Then it calculates the distance between  $N_{\text{DV-hop}}$  and  $A_{\text{near}}$  (here,  $A_{\text{near}}$  is  $A_1$ ), denoted by  $d_{\text{DV-hop}}$ . Note that  $N_x$  has used equation (3) to evaluate its approximate distance to the nearest anchor  $A_{\text{near}}$ , denoted by  $d_{\text{near},N_x}$ .

**Figure 4** Principle of checkout DV-hop: (a) DV-hop and (b) checkout DV-hop (see online version for colours)



The purpose of the checkout step is to change the estimated position from  $N_{\text{DV-hop}}$  (see Figure 4(b)) to a new one called  $N_{\text{checkout}}$ , whose distance to  $A_{\text{near}}$  is  $d_{\text{near},N_x}$ . To achieve this, the easiest and quickest way is to change the position along the line connecting  $N_{\text{DV-hop}}$  and  $A_{\text{near}}$ .  $N_{\text{checkout}}$  is on the line from  $N_{\text{DV-hop}}$  to  $A_{\text{near}}$ , and the distance between  $N_{\text{checkout}}$  and  $A_{\text{near}}$  is  $d_{\text{near},N_x}$ . The position of  $A_{\text{near}}$  is  $(x_{\text{A}_{\text{near}}}, y_{\text{A}_{\text{near}}})$  and  $N_{\text{DV-hop}}$  is located at  $(x', y')$ , therefore the position of  $N_{\text{checkout}}$ , denoted by  $(x_{\text{checkout}}, y_{\text{checkout}})$  can be derived as follows.  $N_{\text{checkout}}$  is chosen as our node estimated position.



$$\begin{cases} x_{checkout} = x' - \left( \frac{d_{DV-hop} - d_{near, N_x}}{d_{DV-hop}} \right) \times (x' - x_{A_{near}}) \\ y_{checkout} = y' - \left( \frac{d_{DV-hop} - d_{near, N_x}}{d_{DV-hop}} \right) \times (y' - y_{A_{near}}) \end{cases} \quad (5)$$

In Gui et al. (2010), simulations have been done using MATLAB with the nodes randomly distributed in a square area. Results have proved that our Checkout DV-hop algorithm is about 15% more accurate than DV-hop algorithm.

### 3.2.2 Selective 3-anchor DV-hop algorithm

Although Checkout DV-hop algorithm is simple, its accuracy improvement is not so considerable. We observe that the accuracy can be improved if we find three anchors being at 'best' positions. Thus, we have proposed Selective 3-Anchor DV-hop algorithm. First, this algorithm generates a group of candidates. Then, from this pool, it chooses the best 3-Anchor group based on its connectivity vector.

Let us consider a network with  $m_d$  anchors  $A_1 A_2 \dots A_{m_d}$ . Through DV-hop algorithm, a normal node  $N_x$  can obtain  $hop_{i, N_x}$ , which is its minimum hop count to each anchor  $A_i$ , as well as  $d_{i, N_x}$ , 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  $m_d$  estimated distance values  $d_{1, N_x} d_{2, N_x} \dots d_{m_d, N_x}$ . So, the quality of these estimated values has a great influence on the accuracy of DV-hop.

In fact, instead of using all  $m_d$  estimated values, three estimated distance values to three different anchors are sufficient for  $N_x$  to calculate its position. For example, we use  $d_{i, N_x}, d_{j, N_x}, d_{k, N_x}$ , 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  $N_x$  by  $(x, y)$ , and the positions of  $A_i, A_j, A_k$  respectively by  $(x_i, y_i), (x_j, y_j), (x_k, y_k)$ , then we can have the following equation:

$$\begin{cases} (x - x_i)^2 + (y - y_i)^2 = d_{i, N_x}^2 \\ (x - x_j)^2 + (y - y_j)^2 = d_{j, N_x}^2 \\ (x - x_k)^2 + (y - y_k)^2 = d_{k, N_x}^2 \end{cases} \quad (6)$$

Solving (6) based on MLE method (Maximum Likelihood Estimation), we can get a 3-anchor estimated position of  $N_x$ , denoted as  $N_{<i,j,k>}$  ( $x_{<i,j,k>}, y_{<i,j,k>}$ ). It is calculated as:

$$\begin{aligned} N_{<i,j,k>} : \begin{bmatrix} x_{<i,j,k>} \\ y_{<i,j,k>} \end{bmatrix} &= C^{-1}B, \\ \text{and } C &= -2 \times \begin{bmatrix} x_i - x_k & y_i - y_k \\ x_j - x_k & y_j - y_k \end{bmatrix}, \\ B &= \begin{bmatrix} d_{i, N_x}^2 - d_{k, N_x}^2 - x_i^2 - y_i^2 + x_k^2 + y_k^2 \\ d_{j, N_x}^2 - d_{k, N_x}^2 - x_j^2 - y_j^2 + x_k^2 + y_k^2 \end{bmatrix} \end{aligned} \quad (7)$$

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 collinear. Otherwise, matrix  $C$  will be singular.

The principle of Selective 3-Anchor DV-hop algorithm is to select the most accurate 3-anchor estimated position and consider it as the final estimated position. Here the criterion for the selection is the connectivity of  $N_x$ .

In DV-hop algorithm, the connectivity of  $N_x$  is specified as the minimum hop counts between  $N_x$  and anchors. For example, if there are  $m_d$  anchors and the minimum hop count from  $N_x$  to each anchor  $A_i$  is  $hop_{i, N_x}$ , then the connectivity of  $N_x$  is the array  $[hop_{1, N_x}, hop_{2, N_x} \dots hop_{m_d, N_x}]$ . In Gui et al. (2011), we have given the relation between connectivity and distance: a smaller connectivity difference between two nodes will result in a smaller distance between them. According to this relationship, the 3-anchor estimated position which has the smallest connectivity difference to  $N_x$  must be the nearest to  $N_x$ . So, 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  $N_{<i,j,k>}$  is still unknown. We therefore give the following method to calculate the hop count between  $N_{<i,j,k>}$  and each anchor.

Through the first two steps of DV-hop,  $N_x$  can obtain anchors' positions as well as its minimum hop counts to all anchors. Then, based on equation (7),  $N_x$  can get its 3-anchor estimated positions, each of them denoted by  $N_{<i,j,k>}$ . So  $N_x$  can calculate the distances between  $N_{<i,j,k>}$  and each anchor  $A_t$ , denoted by  $d_{<i,j,k>, t}$ . Then the problem of calculating the hop count between  $N_{<i,j,k>}$  and  $A_t$  becomes the problem of calculating the distance per hop. Indeed, if  $N_x$  knows the distance per hop between  $N_{<i,j,k>}$  and  $A_t$ , denoted by  $dph_{<i,j,k>, t}$ , then  $N_x$  can calculate the hop count between  $N_{<i,j,k>}$  and  $A_t$  according to equation (8) where  $hop_{<i,j,k>, t}$  is the hop count between  $N_{<i,j,k>}$  and  $A_t$ :

$$hop_{<i,j,k>, t} = \frac{d_{<i,j,k>, t}}{dph_{<i,j,k>, t}} \quad (8)$$

Then we must then find a method to estimate the value of  $dph_{<i,j,k>, t}$ . In fact, all the distance-per-hop information that  $N_x$  has obtained are anchors' distance-per-hop values:  $dph_1, dph_2, \dots, dph_{m_d}$ , including the distance per hop of  $A_t$  denoted as  $dph_t$ . Thus, we need to estimate  $dph_{<i,j,k>, t}$  based on the anchors' distance-per-hop values.

In order to get an approximate value of  $dph_{<i,j,k>, t}$ , three kinds of position relation between  $N_{<i,j,k>}$  and its nearest anchor  $A_{near}$  are considered based on their distance. In the first case, the distance between  $N_{<i,j,k>}$  and  $A_{near}$  is so small that we can use the distance-per-hop value of  $A_{near}$  (denoted by  $dph_{near}$ ) as an approximate value of  $dph_{<i,j,k>, t}$ . Here, as an example, we can set the distance threshold to 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  $N_{<i,j,k>}$  and  $A_{near}$  is so large that we can only use  $dph_t$  as an approximate value of  $dph_{<i,j,k>, t}$ . Here, also as example, the threshold of distance is set to 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 to the average of  $dph_{near}$  and  $dph_t$ . These three cases are shown in Figure 5. In this figure,  $N_p$  and  $N_q$  are two other normal nodes which connect  $N_x$  and  $A_t$ .

Summarising the three cases, we can estimate the value of  $dph_{\langle i,j,k \rangle, t}$  as follow:

$$dph_{\langle i,j,k \rangle, t} \approx \begin{cases} dph_{near}, & \text{when } d_{near} < range/2 \\ dph_t, & \text{when } d_{near} > range \\ (dph_{near} + dph_t)/2, & \text{others} \end{cases} \quad (9)$$

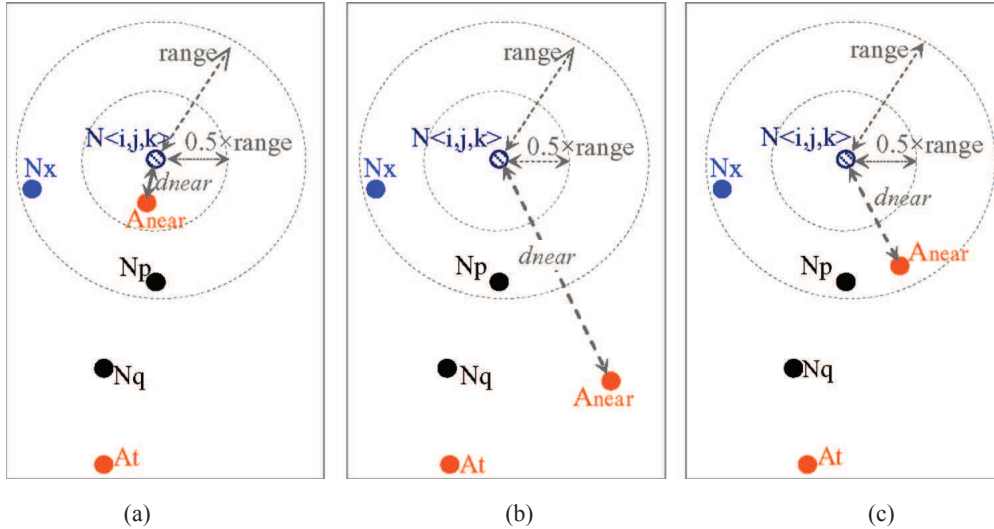
where  $d_{near}$  is the distance between  $N_{\langle i,j,k \rangle}$  and  $A_{near}$ ,  $dph_{near}$  is the distance per hop of  $A_{near}$ .

Using (8) and (9),  $N_x$  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

$$\sum_{t=1}^m |hop_{\langle i,j,k \rangle, t} - hop_t|$$

Then, from the  $c_m^3$  3-anchor estimated positions,  $N_x$  selects the position having the smallest connectivity difference as the final estimated position.

**Figure 5** Three kinds of relative positions: (a)  $d_{near} < range/2$ ; (b)  $d_{near} > range$  and (c)  $range/2 < d_{near} < range$  (see online version for colours)



#### 4 Adaptive Range-free Localisation Protocol (ALP)

During the verification process of our three new algorithms, we noted that most of the existing algorithms were only studied using tools like MATLAB which neglects possible problems of a real network. For example, DV-hop based algorithms need the broadcasts of position related information throughout the network leading to problems such as collisions and link congestion that must be solved by a new localisation protocol. Having found no such protocol, we propose an adaptive range-free protocol (ALP) which is a combination of ALP-3 protocol and ALP+3 protocol. Our new localisation protocol is based on

The procedure of our Selective 3-Anchor DV-hop algorithm is summarised 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 collinear anchors to form a 3-anchor group, and correspondingly generates a 3-anchor estimated position. Then, based on equations (8) and (9),  $N_x$  calculates the connectivity of each 3-anchor estimated position. Finally,  $N_x$  chooses the best 3-anchor estimated position which has the smallest connectivity difference to  $N_x$ .

We should note an exceptional case concerning very low ratio of anchors. For example, in a network with 100 nodes with only five of them being anchors. In this case, some normal nodes may have the same connectivity, thus the algorithm will be quite inaccurate. In this case, checkout DV-hop algorithm is recommended.

The simulation results by MATLAB in Gui et al. (2011) prove that, when the ratio of anchors is more than 10%, Selective 3-Anchor DV-hop algorithm achieves much better precision than other algorithms (Niculescu and Nath, 2003; Hou et al., 2010; Zhang et al., 2009; Lee et al., 2010; Gui et al., 2010). The improvement of precision can be from 20% to 57% depending on the compared algorithms and the different scenarios.

the IEEE 802.15.4 standard with an ad-hoc network topology and non-slotted CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) medium access method. We choose non-slotted CSMA/CA since it's well suited for ad-hoc topology.

##### 4.1 ALP-3 protocol

Our ALP-3 protocol can be used to implement DV-hop based algorithms which localise the normal nodes with less than 3 neighbour anchors. In the following subsections, we will introduce ALP-3 protocol, including the formats of data payload, the improved collision reduction methods and the procedure of the protocol.

### 4.1.1 Proposed formats of data payload

Like DV-hop algorithm, ALP-3 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 message exchange at the first two steps.

Conforming to the MAC general frame format specified in IEEE standard 802.15.4–2009, frames in ALP-3 protocol consist of three basic fields: MHR (MAC header), MAC payload and MFR (MAC footer), as shown in Table 1. We should mention that destination and source addresses use 16-bit short format. Since all frames in ALP-3 protocol are to be broadcasted, the destination address should be  $0x\text{ffff}$ .

**Table 1** Format of data frame in ALP-3 protocol

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

Two formats of data payload are proposed for the first two steps of ALP-3 protocol.

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 2. In Table 2, the data payload is composed of four parts: ‘Data Type’, ‘ $x_i$ ’, ‘ $y_i$ ’ and ‘HopCount’. Data Type identifies the type of information that the frame contains. In fact, in DV-hop algorithm, each anchor  $A_i$  only needs to broadcast two types of information: the position and the distance-per-hop. So we define that Data Type (1 bit) is ‘0’ for a position frame, or ‘1’ for a distance-per-hop frame. Here, it is ‘0’ since ‘ $frame\_pos_i$ ’ is a position frame. ‘HopCount’ is the hop count value initialised to ‘0’ by the initial sender  $A_i$ . This hop count value will increase with augment of hop during the flooding of this frame. Here, HopCount is limited to 7 bits with a maximum value 127 that is sufficient for the network. ‘ $x_i$ ’ and ‘ $y_i$ ’ represents  $A_i$ ’s coordinates. ‘ $x_i$ ’, as well as ‘ $y_i$ ’, is a 32-bit single precision float-point value, conforming to the IEEE standard 754 (IEEE standard 754, 1985).

Then, at Step 2,  $A_i$  broadcasts through the network a distance-per-hop frame ‘ $frame\_dph_i$ ’ which contains its average distance per hop  $dph_i$ . Its format is shown in Table 3. 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. In our case the length of a single precision float-point value should be 32 bits. However, considering ‘Data Type’ is just 1 bit, we assume that the first bit of the float-point value is used for ‘Data Type’. The other 31 bits are used for  $dph_i$ . When a node retrieves the value of  $dph_i$ , this node should automatically add one bit ‘0’ to the end of  $dph_i$ , so that a 32-bit float-point format can be obtained. Since ‘0’ is the last bit, its influence on the value of  $dph_i$  is very low.

**Table 2** Format of  $frame\_pos_i$

Data Payload					
MHR	Data Type (1 bit)	HopCount (7 bits)	$x_i$ (32 bits)	$y_i$ (32 bits)	MFR
(in total 8 bits)					

**Table 3** Format of  $frame\_dph_i$

Data Payload			
MHR	Data Type (1 bit)	$dph_i$ (31 bits)	MFR
in total 32 bits			

### 4.1.2 Proposed enhanced CSMA/CA (E-CSMA/CA) access method

The IEEE standard 802.15.4–2009 defines several channel access methods that can help to reduce collisions, for example slotted CSMA/CA and non-slotted CSMA/CA. The slotted CSMA/CA method requires a network coordinator to frequently send messages for synchronisation and network association. However non-slotted CSMA/CA does not require these transmissions. So it can be used not only with star or tree networks, but also with ad-hoc networks. Due to this simplicity and flexibility, non-slotted CSMA/CA is a popular method for low-cost sensor networks. Thus, in this paper, we mainly focus on non-slotted CSMA/CA method.

DV-hop based algorithms have not considered the problem of frame collisions which however is easy to happen during the broadcasts of frames at the first two steps of DV-hop algorithm. Even if the 802.15.4 non-slotted CSMA/CA is used as the MAC layer protocol, it cannot completely solve the collision problem. That is because usually, in point-to-point communication, the CSMA/CA scheme generates the ACK (acknowledgement) signal to ensure a final successful transmission. However, in DV-hop algorithm, since all the communications are fulfilled as broadcast, no ACK signal is sent, so it becomes non-slotted CSMA/CA without ACK, which cannot make sure transmissions succeed. So we must propose a solution to effectively reduce collisions. In the following we first analyse how the collisions take place and then introduce our solution E-CSMA/CA (non-slotted Enhanced CSMA/CA without ACK).

Collisions may happen when anchors simultaneously broadcast their position frames or distance-per-hop frames. According to the principle of CSMA/CA without ACK, each anchor needs first to wait for a short random period and then, if the channel is still free, sends the frame immediately. Here, the short random period is randomly chosen among 8 values:  $0, t_{bo}, 2 \times t_{bo}, \dots, 7 \times t_{bo}$  (IEEE Standard 802.15.4–2009, 2009), where  $t_{bo}$  is the back-off period. According to the IEEE standard 802.15.4–2009, if the data rate is 250 kbps, then  $t_{bo}$  is 320  $\mu\text{s}$  and the maximum value of this random period is  $7 \times 320 \mu\text{s} = 2.24 \text{ ms}$ . With such a short random waiting period, when frames are simultaneously broadcasted throughout the network, collisions easily occur.

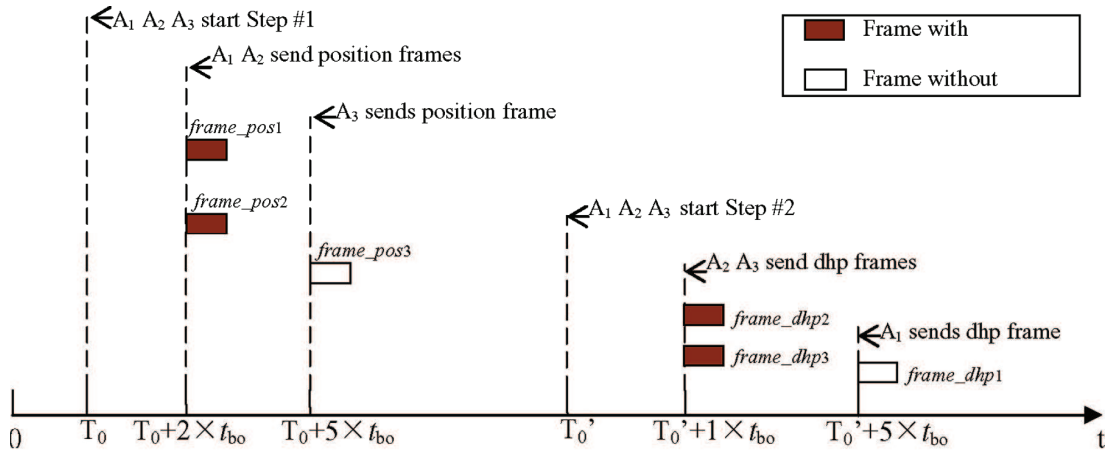
The solution we use to reduce collisions is to make the senders (nodes ready for sending frames) wait for another longer random duration before they perform CSMA/CA. So the probability of collision can be reduced. Details about this longer waiting period are described in the following.

At the beginning of Step 1 of DV-hop, each anchor  $A_i$  first waits for a random duration denoted by  $t_{wpi}$ . 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 by  $dph_i$ , it waits for a random duration denoted by  $t_{wdi}$ . Then,  $A_i$  performs CSMA/CA before sending its distance-per-hop frame  $frame\_dph_i$ .

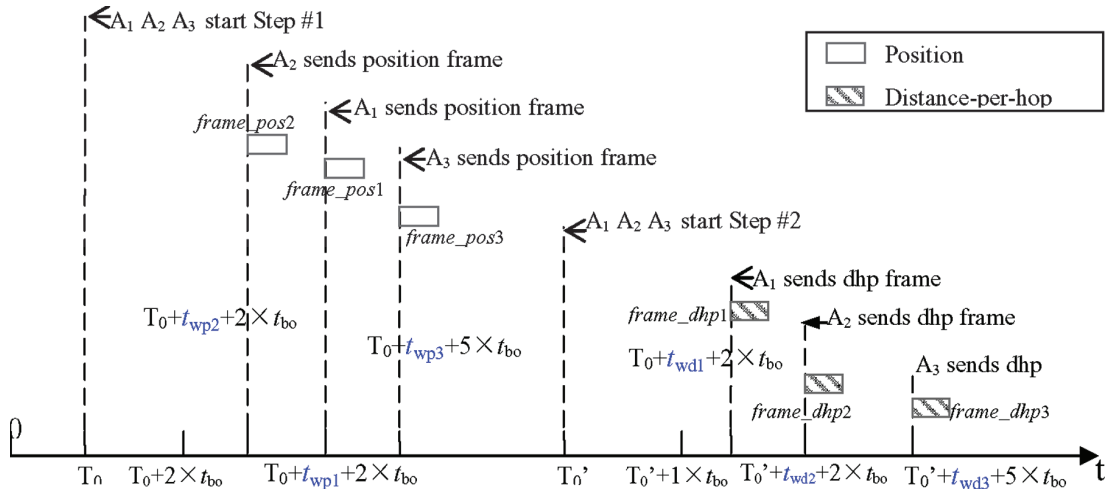
Figure 6 shows how collisions happen while Figure 7 shows how our access method E-CSMA/CA works.

In Figure 6, three anchors  $A_1$   $A_2$   $A_3$  start their first step simultaneously at the time  $T_0$ . Then, they perform the non-slotted CSMA/CA without ACK.  $A_1$  and  $A_2$  happen to choose the same period  $2 \times t_{bo}$ . 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. The same phenomenon could happen at Step 2 with  $A_2$  and  $A_3$  choosing the same waiting period  $1 \times t_{bo}$ . Figure 7 shows an example of our collision reduction method using the same scenario of Figure 6. 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. At the cost of additional waiting time, our method reduces the probability of simultaneous emissions; therefore, fewer collisions can occur.

**Figure 6** Collisions occur at Step 1 and Step 2 (see online version for colours)



**Figure 7** Example of our access method E-CSMA/CA (see online version for colours)

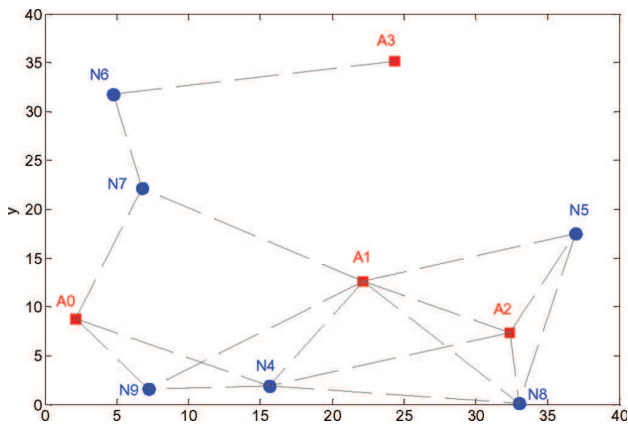


In fact, our collision reduction method E-CSMA/CA should also be applied to the relay nodes. These relay nodes, either anchors or normal nodes, help relay the position frame or distance-per-hop frame by broadcast. According to our method, every time a relay node is ready to perform CSMA/CA, this node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

Through simulations we have observed that collision probability can be effectively reduced by our E-CSMA/CA method. In the simulations, a small network is located in a  $40 \times 40 \text{ m}^2$  area. Inside the area, we uniform-randomly distribute 10 nodes. That means, the positions of the nodes are randomly assigned. All the nodes are static. The topology of reference network is shown in Figure 8.



**Figure 8** Network topology in the simulation (see online version for colours)



Among the nodes, 4 are anchors while the 6 others are normal nodes. The communication range is set to be 20 m. The physical layer of the network conforms to the IEEE standard 802.15.4–2009. In MAC layer, we will investigate and compare two methods: non-slotted CSMA/CA and our E-CSMA/CA. In the following, the simulation results will step by step display the process of DV-hop algorithm. The process of Step 1 is shown in Figure 9 which comprises two subfigures. In Figure 9(a), non-slotted CSMA/CA is used while in Figure 9(b) it is our E-CSMA/CA method.

Ideally, all nodes can receive and then relay the position frame from each anchor. However, in Figure 9(a), we can observe that, using non-slotted CSMA/CA when the anchors broadcast their positions simultaneously, nobody receives the position of  $A_2$  while the position frame of  $A_3$  is only received by  $N_6$ . Why other nodes cannot receive the position of  $A_2$  or  $A_3$ ? The reason should be frame collisions. For example, from the network topology in Figure 8, since  $A_3$  has only one neighbour node  $N_6$ , its position frame should be first received by  $N_6$ . Then,  $N_6$  relays the position frame of  $A_3$ . The neighbour of  $N_6$ , that is  $N_7$ , is supposed to receive this relayed frame. But at the same time,  $N_7$  is also relaying the position frame of  $A_0$ . Thus, collision happens on these two relayed frames.

On contrary, good results are shown in Figure 9(b) for our E-CSMA/CA method. We can notice that the position frame from each anchor has been successfully received and relayed by all nodes which shows that no collisions exist. This is contributed by the random waiting time added to non-slotted CSMA/CA.

A similar phenomenon can also be observed for Step 2. Considering the similarity of simulation results for Step 1 and Step 2, here we do not give the result figure of Step 2. From the simulation results, we can conclude that our E-CSMA/CA method is an efficient solution to reduce the frame collisions in DV-hop localisation algorithm.

**Figure 9** Step 1 of DV-hop algorithm simulated by WSNet: (a) non-slotted CSMA/CA and (b) our E-CSMA/CA

<pre> position of 0 is 2.153714, 8.732726 position of 1 is 22.120116, 12.637730 position of 2 is 32.373271, 7.341937 position of 3 is 24.372541, 35.170193 position of 4 is 15.624166, 1.871338 position of 5 is 36.933841, 17.514913 position of 6 is 4.739949, 31.790328 position of 7 is 6.761391, 22.171018 position of 8 is 33.043891, 0.103658 position of 9 is 7.244737, 1.577186 [Anchor] Node 0 broadcast position packet, seq=1 Now the time is 1000000000 [Anchor] Node 1 broadcast position packet, seq=1 Now the time is 1000000000 [Anchor] Node 2 broadcast position packet, seq=1 Now the time is 1000000000 [Anchor] Node 3 broadcast position packet, seq=1 Now the time is 1000000000 [0/1] Node 6 relays the position of 3 [0/1] Node 9 relays the position of 0 [0/1] Node 7 relays the position of 0 [0/1] Node 4 relays the position of 0 [0/1] Node 5 relays the position of 1 [0/1] Node 8 relays the position of 1 [0/1] Node 2 relays the position of 1 [0/1] Node 4 relays the position of 1 [0/1] Node 9 relays the position of 1 [0/1] Node 0 relays the position of 1 [0/1] Node 7 relays the position of 1 [0/1] Node 6 relays the position of 1 [0/1] Node 3 relays the position of 1 </pre>	<pre> position of 0 is 2.153714, 8.732726 position of 1 is 22.120116, 12.637730 position of 2 is 32.373271, 7.341937 position of 3 is 24.372541, 35.170193 position of 4 is 15.624166, 1.871338 position of 5 is 36.933841, 17.514913 position of 6 is 4.739949, 31.790328 position of 7 is 6.761391, 22.171018 position of 8 is 33.043891, 0.103658 position of 9 is 7.244737, 1.577186 [Anchor] Node 2 broadcast position packet, seq=1 Now the time is 1012500000 [0/1] Node 8 relays the position of 2 [0/1] Node 5 relays the position of 2 [0/1] Node 1 relays the position of 2 [0/1] Node 4 relays the position of 2 [0/1] Node 7 relays the position of 2 [0/1] Node 9 relays the position of 2 [0/1] Node 0 relays the position of 2 [0/1] Node 6 relays the position of 2 [0/1] Node 3 relays the position of 2 [0/1] Node 0 relays the position of 2 [Anchor] Node 3 broadcast position packet, seq=1 Now the time is 1018750000 [0/1] Node 6 relays the position of 3 [0/1] Node 7 relays the position of 3 [0/1] Node 0 relays the position of 3 [0/1] Node 1 relays the position of 3 [0/1] Node 2 relays the position of 3 [0/1] Node 4 relays the position of 3 [0/1] Node 5 relays the position of 3 [0/1] Node 8 relays the position of 3 [0/1] Node 9 relays the position of 3 [Anchor] Node 0 broadcast position packet, seq=1 Now the time is 1068750000 [0/1] Node 9 relays the position of 0 [0/1] Node 7 relays the position of 0 [0/1] Node 4 relays the position of 0 [0/1] Node 6 relays the position of 0 [0/1] Node 1 relays the position of 0 [0/1] Node 8 relays the position of 0 [0/1] Node 2 relays the position of 0 [0/1] Node 3 relays the position of 0 [0/1] Node 5 relays the position of 0 [Anchor] Node 1 broadcast position packet, seq=1 Now the time is 1093750000 [0/1] Node 2 relays the position of 1 [0/1] Node 4 relays the position of 1 [0/1] Node 5 relays the position of 1 [0/1] Node 8 relays the position of 1 [0/1] Node 7 relays the position of 1 [0/1] Node 9 relays the position of 1 [0/1] Node 6 relays the position of 1 [0/1] Node 0 relays the position of 1 [0/1] Node 3 relays the position of 1 </pre>
---	---

(a)

(b)



### 4.1.3 Parameters for the end of each step

In DV-hop algorithm, Step 1 ends as soon as every node in the network has received all anchors' position frames while Step 2 ends on condition that all anchors' distance-per-hop frames have been received. These 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 of two issues. In the following, we analyse the issues and propose several parameters to control the end of the first two steps of DV-hop.

As for the first issue, it is unnecessary for nodes to receive all anchors' positions, especially when the number of anchors is very large, because mobile normal nodes need to calculate their positions as quickly as possible. Therefore, each node needs to set a maximum number of anchors whose information they take into account. This maximum number can be denoted by 'num\_wait\_pos'. As long as a normal node has received num\_wait\_pos anchors' positions, it can stop relaying position frames and end Step 1. As for anchors, when an anchor has received num\_wait\_pos-1 anchors' positions, it can end Step 1. (Here, it's 'num\_wait\_pos-1', 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 issue occurs when collisions happen or the total number of anchors is less than 'num\_wait\_pos' or 'num\_wait\_dph'. When collisions exist at 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 may never receive as many as 'num\_wait\_pos' anchors positions, 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 this issue.

In order to end Step 1, we need to set a timer for each node  $N_i$  at the time instant  $T_i^0 + t_{s1}$ . Here, since our protocol is periodical,  $T_i^0$  is  $N_i$ 's beginning time of localisation period.  $t_{s1}$  is the maximum duration of Step 1, which is configured and shared by all nodes. Before the expiration of  $T_i^0 + t_{s1}$ , those anchors who have already received as many as 'num\_wait\_pos-1' anchors' positions must immediately end Step 1. When  $T_i^0 + t_{s1}$  arrives, the anchors who have not yet received the specified amount of data need to immediately end Step 1.

In order to end Step 2 (Start Step 3), we need to set a timer at the time instant  $T_i^0 + t_{s1} + t_{s2}$ . Here,  $t_{s2}$  is the maximum duration of Step 2, shared by all normal nodes. In fact, Step 3 of DV-hop is designed for normal nodes to calculate positions. Hence, the timer for starting Step 3 is specific to normal nodes. Before  $T_i^0 + t_{s1} + t_{s2}$ , 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 start Step 3. When  $T_i^0 + t_{s1} + t_{s2}$  arrives, other

normal nodes, who 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 2, while Step 3 only includes position calculation. Since broadcasts usually take much more time than calculation, the total duration of Step 1 and Step 2 is very close to the entire period of localisation  $t_p$ . That is,  $t_{s1} + t_{s2} \approx t_p$ . Besides, since Step 1 and Step 2 both broadcast frames, their duration should be similar. We can consider  $t_{s1} \approx t_{s2}$ . For example,  $t_{s1}$  could be set as  $t_p/2$ , while  $t_{s2}$  could be set as  $t_p \times (3/8)$ . Then, the time left is devoted to Step 3, that is:  $t_p - t_{s1} - t_{s2} = t_p/8$ .

### 4.1.4 Procedure of ALP-3 protocol

The execution of ALP-3 protocol is shown in the following figure. Figure 10(a) shows the procedure followed by each anchor  $A_i$ . The period begins at  $T_i^0$ . Then, according to our collision avoidance method E-CSMA/CA,  $A_i$  first waits for a random duration  $t_{wpi}$ , and then broadcasts through the network its position frame which has been defined in Section 4.1.1. 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  $A_i$  couldn't receive as many as 'num\_wait\_pos-1' anchors' position frames until the time instant  $T_i^0 + t_{s1}$ , it will still end Step 1 at  $T_i^0 + t_{s1}$ .

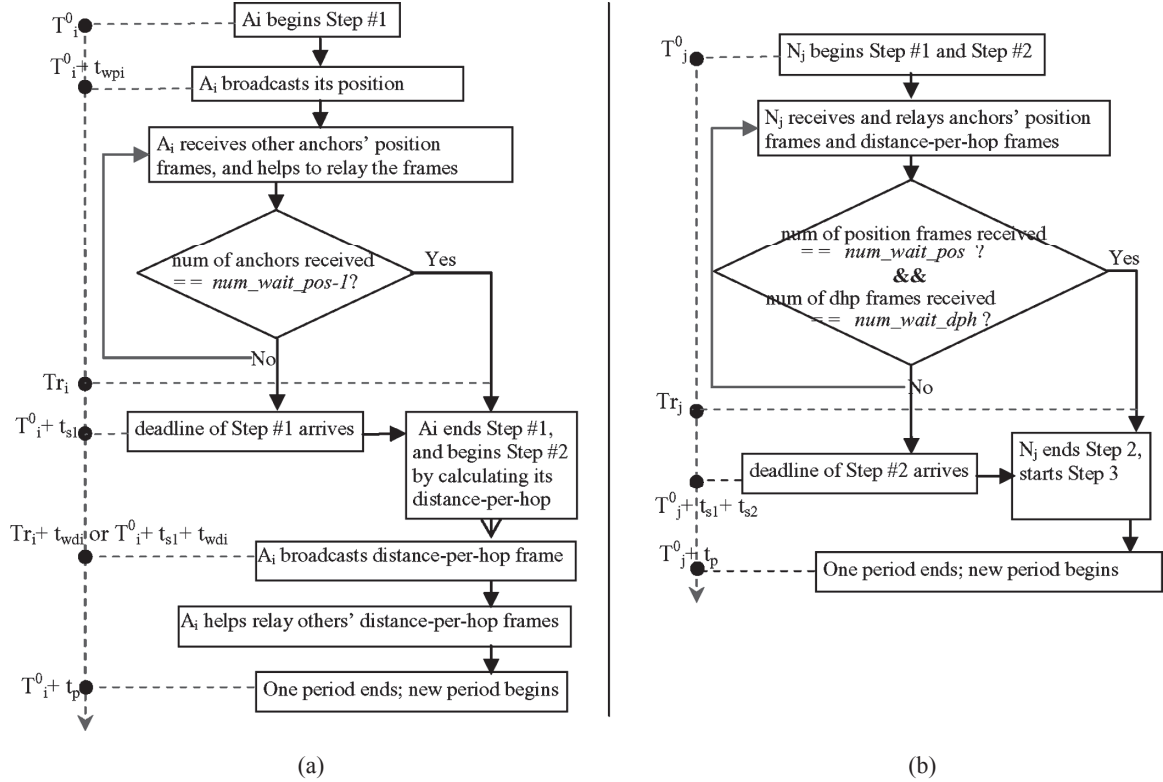
After Step 1,  $A_i$  begins Step 2 by calculating its distance-per-hop. Then,  $A_i$  waits for a random duration  $t_{wdi}$  and broadcasts through the network its distance-per-hop frame. Meanwhile,  $A_i$  also helps relay the distance-per-hop frames of other anchors. Here, the end of  $A_i$ 's Step 2 is also the end of its participation in the localisation period since the third step is designed for normal nodes.

Figure 10(b) shows the procedure for each normal node  $N_j$ . It begins its period at  $T_j^0$ . During the first two steps,  $N_j$  receives and relays anchors' frames. When  $N_j$  has received as many as num\_wait\_pos anchors' positions and as many as num\_wait\_dph anchors' distance-per-hop, it will immediately end the first two steps and start the third step. 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  $T_j^0 + t_{s1} + t_{s2}$ , it will end Step 2 anyway.

## 4.2 ALP+3 protocol

In the previous section, we focused on ALP-3 protocol which is very useful to localise A-3 normal nodes (with less than 3 neighbour anchors). However, we should note a disadvantage of DV-hop based algorithms implemented by ALP-3 protocol: the high network overhead when there are many anchors. In fact, when the ratio of anchors is high, normal nodes can have at least 3 neighbour anchors. In this case, instead of using DV-hop based algorithms, we recommend the A+3 algorithms, such as Centroid, CPE and Mid-perpendicular. Therefore, we propose ALP+3 protocol which can implement those A+3 algorithms.

**Figure 10** Procedure of ALP-3 protocol: (a) procedure for each anchor  $A_i$  and (b) procedure for each normal node  $N_j$



ALP+3 protocol is supposed to function in case of high ratio of anchors. In this case, we do not suggest that all anchors periodically broadcast their positions, because this leads to much more network overhead. It is better for the anchors to broadcast positions only when normal nodes ask them to do it.

Thus, the basic principle of ALP+3 protocol is as follows, including 3 steps. First, when a normal node  $N_x$  needs to calculate its position, it broadcasts a localisation request to its neighbourhood. Second, if a neighbour anchor of  $N_x$  detects this request, this anchor sends its position to  $N_x$ . Finally, if the node has collected at least 3 neighbour anchors' positions during a certain period,  $N_x$  can calculate its position by Centroid, CPE or Mid-perpendicular.

In the following, the protocol is explained in details. At Step 1, the normal node  $N_x$  broadcasts to its neighbourhood a localisation request frame, denoted as *frame\_req*. When broadcasting *frame\_req*, our E-CSMA/CA method should be used to reduce collisions because several normal nodes may be simultaneously ready to send their request frames. Here, the additional random waiting time in E-CSMA/CA method is denoted as  $t_{wlr}$ . Considering this request frame is broadcasted, no ACK signal is required.

*frame\_req* conforms to the command frame format in IEEE standard 802.15.4-2009. Shown in Table 4, *frame\_req* has 3 parts: MHR, MAC payload and MFR. Since *frame\_req* is broadcasted by  $N_x$ , the source address must be the address of  $N_x$ . The MAC payload only has an 8-bit field 'Command Type'. Its value is set to be 04.

According to the IEEE standard, this value means *frame\_req* is used to request data (positions of anchors).

At Step 2, the anchors who have received  $N_x$ 's *frame\_req* should send their positions to  $N_x$ . These anchors are  $N_x$ 's neighbour anchors. The number of neighbour anchors is at least 3, maybe even bigger such as 7 or 8, depending on the specific scenario. If all these neighbour anchors demand each normal node  $N_x$  to send back ACK signals, then the network overhead will increase a lot. So, at this step, no ACK signal is demanded.

The position frame sent by the anchor  $A_i$  to  $N_x$  is denoted as *frame\_pos<sub>i,Nx</sub>*. It conforms to the data frame format in IEEE standard 802.15.4-2009. Shown in Table 5, the source address is the 16-bit short MAC address of  $A_i$ , while the destination address is that of  $N_x$ . *Data payload* comprises the coordinate of  $A_i$ .

For transmitting *frame\_pos<sub>i,Nx</sub>*, our E-CSMA/CA method is recommended to reduce frame collisions. Because  $N_x$  may have quite a few neighbour anchors (for example, as many as 6), all these anchors receive  $N_x$ 's localisation request at the same time. If they perform non-slotted CSMA/CA before sending position frames, since the random waiting time is very small, two anchors may simultaneously send out frames, resulting in a collision. Thus, we can use E-CSMA/CA method which has an additional random waiting time, denoted by  $t_{wpi}$ .

Before Step 3, assume that  $N_x$  has received  $m$  anchors' position frames during a period  $t_{recv}$ . Here,  $t_{recv}$  is also the duration of Step 1 and 2 because  $N_x$  is always collecting anchors' positions after sending the request. At Step 3,

$N_x$  calculates its position. An example of the procedure for ALP+3 protocol is shown in Figure 11.

**Table 4** Format of *frame\_req*

MHR		MAC Payload		MFR
Frame Control (16 bits)	Sequence Number (8 bits)	Destination Address (0xffff, 16 bits)	Source Address (16 bits)	Command Type (0x04, 8 bits)
				FCS (16 bits)

**Table 5** Format of *frame\_pos<sub>i,Nx</sub>*

MHR		Data Payload			MFR
Frame Control (16 bits)	Sequence Number (8 bits)	Destination Address (16 bits)	Source Address (16 bits)	$x_i$ (32 bits)	$y_i$ (32 bits)
				FCS (16 bits)	

**Figure 11** Example of procedure of ALP+3 protocol: (a) network topology and (b) procedure of ALP+3 protocol (see online version for colours)

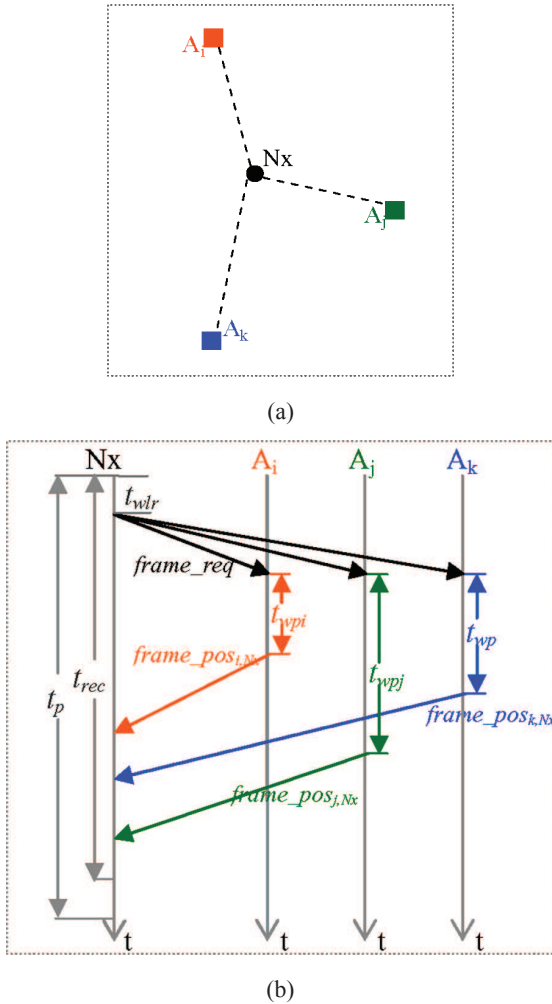


Figure 11(a) gives an example of network topology. The normal node  $N_x$  has three neighbour anchors,  $A_i$ ,  $A_j$ , and  $A_k$ . Shown in Figure 11(b),  $N_x$  collects the position frames from the anchors during the period  $t_{rec}$ . The entire periodical duration of the three steps is denoted as  $t_p$ .

### 4.3 Adaptive Range-free Localisation Protocol (ALP): Combination of ALP-3 and ALP+3

ALP-3 protocol and ALP+3 protocol both have advantages and disadvantages. ALP+3 protocol is simple but it requires normal node has at least 3 neighbour anchors. ALP-3 protocol can serve the case of low ratio of anchors but it has considerable network overhead. In order to take advantage of the two protocols, the combination of the above two protocols is regarded as our ALP.

In this adaptive protocol, the choice between ALP-3 protocol and ALP+3 protocol can be decided by each normal node or by network administrator. If it is decided by each normal node, the corresponding protocol is chosen according to the number of neighbour anchors. That means, a normal node will choose ALP+3 protocol when it has at least 3 neighbour anchors; otherwise, it will choose ALP-3 protocol. But this method has a practical problem considering the different communication manners between the two protocols. In ALP+3 protocol, anchors are in passive mode: they wait for requests from normal nodes, if receive a request, they broadcast position frames only to neighbours. However, in ALP-3 protocol, anchors are in active mode: they do not need to listen to request, and they should broadcast their positions related information throughout the network. Suppose that in a network, anchors stay in passive mode by default, and most normal nodes use ALP+3 protocol, while only one normal node needs to use ALP-3 protocol. The problem is how the particular one normal node informs all anchors to change from passive mode into active mode. The solution can be: this normal node broadcasts a special request frame throughout the network; when receiving this special frame, anchors need to begin the process of ALP-3 protocol. We can notice that the broadcast of this special frame increases the network overhead.

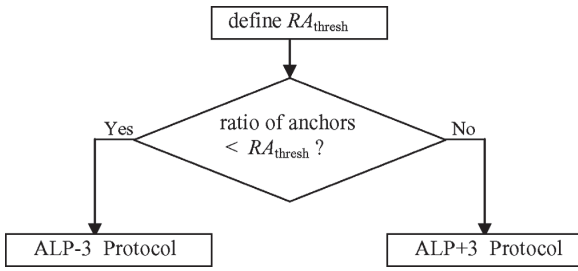
Thus, we suggest that the choice between the two protocols is decided by the network administrator. So, when the choice is made, the network will use one protocol, ALP+3 or ALP-3. In fact, in case of high ratio of anchors, considering the network overhead, ALP+3 protocol is a better solution than ALP-3 protocol. Thus, in order to choose the proper protocol, we need to set a threshold for the ratio of anchors, denoted by  $RA_{thresh}$ .

Suppose that in the network the number of anchors is stable and the administrator has known the ratio of anchors. Then, if the ratio of anchors is lower than  $RA_{thresh}$ , the administrator chooses ALP-3 protocol because normal nodes are mostly A-3 nodes. But when the ratio of anchors is higher than  $RA_{thresh}$ , in order to avoid a large number of network traffic, ALP+3 protocol should be used.

The value of  $RA_{thresh}$  is decided by the administrator according to the maximum traffic that the network can accept. A lower  $RA_{thresh}$  indicates the network can only accept lower network overhead. But the value of  $RA_{thresh}$  cannot be too low; otherwise, many A-3 normal nodes are unable to be localised. After the administrator sets the value for  $RA_{thresh}$ , comparing the ratio of anchors with

$RA_{\text{thresh}}$ , the corresponding protocol can be chosen. The idea is summarised in Figure 12.

**Figure 12** Basic principle of Adaptive Range-free Localisation Protocol (ALP)



## 5 Simulation and evaluation of ALP protocol

### 5.1 Simulation scenarios

The simulator we use is WSNNet which is an event-driven simulator designed by three researchers from INRIA (<http://wsnet.gforge.inria.fr/tutorial.html>). Compared to others like NS-2 and OPNET, WSNNet not only facilitates the development of new models, but also supplies sufficient modules at each layer (Hamida et al., 2008). Using WSNNet, we have implemented our ALP protocol as a model in C language. The simulation scenarios take place within a  $100 \times 100 \text{ m}^2$  area. Inside the area, 100 nodes including anchors and normal nodes are randomly distributed. Scenario parameters and their values are listed in Table 6.

As shown in the table,  $RA_{\text{thresh}}$  is set to be 40%. That means, when the ratio of anchors is less than 40%, ALP-3 protocol is used. When the ratio of anchors is no less than 40%, ALP+3 protocol turns to work.

**Table 6** Scenario parameters

Radio range of nodes	15 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
Ratio of anchors	5, 10, 15, 17, 19, 20, 25, 30, 40, 50, 60, 70, 80, 90/100
Total simulation time	1000 localisation periods
Localisation period $t_p$	6 s for ALP-3 protocol; 3 s for ALP+3 protocol
$A_i$ 's random waiting time: $t_{wpi}$ and $t_{wdr}$ (ALP-3 protocol)	and randomly selected between 0 s and 0.5 s
Maximum duration of Step 1: $t_{s1}$ (ALP-3 protocol)	$1/2 * t_p = 3 \text{ s}$
Maximum duration of Step 2: $t_{s2}$ (ALP-3 protocol)	$3/8 * t_p = 2.25 \text{ s}$

**Table 6** Scenario parameters (continued)

Maximum number: $num\_wait\_pos$ (ALP-3 protocol)	30
Maximum number: $num\_wait\_dph$ (ALP-3 protocol)	20
$RA_{\text{thresh}}$ (ALP+3 protocol)	40%
Normal node's waiting time: $t_{wlr}$ (ALP+3 protocol)	Randomly selected between 0 and 100 ms
$A_i$ 's waiting time: $t_{wpi}$ (ALP+3 protocol)	Randomly selected between 0 and 100 ms
$t_{recv}$ (duration of Step 1 and 2) (ALP+3 protocol)	2.5 s
Network synchronised or not	Synchronised (all nodes start at the same time) Unsynchronised
Nodes mobility	Static (for all nodes) Mobile (anchors static, but normal nodes move at a speed of 0.5 m/s in billiard mode)

### 5.2 Simulation results and related analysis

The simulation results of three scenarios (static, synchronised mobile, and unsynchronised mobile) are presented in Figure 13 and 14. The data is collected at each ratio of anchors. Figure 13 shows the average location error per node per localisation period, expressed as a percentage of the radio range. Figure 14 presents the average number of transmitted frames per localisation period. In total, six algorithms are compared. Three of them are DV-hop based algorithms, including DV-hop, Checkout DV-hop and Selective 3-Anchor DV-hop which are evaluated based on ALP-3 protocol. The others are A+3 algorithms, such as Centroid, CPE and Mid-perpendicular which function with ALP+3 protocol.

The average location errors of the algorithms in static scenario are shown in Figure 13(a). We can notice that all the A+3 algorithms have much better accuracy than DV-hop and Checkout DV-hop while the ratio of anchors is no less than 40%. We can also notice that, among the three A+3 algorithms, our Mid-perpendicular algorithm has the best accuracy, although the improvement is only about 15% on average.

Figure 13(b) shows the average location error of algorithms in synchronised mobile scenario. Comparing Figure 13(a) and Figure 13(b), we can note that, when nodes change from static to mobile, the accuracy of A+3 algorithms decreases a little. However, the accuracy of DV-hop based algorithms has a larger decrease.

This small decrease of A+3 algorithms is contributed by the small localisation period of ALP+3 protocol which is only 3 s. Considering the speed of nodes movement is 0.5 m/s, normal nodes move only  $0.5 \times 3 = 1.5 \text{ m}$ . However, in ALP-3 protocol for DV-hop based algorithms, frames need to have enough time to broadcast through the network, thus the localisation period of ALP-3 protocol is configured as long as 6 s. During 6 seconds, normal nodes can move as



far as  $0.5 \times 6 = 3$  m. Therefore, node movement has a more important influence on the accuracy of ALP-3 protocol than that of ALP+3 protocol.

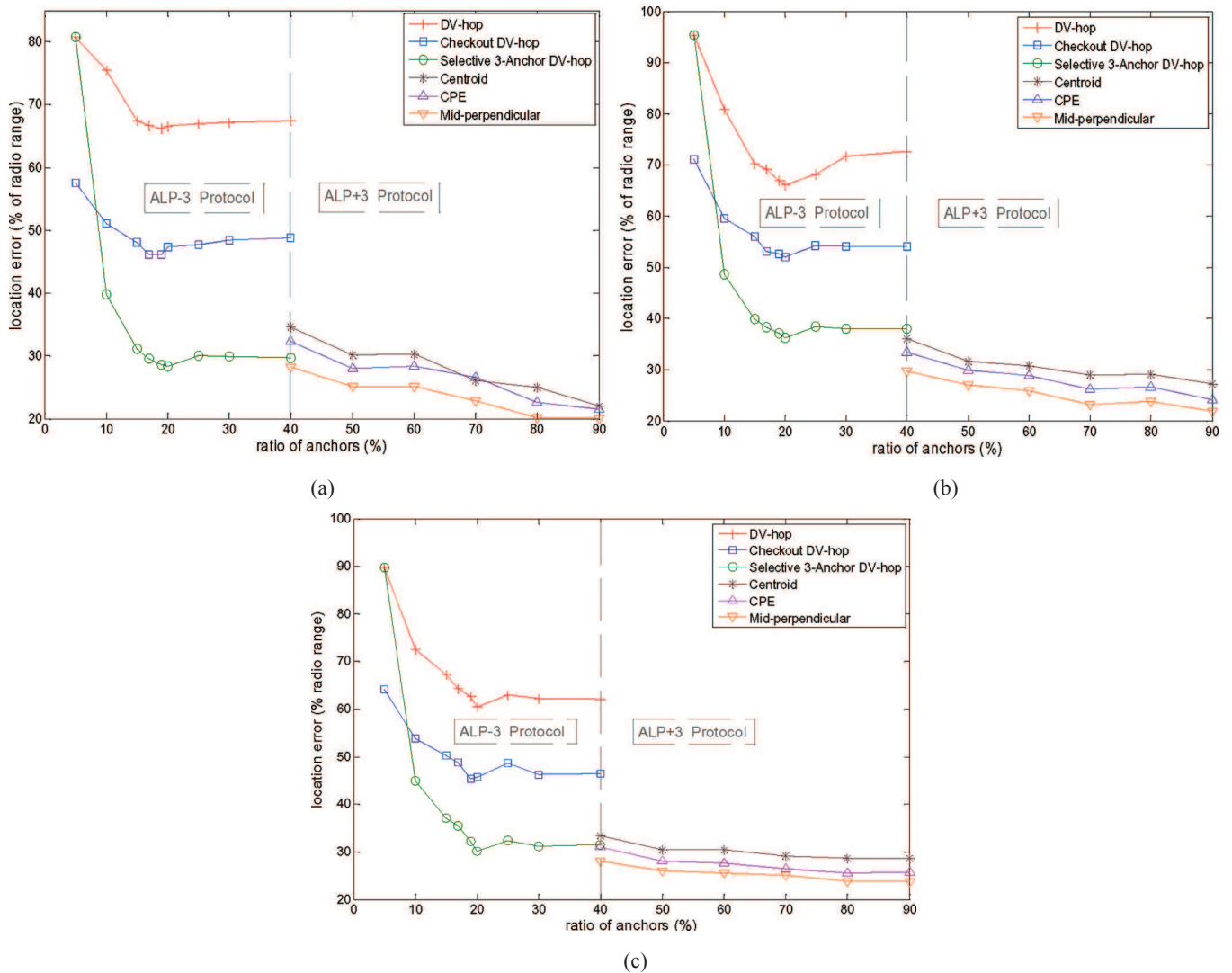
Figure 13(c) presents the average location error in unsynchronised mobile scenario. Comparing Figure 13(b) with Figure 13(c), we can see the different improvement of ALP-3 protocol and ALP+3 protocol. Based on ALP-3 protocol, DV-hop based algorithms have an obvious accuracy improvement in unsynchronised mobile scenario compared with the accuracy in synchronised mobile scenario. However, A+3 algorithms only have a slight improvement in unsynchronised mobile scenario. The reason is as follows. ALP-3 protocol has a large number of broadcast traffic, while ALP+3 protocol has much lower traffic. So, the possibility of frame collisions in ALP+3 protocol is much lower than that in ALP-3 protocol. Thus, as for ALP+3 protocol, the few collisions can be effectively reduced by our E-CSMA/CA method resulting in no significant change between synchronised scenario and unsynchronised scenario. However, considering the massive

traffic in ALP-3 protocol, our E-CSMA/CA may be not enough to avoid collisions. In unsynchronised scenario, nodes (especially the anchors) begin their broadcast periods at different time, so it helps to further reduce frame collisions. Therefore, as for ALP-3 protocol, the accuracy in unsynchronised scenario is obviously better than in synchronised scenario.

In the following, the network overhead is investigated. All DV-hop based algorithms use the same protocol: our ALP-3 protocol. Thus they have the same network overhead. The A+3 algorithms using our ALP+3 protocol also have the same network overhead. Therefore, we need to compare the network overhead of ALP-3 protocol with that of ALP+3 protocol.

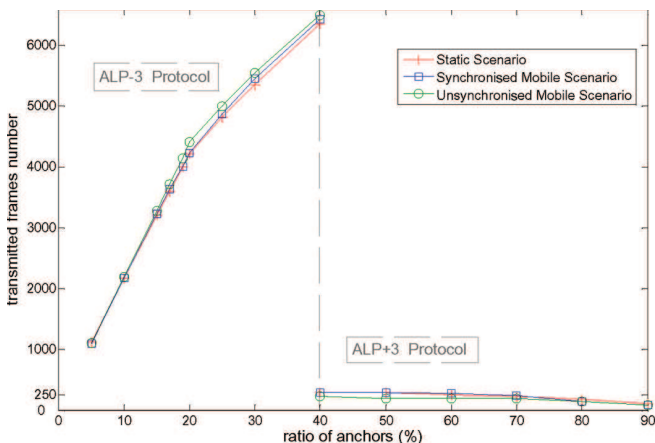
The network overhead of the protocols is quantised by the average number of transmitted frames by all 100 nodes per localisation period. Simulation results are shown in Figure 12. From this figure, we can notice that the network overhead of ALP-3 protocol is much higher than that of ALP+3 protocol.

**Figure 13** Location error in different scenarios with ALP protocol: (a) location error in static scenario; (b) location error in synchronised mobile scenario and (c) location error in unsynchronised mobile scenario (see online version for colours)





**Figure 14** Number transmitted frames for ALP protocol (see online version for colours)



Now, we estimate the approximate value of number of transmitted frames for ALP-3 and ALP+3 protocols. In ALP-3 protocol, the network traffics exist only at the first two steps. At Step 1, each anchor  $A_i$  broadcasts its position frame  $frame_{pos_i}$  throughout the network. In order to make all nodes be aware of  $frame_{pos_i}$ , every node in the network needs to relay this frame once. Thus, if the total number of nodes is  $num$ , the number of anchors is  $num \times \text{'ratio of anchors'}$ , so the number of transmitted frames at Step 1 of ALP-3 protocol is at least  $num \times (num \times \text{'ratio of anchors'}) = num^2 \times \text{'ratio of anchors'}$ . The same result can be obtained for Step 2. Thus, as for all the three scenarios, the approximate value of number of transmitted frames for ALP-3 protocol is  $2 \times num^2 \times \text{'ratio of anchors'}$ . To verify this, for example in Figure 12, when the ratio of anchors is 5%, the number of transmitted frames is about 1000. This value is just  $2 \times 100^2 \times 5\%$  considering the total number of nodes is 100.

In ALP+3 protocol, the network overhead also exists only at the first two steps. At Step 1, each normal node broadcasts its localisation request just to its neighbour nodes. Thus, the number of transmitted frames at Step 1 is exactly  $num \times (1 - \text{'ratio of anchors'})$  which is also the number of normal nodes. At Step 2, the neighbour anchors of each normal node respond the request by sending back their positions. Thus, if on average there are  $m$  neighbour anchors for each normal node, then the approximate value of number of transmitted frames for ALP+3 protocol is  $num \times (1 - \text{'ratio of anchors'}) \times m$ . To verify this result, for example in Figure 12, when the ratio of anchors is 40%, the number of transmitted frames is about 250 which is nearly  $100 \times (1 - 40\%) \times 4 = 240$ , where  $m$  is assumed to be 4. Considering  $m$  is usually a small value (at least 3) depending on network topology and the ratio of anchors, the number of transmitted frames of ALP+3 protocol is much less than that of ALP-3 protocol.

### 5.3 Brief summary of evaluations on ALP protocol

From the above analysis on simulation results, we have noted that: ALP-3 protocol has much higher overhead than ALP+3 protocol. The overhead can be quantised by the

metric 'number of transmitted frames'. The approximate value of number of transmitted frames for ALP-3 protocol is  $2 \times num^2 \times \text{'ratio of anchors'}$  while that for ALP+3 protocol is  $num \times (1 - \text{'ratio of anchors'}) \times m$ , where  $m$  is the average number of neighbour anchors for each normal node and  $num$  is the total number of nodes. So, given the ratio of anchors, the network administrator can estimate the network overhead for both protocols.

Thus, the maximum acceptable network overhead has its corresponding maximum ratio of anchors which is defined as the threshold of ratio of anchors ' $RA_{thres}$ '. When the ratio of anchors is lower than  $RA_{thres}$ , ALP-3 protocol needs to be used; but when the ratio of anchors is higher than  $RA_{thres}$ , in order to avoid a large number of network traffic, ALP+3 protocol should be used. This is also the basic principle of our ALP protocol.

Based on the corresponding protocols, the accuracy of the related algorithms has been evaluated in network scenarios. Although the improvement by our Mid-perpendicular algorithm and Checkout DV-hop algorithm is not so significant, our Selective 3-Anchor DV-hop algorithm has an accuracy about 35% better than our Checkout DV-hop algorithm and about 50% better than DV-hop algorithm.

Node mobility has a bigger influence on the accuracy of DV-hop based algorithms than that of A+3 algorithms. The reason is: while ALP+3 protocol has the broadcast only to neighbour nodes, ALP-3 protocol need more time to broadcast information throughout the network. Thus, the localisation period of ALP-3 protocol is longer than that of ALP+3 protocol. Therefore, moving at the same speed in ALP-3 protocol, normal nodes move away further during one period than in ALP+3 protocol.

Synchronisation also has an important influence on ALP-3 protocol. Compared with synchronised mobile scenario, ALP-3 protocol has an obvious accuracy improvement in unsynchronised mobile scenario. However, ALP+3 protocol only has a slight improvement in unsynchronised scenario. This reveals that our E-CSMA/CA method is already qualified for ALP+3 protocol but not sufficient for ALP-3 protocol. After all, synchronisation is not a necessary condition for both protocols.

As for calculation time, for both protocols, since the position calculation is restricted to Step 3, the calculation time does not exceed the duration of Step 3. In our simulation, the duration of Step 3 for ALP-3 protocol is set to be 0.75 s while that for ALP+3 protocol is 0.5 s. Therefore, all the related algorithms spend a little time calculating the position.

Table 7 gives a brief comparison on accuracy and overhead of the protocols.

## 6 Conclusions and perspectives

In the context of low-cost wireless ad-hoc sensor networks, the range-free localisation scheme is not only more cost-effective than the range-base scheme, but also more robust. When we implement typical range-free algorithms in

network scenarios, some problems such as frame collisions, node mobility and synchronisation, should be taken into consideration. Thus, based on IEEE standard 802.15.4–2009, we propose in this paper an ALP which is a combination of ALP-3 protocol and ALP+3 protocol.

**Table 7** Brief comparison on the protocols and algorithms

<i>Adaptive range-free localisation protocol (ALP)</i>	
<i>ALP+3 protocol</i>	<i>ALP-3 protocol</i>
Accuracy Mid-perpendicular > CPE > Centroid > Scenarios: unsynchronised mobile > static > synchronised mobile	Selective 3-Anchor DV-hop > Checkout DV-hop > DV-hop
Network overhead $num \times (1 - \text{"ratio of anchors"}) \times m$	$2 \times num^2 \times \text{"ratio of anchors"}$

In our ALP-3 protocol, we design new data payload formats and 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 based algorithms. ALP-3 protocol can be used to implement the DV-hop based algorithms, including the original DV-hop algorithm, our Checkout DV-hop algorithm and our Selective 3-Anchor DV-hop algorithm.

In our ALP+3 protocol, normal nodes broadcast their localisation request to neighbour nodes and then their neighbour anchors respond by sending back anchors' positions. In the protocol, we design new data payload formats and our E-CSMA/CA method is also used to reduce frame collisions. ALP+3 protocol can be used to implement the A+3 algorithms including Centroid, CPE and Mid-perpendicular.

Our protocols are implemented using the simulator WSNnet in the IEEE 802.15.4 wireless network. Comparative network simulation results are presented and analysed in terms of localisation accuracy, overhead, node mobility and node synchronisation. Results show that, globally, our new algorithms have better accuracy than the existing typical range-free algorithms. We can also note that, in term of overhead, DV-hop based algorithms have much higher network overhead than A+3 algorithms like Centroid and CPE because DV-hop based algorithms require to broadcast throughout the network. In terms of mobility, node mobility can have a more important influence on the accuracy of DV-hop based algorithms than that of A+3 algorithms. Finally, it should be noted that node synchronisation is not necessary for our protocols in ad hoc topology.

In the future, we will make our ALP protocol work automatically without the configuration of network administrator. In this paper, the network administrator chooses ALP-3 or ALP+3 protocol for all nodes. This mechanism has low network overhead. But it is not flexible because the network can change with node movement, node failure and new association. Thus, we will investigate a new

mechanism to let each node choose its proper protocol without much increase of network overhead.

## References

- Dai, H., Chen, A.G., Gu, X.F. and He, L. (2011) 'Localisation algorithm for large-scale and low-density wireless sensor networks', *Electronics Letters*, Vol. 47, No. 15, 21 July, pp.881–883.
- Doherty, L., Pister, K. and Ghaoui, L. (2001) 'Convex position estimation in wireless sensor networks', *Proceedings of IEEE INFOCOM '01*, April, Alaska, USA, Vol. 3, pp.1655–1663.
- Gu, Y., Lo, A. and Niemegeers, I. (2009) 'A survey of indoor positioning systems for wireless personal networks', *IEEE Communications Surveys & Tutorials*, Vol. 11, No. 1, First Quarter, pp.13–32.
- Guerrero, E., Xiong, H.G. and Gao, Q. (2009) 'A distributed range-free localization algorithm for wireless sensor networks based on a mobile robot', *International Conference on Advanced Technologies for Communications*, October, Hai Phong, Vietnam, pp.93–98.
- Gui, L., Val, T. and Wei, A. (2011) 'Improving localization accuracy using selective 3-anchor DV-hop algorithm', *IEEE Vehicular Technology Conference (VTC 2011-fall)*, September, San Francisco, USA, pp.1–5.
- Gui, L., Wei, A., and Val, T. (2010) 'A two-level range-free localization algorithm for wireless sensor networks', *IEEE Conference on Wireless Communications Networking and Mobile Computing*, Chengdu, China, pp.1–4.
- Hamida, E., Chelius, G., Gorce, J. (2008) 'On the complexity of an accurate and precise performance evaluation of wireless networks using simulations', *Proceedings of the 11th ACM-IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2008)*, October, Vancouver, Canada, pp.395–402.
- Hou, S., Zhou, X. and Liu, X. (2010) 'A novel DV-hop localization algorithm for asymmetry distributed wireless sensor networks', *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, Vol. 4, pp.243–248.
- IEEE standard 754 (1985) *IEEE Standard for Binary Floating-Point Arithmetic*, IEEE Computer Society.
- IEEE Standard 802.15.4–2009 (2009) *IEEE Standard for Wireless Personal Area Networks (PANs)*, IEEE Standards Association.
- Kovavisaruch, L. and Ho, K.C. (2005) 'Alternate source and receiver location estimation using TDOA with receiver position uncertainties', *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, March, Pennsylvania, USA, pp.iv/1065–iv/1068.
- Kumar, P., Reddy, L. and Varma, S. (2009) 'Distance measurement and error estimation scheme for RSSI based localization in wireless sensor networks', *Fifth IEEE Conference on Wireless Communication and Sensor Networks (WCSN)*, December, Allahabad, India, pp.1–4.
- Lee, B. and Chung, W.Y. (2011) 'Multitarget three-dimensional indoor navigation on a PDA in a wireless sensor network', *IEEE Sensors Journal*, Vol. 11, No. 3, March, pp.799–807.
- Lee, J., Chung, W., Kim, E. and Hong, I. (2010) 'Robust DV-hop algorithm for localization in wireless sensor network', *International Conference on Control Automation and Systems*, Kintex, Korea, pp.2506–2509.

- Li, Z., Dehaene, W. and Gielen, G. (2009) 'A 3-tier UWB-based indoor localization system for ultra-low-power sensor networks', *IEEE Transactions on Wireless Communications*, Vol. 8, No. 6, June, pp.2813–2818.
- López, G., Custodio, V. and Moreno, J.I. (2010) 'LOBIN: E-textile and wireless-sensor-network-based platform for healthcare monitoring in future hospital environments', *IEEE Transactions on Information Technology in Biomedicine*, Vol. 14, No. 6, November, pp.1446–1458.
- Niculescu, D. and Nath, B. (2003) 'DV based Positioning in ad hoc networks', *Journal of Telecommunication Systems*, Vol. 22, No. 1, pp.267–280.
- Ouyang, R., Wong, A. and Woo, K. (2010) 'GPS localization accuracy improvement by fusing terrestrial TOA measurements', *IEEE International Conference on Communications (ICC)*, May, Cape Town, South Africa, pp.1–5.
- Patro, R. (2004) 'Localization in wireless sensor network with mobile beacons', *Proceedings of 23rd IEEE Convention of Electrical and Electronics Engineers in Israel*, September, Tel-Aviv, Israel, pp.22–24.
- Postolache, O.A., Pereira, J.M.D. and Girao, P.M.B.S. (2009) 'Smart sensors network for air quality monitoring applications', *IEEE Transactions on Instrumentation and Measurement*, Vol. 58, No. 9, September, pp.3253–3262.
- Rong, P. and Sichitiu, M. (2006) 'Angle of arrival localization for wireless sensor networks', *Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, USA*, Vol. 1, pp.374–382.
- Sheu, J., Chen, P.C. and Hsu, C.S. (2008) 'A distributed localization scheme for wireless sensor networks with improved grid-scan and vector-based refinement', *IEEE Transactions on Mobile Computing*, Vol. 7, No. 9, pp.1110–1123.
- Sheu, J., Li, J. and Hsu, C. (2006) 'A distributed location estimating algorithm for wireless sensor networks', *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, Vol. 1, June, pp.1–8.
- Tang, L., Wang, K.C., Huang, Y. and Gu, F. (2007) 'Channel characterization and link quality assessment of IEEE 802.15.4-compliant radio for factory environments', *IEEE Transactions on Industrial Informatics*, Vol. 3, No. 2, May, pp.99–110.
- Vishnubhotla, R., Rao P.S., Ladha, A., Kadiyala, S., Narmada, A., Ronanki, B. and Illapakurthi, S. (2010) 'ZigBee based multi-level parking vacancy monitoring system', *2010 IEEE International Conference on Electro/Information Technology*, May, Normal, USA, pp.1–4.
- Voltz, P.J. and Hernandez, D. (2004) 'Maximum likelihood time of arrival estimation for real-time physical location tracking of 802.11a/g mobile stations in indoor environments', *Position Location and Navigation Symposium (PLANS)*, April, California, USA, pp.585–591.
- Zhang, J., Song, G., Qiao, G., Meng, T. and Sun, H. (2011) 'An indoor security system with a jumping robot as the surveillance terminal', *IEEE Transactions on Consumer Electronics*, Vol. 57, No. 4, November, pp.1774–1781.
- Zhang, Z., Xu, G., Li, Y. and Huang, S. (2009) 'DV-hop based self-adaptive positioning in wireless sensor networks', *IEEE Conference on Wireless Communications Networking and Mobile Computing*, Beijing, China, pp.1–4.

## Website

<http://wsnet.gforge.inria.fr/tutorial.html>