Connectivity Based DV-Hop Localization for Internet of Things

Linqing Gui, Zhou Yang, Fu Xiao, Feng Shu, Thierry Val

Connectivity Based DV-Hop Localization for Internet of Things

Linqing Gui, Yang Zhou, Fu Xiao, Feng Shu, Thierry Val

***Abstract*—Due to the cost-effective advantage, range-free localization schemes are attractive for low-cost Internet of Things applications. As a distinguishing range-free scheme, DV-Hop localization can localize those unknown nodes which have less than 3 or even no neighbor anchors. However, the localization results by existing DV-Hop based algorithm are found to be inconsistent with the real connectivity between nodes. The inconsistency inspires us to propose two algorithms to improve localization accuracy. First a Centralized Connectivity-based DV-Hop (CCDV-Hop) algorithm is proposed to optimize the accuracy of DV-Hop localization. Establishing an optimization problem which takes the real connectivity between any two nodes as the constraints, the proposed algorithm can make the localization results conform to the real connectivity. Then an algorithm with lower complexity is proposed, namely Distributed Connectivity-based DV-Hop (DCDV-Hop) algorithm, which can obtain near-optimal localization performance in distributed networks. Without including the connectivity of all nodes, the constraints in the proposed DCDV-Hop algorithm only consider the real connectivity within two hops. Simulation results show that despite higher complexity, the proposed algorithms can achieve much better accuracy than other DV-Hop based methods.**

***Index Terms*—Internet of Things, localization, range-free, DV-Hop.**

I. Introduction

INTERNET of Things (IoT) has been widely used in a variety of applications, such as environment protection, health surveillance, disaster relief and object tracking [1]. IoT usually consists of small sensor nodes that can wirelessly communicate with each other. Typically powered by a tiny battery source which has finite lifetime, sensor nodes have limited transmission power and small communication range [2]. So multi-hop communication is usually required for the data transmission between distant nodes. In multi-hop networks, localization is one of the fundamental issues because the received data could be meaningless if the informations about where they have been measured are unknown [3]. Localization is also essential for the design of energy-efficient location-based techniques such as multi-hop routing and data fusion [4].

In general, existing IoT localization schemes can be classified into two categories: range-based and range-free [5]. In order to localize target nodes (namely unknown nodes) in 2-Dimension space, both these two types of schemes require at least 3 anchor nodes in the network which have already known their own positions. The main principle of range-based schemes is to first measure the distance or direction angle between one unknown node and each anchor node, then to calculate the position of the unknown node by trilateration or triangulation [6]. The distance is usually measured based on time of arrival (TOA) [7], time difference of arrival (TDOA) [8], received signal strength indicator (RSSI) [9] or channel state information (CSI) [10], while the direction angle is obtained by angle of arrival (AOA) estimation [11]. Though more accurate than range-free schemes, the range-based schemes have two major drawbacks. First, the ranging information is easily affected by multipath fading, noise and environment variations. Second, special transceivers are always required to precisely measure TOA or AOA, which increase the overall cost [12]. However, range-free schemes don't possess these drawbacks because they only utilize connectivity information between nodes.

Due to the cost-effective advantage, range-free schemes are preferable for low-cost IoT applications such as the positioning of aged people in a large retirement home, or the behavior analysis of clients in a large shopping mall. For these applications, the localization accuracy of several meters can be accepted especially considering privacy issue. Many range-free schemes have been proposed in the literature, such as Centroid [13], Regulated signature distance (RSD) [14], Concentric Anchor Beacon (CAB) [15], Hitball [16], Distance Vector Hop (DV-Hop) [17] and other DV-Hop based algorithms [18]–[28]. Among these range-free schemes, DV-Hop based localization methods have a distinguished advantage that they can localize the unknown nodes which have less than 3 neighbor anchor nodes in radio range. Thus DV-Hop localization can be applied in the networks with low-density and sparsely-distributed anchor nodes. For example, in a network with total 100 nodes, only 10 are anchors and they are deployed sparsely. Therefore in this paper we focus on DV-Hop localization.

DV-Hop algorithm consists of three steps. First, through the flood of information, all nodes obtain the positions of anchors as well as smallest hop count to each anchor; second, each anchor calculates its distance-per-hop and floods this value; third, every unknown node calculates its estimated distance to

This work is supported in part by the National Natural Science Foundation of China (Grant 61972201, 61602245), the Nature Science Foundation of Jiangsu for Distinguished Young Scientist (Grant BK20170039), and the Natural Science Foundation of Jiangsu Province (Grant BK20190068).

L. Gui , Y. Zhou and F. Shu are with School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, 210094 China. (E-mail: guilinqing@gmail.com, zhou_yang@njust.edu.cn, shufeng@njust.edu.cn.).

Corresponding author: F. Xiao is with the College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, 210003 China, and also with the Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing University of Posts and Telecommunications, Nanjing. (E-mail: xiaof@njupt.edu.cn).

T. Val is with CNRS-IRIT, University of Toulouse, Toulouse, 31703 France (E-mail: val@irit.fr).

each anchor and then estimates its own position. In original DV-Hop algorithm, localization error mainly results from the estimation error of the distance between an unknown node and each anchor node. The real distance between the two aforementioned nodes is the product of hop count between them and the real distance-per-hop between them. However, the real distance-per-hop between the unknown node and each anchor is difficult to obtain, thus, the authors in [17] replaced it with the average distance-per-hop of each anchor. As a result, the estimated distance between the unknown node and each anchor is erroneous.

In order to improve localization accuracy of DV-Hop algorithm, researchers have tried to decrease the estimation error of the distance between the unknown node and each anchor. For example, the authors in [22] used the shuffled frog leaping algorithm (SFLA) to compute distance-per-hop. The authors in [23] proposed a modified DV-Hop localization algorithm based on communication radius dynamic adjustment. This algorithm introduces the improvement of multiple communication radiuses to correct the least hop counts between unknown nodes and anchor nodes, then corrects the distance-per-hop by multiplying weight coefficients. In [24], the authors proposed an improved DV-Hop algorithm based on bacterial foraging optimization (BFO). In this algorithm, the distance-per-hop is calculated by BFO according to the minimum hops of nodes and the position information of anchor nodes. The authors in [25] proposed DV-maxHop algorithm which introduced a control variable MaxHop. When a node receives the position of any anchor whose hop count is greater than MaxHop, the information will be ignored and not propagated further. Through this mechanism, the estimated distance with large error (normally corresponding to two distant nodes) can probably be excluded.

While the above methods focused on the estimation of distance-per-hop at the second step, further improvement can be fulfilled at the third step of DV-Hop or even after all three steps. For example, in [26], the authors focused on the improvement at the third step, proposing RAS DV-Hop algorithm and GOS DV-Hop algorithm. In RAS DV-Hop algorithm, one unknown node first calculated several candidate positions based on different reference anchors and then choosed the best candidate which had the minimum mean square error of estimated distances to anchors. On the other hand, GOS DV-Hop algorithm provided a global optimized solution for the third step of DV-Hop.

All the aforementioned algorithms have a common problem that their localization results are probably inconsistent with the real connectivity between nodes. One main reason is that all the above algorithms hardly checked whether the localization results conformed to the real connectivity between all nodes, let alone correcting the results based on the real connectivity. Therefore, the connectivity derived from the localization results of the above algorithms, which we call the estimated connectivity, are probably different from the real connectivity. This difference between the estimated connectivity and the real connectivity inspires us to propose new algorithms for optimizing the accuracy of DV-Hop localization. If the localization results could be finally in accordance with the real connectivity, it is possible for the proposed algorithms to reach the best accuracy that range-free localization can attain, because the underlying principle of range-free localization is to localize nodes based on their connectivity.

Both the estimated connectivity and the real connectivity are not hard to obtain. The estimated connectivity can be calculated based on the localization results and the communication range, while the real connectivity can be obtained through the communication between nodes. However, the difficulty lies in how to make the localization results conform to the real connectivity of all nodes in the network. This paper not only introduces our finding about the consistency of the connectivity, but also proposes two algorithms to tackle with the above difficulty. The main contribution of this paper is summarized as follows.

(1) To optimize the accuracy of DV-Hop localization, a Centralized Connectivity-based DV-Hop (CCDV-Hop) algorithm is proposed. By establishing an optimization problem which takes the real connectivity between any two nodes as the constraints, the proposed algorithm can make the localization results conform to the real connectivity. To tackle the original optimization problem, SCA (sequential convex approximation) technique and Taylor-expansion based approximation are employed. The numerous connectivity constraints result in relatively high complexity, thus the proposed algorithm is suitable for centralized sensor networks.

(2) To reduce the computation complexity of CCDV-Hop, a Distributed Connectivity-based DV-Hop (DCDV-Hop) algorithm is proposed, which can obtain near-optimal localization performance in distributed networks. Without including the connectivity of all nodes, the constraints of the optimization problem in the proposed algorithm only consider the real connectivity within two hops. In the proposed algorithm, every unknown node needs to iteratively perform two steps, including position exchange step and position estimation step. The former step is to obtain the latest estimated position of the unknown nodes within 2 hops, while the latter step is to calculate the position by solving the corresponding optimization problem.

(3) The performance of two proposed localization algorithms have been thoroughly investigated. The proposed algorithms are compared to latest DV-Hop based algorithms. Their performance have been evaluated in terms of localization accuracy and complexity. Simulation results show that the proposed algorithms can achieve much better accuracy than other methods, though at the cost of higher complexity.

II. Model of Connectivity

Since the main idea of our contribution is to improve localization accuracy by taking full advantage of the connectivity between nodes, in this section, the connectivity of nodes will be modeled in a way associating the connectivity with the positions of nodes.

In DV-Hop localization, the network connectivity between an unknown node and an anchor can be expressed as the minimum hop count between those two nodes [20]. If this concept is extended to any two nodes in the network, the

connectivity of one concerned node can be expressed as the set of minimum hop counts between this concerned node and other node. Thus we have

$$C_i^{real} = \{hop_{i,k}\},\ \forall k \in \{1, 2, \ldots, N\} \text{ and } k \neq i, \quad (1)$$

where C_i^{real} denotes the network connectivity of node i, $hop_{i,k}$ is the smallest hop count between node i and another node k, and N is the total number of nodes.

In fact, beside the minimum hop counts, the connectivity of node i can also be expressed in a simpler way as (2) by a set of neighborhood relationship. Here the neighborhood between two nodes means whether they are in the communication range of each other.

$$C_i^{real} = \{IsNeighbor_{i,k}^{real}\},\ \forall k \in \{1, 2, \ldots, N\},\ k \neq i, \quad (2)$$

where $IsNeighbor_{i,k}^{real} = 1$ if node k and node i can directly communicate with each other; otherwise, $IsNeighbor_{i,k}^{real} = 0$.

Although (2) only denotes the neighborhood of node i, when all nodes are considered (i.e., all values of i ranging from 1 to N are considered), the set of (2) can express the neighborhood between any two nodes in the network. In the same way, the set of (1) with all nodes included can express the minimum hop counts between any two nodes. The set of (2) is equivalent to the set of (1), because the set of neighborhood between any two nodes can be used to produce the whole topology of the network, then the smallest hop counts between any two nodes can be retrieved. Moreover, in practice, compared to hop counts, the neighborhood between nodes is easier to be obtained, because we can know whether one node is in the communication range of another node just by direct signal transmission/reception between those two nodes.

On the other hand, the estimated connectivity of node i can be expressed as

$$C_i^{esti} = \{IsNeighbor_{i,k}^{esti}\},\ \forall k \in \{1, 2, \ldots, N\},\ k \neq i, \quad (3)$$

where $IsNeighbor_{i,k}^{esti}$ is the neighborhood between the estimated positions of node i and node k. $IsNeighbor_{i,k}^{esti}$ can be calculated as

$$\begin{cases} IsNeighbor_{i,k}^{esti} = 1, when\ \|L_i - L_k\| \leq R, \\ IsNeighbor_{i,k}^{esti} = 0, when\ \|L_i - L_k\| > R, \end{cases} \quad (4)$$

where $L_i = (x_i, y_i)^T$ and $L_k = (x_k, y_k)^T$ are the estimated positions of node i and node k by using a DV-Hop based algorithm, respectively, $\|L_i - L_k\|$ is the distance between node i and node k, i.e. $\|L_i - L_k\| = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}$, and R is the communication range of nodes (all nodes are assumed to have the same range).

Then the connectivity difference of node i is defined as

$$\begin{aligned} Conn\_Diff_i &= \|C_i^{esti} - C_i^{real}\| \\ &= \sum_k |IsNeighbor_{i,k}^{esti} - IsNeighbor_{i,k}^{real}|. \end{aligned} \quad (5)$$

As to the existing DV-Hop based algorithms, the connectivity difference $Conn\_Diff_i$ is usually not equal to 0, because the existing algorithms hardly check whether the localization results conform to the real connectivity.

Our objective is to obtain more accurate estimated positions which can conform to the real connectivity between nodes. When the objective is achieved, the relation between the real connectivity and the estimated positions of nodes should satisfy

$$\begin{aligned} &\forall \text{ two nodes } i \text{ and } k, \\ &\begin{cases} \|L_i - L_k\| \leq R, \text{if } IsNeighbor_{i,k}^{real} \text{ is } 1, \\ \|L_i - L_k\| > R, \text{if } IsNeighbor_{i,k}^{real} \text{ is } 0. \end{cases} \end{aligned} \quad (6)$$

The above relation expression between the connectivity and the positions of nodes will be used to construct our proposed algorithms.

III. Proposed DV-Hop Localization algorithms based on Connectivity of Nodes

In this section, we will propose two connectivity-based DV-Hop localization algorithms. One is Centralized Connectivity-based DV-Hop (CCDV-Hop) and the other is Distributed Connectivity-based DV-Hop (DCDV-Hop). The choice between these two algorithms depends on whether there exists in the network a powerful node (with the most energy and high computation ability) or not. If the network has such a powerful node, this node acts as the central node and executes our CCDV-Hop algorithm to estimate the positions of all unknown nodes. Otherwise, since there is no central node, each unknown node executes our DCDV-Hop algorithm by itself to estimate its own position. Next these two proposed algorithms will be illustrated in detail.

A. Proposed CCDV-Hop

Our proposed CCDV-Hop algorithm can be applied in centralized scenarios where there is a central node with the most energy and strong computing power. Based on the connectivity of all nodes in the network, our CCDV-Hop algorithm aims to provide optimal estimated positions for unknown nodes. To achieve this, our CCDV-Hop algorithm first constructs an optimization problem which makes the best use of the real connectivity between nodes, and then solves this complex problem by multiple iterations. In the following, the construction and solution of the optimization problem will be illustrated in detail.

To construct an optimization problem, normally we should first construct an objective function and then give the constraints. However, in our case, before the objective function, our constraints have already been presented as (7) in the previous section. That is, our constraints are the real connectivity between any two nodes. Then we need to construct the objective function.

In order to reduce computation complexity of our algorithm, the objective function is set to minimizing the distance between the estimated position by our CCDV-Hop and that by the original DV-Hop. The reason is illustrated as follows. Generally, multiple iterations are required to solve the optimization problem of our CCDV-Hop algorithm. The start

point of the iterations can be set to the result of original DV-Hop algorithm. With more iterations executed, the solution of the optimization problem should be farther away from the start point. But more iterations also increase the complexity of the algorithm. Therefore, in order to reduce computational complexity of our algorithm, iterations should be reduced, and then the distance between the start point and the solution of the optimization problem will also be reduced.

According to the above illustration, the optimization problem can be expressed as

$$\min \sum_{i=1}^{M}(w_i \times \|L_i - \overline{L_i}\|) \tag{7}$$
$$s.t.\ \forall i \in \{1,2,\ldots,M\}$$
$$\|L_i - L_j\| \leq R,\ (\forall j,\ IsNeighbor_{i,j} = 1) \tag{8}$$
$$\|L_i - L_q\| > R,\ (\forall q,\ IsNeighbor_{i,q} \neq 1) \tag{9}$$

where M is the number of unknown nodes, w_i is weight coefficient, L_i (as optimization variable) represents the coordinates of unknown node i, $\overline{L_i}$ is the localization result of unknown node i by original DV-Hop algorithm, R is the communication range, L_j and L_q represent the coordinates of a neighbor node j and a non-neighbor node q, respectively. It should be noted that L_j and L_q are not always unknown. Only when node j and q are unknown nodes, L_j and L_q are unknown; but when they are anchor nodes, L_j and L_q are already known.

In the following, the weight coefficient w_i will be illustrated in detail. (Here the subscript i denotes the i-th unknown node.) Based on all $\overline{L_i}$ (position results of original DV-hop algorithm) and the positions of all anchors, we can obtain the estimated connectivity of each unknown node at the position $\overline{L_i}$. This estimated connectivity of $\overline{L_i}$, denoted as $C_{\overline{L_i}}$, is probably different from the real connectivity of node i. If their difference is large, the estimation error of $\overline{L_i}$ should also be large. Thus in order to reduce localization error, the adjustment on $\overline{L_i}$ should be large, i.e., the distance between L_i (the optimal variable representing the estimated position of node i) and $\overline{L_i}$ should be large. In this case, w_i should be set to a small value, so that the the distance between L_i and $\overline{L_i}$ will be much less limited by the objective function. Therefore, when the difference between the estimated connectivity of $\overline{L_i}$ and the real connectivity of node i is large, w_i is set to a small value. On the contrary, if the difference is small, since the connectivity of node i at the estimated position $\overline{L_i}$ is already close to the real connectivity, the estimation error of $\overline{L_i}$ should be small, so we don't need to change $\overline{L_i}$ much, then w_i is set to a relatively large value. The expression of w_i is

$$w_i = \frac{1}{\|C_i - C_{\overline{L_i}}\| + 1}, \tag{10}$$

where C_i the real connectivity of node i and $C_{\overline{L_i}}$ is the estimated connectivity of each unknown node at the position $\overline{L_i}$.

The number of constraints in the optimization problem is calculated as follows. If node i has J_i neighbor nodes, then (8) consists of J_i inequations. On the other hand, if node i has Q_i non-neighbor nodes, then (9) consists of Q_i inequations. Since there are in total N nodes in the network, we have $J_i + Q_i = N - 1$. Then if there are M unknown nodes in the network, since each unknown node has $N-1$ constraints, there are totally $M \times (N-1)$ constraints in our optimization problem.

To solve the optimization problem, equivalent conversion should be performed first because it is difficult to directly tackle the original optimization problem. In the original optimization problem, the objective function and all constraints are all expressed in the form of distance calculation which involves the root operation and are hard to tackle. Equivalently we first transform all the distance to the square of distance, then the original optimization problem becomes

$$\min \sum_{i=1}^{M}(w_i \times \|L_i - \overline{L_i}\|^2) \tag{11}$$
$$s.t.\ \forall i \in \{1,2,\ldots,M\}$$
$$\|L_i - L_j\|^2 \leq R^2,\ (\forall j,\ IsNeighbor_{i,j} = 1) \tag{12}$$
$$\|L_i - L_q\|^2 > R^2.\ (\forall q,\ IsNeighbor_{i,q} \neq 1) \tag{13}$$

It can be observed that the objective function in (11) and the inequation constraints in (12) are all convex, while all the inequation constraints in (13) are not convex. To make the above optimization problem tackable, SCA (Sequential Convex Approximation) technique will be employed. Then through multiple iterations of SCA, the approximate optimal solution of the original optimization problem can be derived. Another important conversion is that in every iteration, each non-convex inequation constraints in (13) needs to be transformed to a linear inequation based on first-order Taylor expansion. Since L_q can be the position of either an anchor node or an unknown node, the linear transformation of (13) in the case of anchor node will be different from that in the case of unknown node. In the following, the transformation of (13) in the n_{th} iteration will be illustrated in detail according to the above two cases.

Case 1: L_q is unknown

Since L_q and L_i are both unknown, the left side of each inequality in (13) is a function with 2 variables L_i and L_q. Then the first-order Taylor expansion of the left side of (13) should be performed upon L_q and L_i simultaneously. Similar to [29], the Taylor expansion point at n_{th} iteration is set to the approximate optimal solution at the $(n-1)_{\text{th}}$ iteration, denoted as $L_i(n-1)$ and $L_q(n-1)$. Thus the first-order Taylor expansion of $\|L_i - L_k\|^2$ at $L_i = L_i(n-1)$ and $L_q = L_q(n-1)$ can be expressed as

$$\begin{aligned}
&\|L_i - L_q\|^2 \\
&\approx \|L_i(n-1) - L_q(n-1)\|^2 \\
&\quad + (L_i - L_i(n-1)) \cdot \frac{\partial(\|L_i(n-1) - L_q(n-1)\|^2)}{\partial L_i} \\
&\quad + (L_q - L_q(n-1)) \cdot \frac{\partial(\|L_i(n-1) - L_q(n-1)\|^2)}{\partial L_q} \\
&= 2(L_i(n-1) - L_q(n-1))^T \cdot (L_i - L_q) \\
&\quad - \|L_i(n-1) - L_q(n-1)\|^2.
\end{aligned} \tag{14}$$

Substituting (14) into the left of (12), we can transform each

non-convex constraint in (12) into a linear constraint as

$$\begin{aligned} &2(L_i(n-1)-L_q(n-1))^T\cdot(L_i-L_q)- \\ &\|L_i(n-1)-L_q(n-1)\|^2 > R^2. \\ &(\forall q,\ IsNeighbor_{i,q}=0) \end{aligned} \tag{15}$$

Case 2: L_q is already known (q corresponding to an anchor node)

Compared to the unknown case, the transformation of (12) in this case will be simpler because there is only one variable L_i on the left side of (13). Thus the first-order Taylor expansion of the left side of (13) will be performed only upon L_i. The Taylor expansion point is still set to $L_i(n-1)$ which is the approximate optimal solution at the $(n-1)_{\text{th}}$ iteration. So the first-order Taylor expansion of $\|L_i-L_q\|^2$ at $L_i=L_i(n-1)$ can be expressed as

$$\begin{aligned} &\|L_i-L_q\|^2 \\ &\approx \|L_i(n-1)-L_q\|^2 \\ &\quad+(L_i-L_i(n-1))\cdot\frac{\partial(\|L_i(n-1)-L_q\|^2)}{\partial L_i} \\ &=\|L_i(n-1)-L_q\|^2 \\ &\quad+2(L_i(n-1)-L_q)^T\cdot(L_i-L_i(n-1)). \end{aligned} \tag{16}$$

Substituted with (III-A), each non-convex constraint in (13) can be transformed into a linear constraint as

$$\begin{aligned} &2(L_i(n-1)-L_q)^T\cdot(L_i-L_i(n-1))+ \\ &\|L_i(n-1)-L_q\|^2 > R^2. \\ &(\forall q,\ IsNeighbor_{i,q}=0) \end{aligned} \tag{17}$$

Combining the above two cases, at the n_{th} iteration, $\forall q$, satisfying $IsNeighbor_{i,q}=0$, can be transformed to

$$\begin{cases} \text{If } L_q \text{ is unknown,} \\ 2(L_i(n-1)-L_q(n-1))^T\cdot(L_i-L_q)- \\ \quad\|L_i(n-1)-L_q(n-1)\|^2>R^2, \\ \text{if } L_q \text{ is known,} \\ 2(L_i(n-1)-L_q)^T\cdot(L_i-L_i(n-1))+ \\ \quad\|L_i(n-1)-L_q\|^2>R^2. \end{cases} \tag{18}$$

To solve the new optimization problem comprising of (11), (12) and (18), we can use convex optimization tools such as CVX (a Matlab software for disciplined convex programming). The corresponding solution denoted as $L_i(n)$ $(i=1,2,\dots M)$ represents the estimated position of all unknown nodes at the n_{th} iteration. Then in the $(n+1)_{\text{th}}$ iteration, $L_i(n)$ and $L_q(n)$ (if q corresponds to unknown node) will be used as the expansion point of the first-order Taylor expansion on (13).

In SCA, the aforementioned process at each iteration will repeat multiple times until the objective function satisfies the end condition. As the end condition, the convergence rate, defined as the difference between the objective function values of two adjacent iterations, should go below a preset threshold. To verify the achievability of the end condition, we prove the convergence based on the reduction to absurdity as follows. If the aforementioned multiple-iteration process cannot be convergent, the value of the objective function in (9) is always no less than 0. Therefore, the aforementioned multiple-iteration process should be convergent. When the objective function meets the end condition, the approximate optimal solution of the original optimization problem can be obtained, which also gives the estimated positions of all unknown nodes. The whole procedure can be summarized as Algorithm 1.

Algorithm 1 CCDV-Hop algorithm

1. Given the distribution of nodes and the radio range;
2. Determine the real connectivity of each node;
3. Obtain $\overline{L}_i$ (the estimated position of each unknown node i by the original DV-Hop algorithm;
4. Initialize $L_i(0)=\overline{L}_i$, $L_q(0)=\overline{L}_q$ (if L_q is unknown) or L_q (if L_q is known);
5. Repeat
   1) transform (13) to (18) by the first-order Taylor expansion at the n_{th} iteration;
   2) solve the new optimization problem comprised of (11), (12) and(18) by CVX;
   3) obtain the optimal solution $L_i(n)$;
   4) calculate objective function $f_n=\sum_{i=1}^{M} w_i\|L_i(n)-\overline{L}_i\|^2$;
   5) use $L_i(n)$ and $L_q(n)$ for the Taylor expansion at next iteration.
6. Until $\mid f_n-f_{n-1}\mid$<preset threshold;
7. Return $L_i^*=L_i(n)$.

In the following analysis, the computational complexity of our CCDV-Hop algorithm is derived. To solve our convex problem comprised of (11), (12), (18), CVX tools usually convert it into a SOCP (Second-Order Cone Programming) problem as

$$\min_{(L_1,L_2,\dots,L_M,z)} z \tag{19}$$

$$s.t.$$

$$\|\sqrt{w_1}(L_1-\overline{L}_1),\dots,\sqrt{w_M}(L_M-\overline{L}_M),\frac{z-1}{2}\|\le\frac{z+1}{2}, \tag{20}$$

$$\|L_i-L_j\|\le R,\ (\forall j,\ IsNeighbor_{i,j}=1), \tag{21}$$

$$a_qL_i+b_q>R^2,\ (\forall q,\ IsNeighbor_{i,q}\neq 1), \tag{22}$$

where the parameters a_q and b_q can be obtained from (18).

According to [30], the complexity of solving a SOCP problem is derived as $\mathcal{O}(\sqrt{N_{cons}}n^2\sum_k n_k)$, where N_{cons} is the number of constraints, $\sqrt{N_{cons}}$ represents the upper bound of the number of iterations, n is the total dimensions of all optimization variables, n_k is the dimension of the k-th constraint.

As to the above problem, the number of constraints in (20) is 1, the number of constraints in (21) is denoted as N_{neigh}, while the number of constraints in (22) is denoted as $N_{non\text{–}neigh}$. So $N_{cons}=1+N_{neigh}+N_{non\text{–}neigh}$. Since the above problem has $2M+1$ scalar optimization variables, n equals $2M+1$. From the constraints, it can be found that $n_1=2,\ n_2=n_3=\dots=n_{N_{neigh}+1}=2,\ n_{N_{neigh}+2}=\dots=n_{N_{non-neigh}+N_{neigh}+1}=1$. Then the complexity of CCDV-Hop algorithm is calculated as

$$Comp\_CCDV-Hop=$$

$$\mathcal{O}(\sqrt{1+N_{neigh}+N_{non-neigh}}(2M+1)^2(2+2N_{neigh}+N_{non-neigh})). \tag{23}$$

Given a sensor network, N_{neigh} and $N_{non-neigh}$ can be regarded to be proportioned to N, so the above complexity can be simplified to $\mathcal{O}(N^{1.5}M^2)$. Since M is a portion of N, the complexity can be further expressed as $\mathcal{O}(N^{3.5})$.

B. Proposed DCDV-Hop algorithm

Making best use of the connectivity of all nodes, our proposed CCDV-Hop algorithm is supposed to provide approximate optimal estimated positions for all unknown nodes. However, the computation complexity of CCDV-Hop is high because it not only has many constraints but also requires multiple iterations to solve the optimization problem. In order to reduce the computation complexity but keep nearly similar localization accuracy, DCDV-Hop is proposed in this section.

As a distributed algorithm, our DCDV-Hop is executed by each unknown node which optimizes its estimated position based on its connectivity. However, here the connectivity is a two-hop connectivity defined as

$$C_i^{2hop} = IsNeighbor_{i,k},\ \forall\ k \text{ within two hops of node } i. \tag{24}$$

This two-hop connectivity can be regarded as a part of the complete network connectivity expressed in (7), but it is the most important part. Since all the neighbor nodes and the nearest non-neighbor nodes are considered in the two-hop connectivity, the assemble of the two-hop connectivity of all nodes is approximate to the complete network connectivity.

The two-hop connectivity should be acquired at the first beginning of our DCDV-Hop algorithm because each unknown node needs to utilize the acquired connectivity to estimate the position. To obtain the two-hop connectivity, node i only needs to communicate with the nodes within two hops. Subsequently, the communication traffic of our DCDV-Hop will be much less than that of CCDV-Hop because the latter requires the connectivity between node i and any other node. Moreover, the small-scale connectivity also helps to reduce the computation complexity of our DCDV-Hop because the two-hop connectivity generates much fewer constraints for the optimization problem.

The main principle of our DCDV-Hop algorithm is briefly presented as follows. Each unknown node needs to iteratively perform two steps. During each iteration, the first step is position exchange while the second step is position estimation. At the first step, each unknown node N_i broadcasts its current estimated position (obtained from last iteration) to the nodes within 2 hops. It should be noted that, the initial estimated position of N_i can be obtained by the original DV-Hop algorithm. After the first step, every node within 2 hops of N_i should be aware of the current estimated position of N_i. Then at the second step, based on the estimated positions of nodes within 2 hops as well as the two-hop connectivity, each unknown node constructs and solves an optimization problem, obtaining its new estimated position. The new position will be taken as its current estimated position at the first step in next iteration. In the following, the construction and solution of the optimization problem in our DCDV-Hop algorithm will be illustrated in detail.

In DCDV-Hop, during each iteration, each unknown node establishs an optimization problem similar to that in (11, 12, 13) but much simpler, because the unknown node only needs to estimate its own position upon its 2-hop connectivity. Thus during each iteration, each unknown node N_i builds an optimization problem as

$$\min_{L_i} \|L_i - \overline{L}_i\| \tag{25}$$

$$s.t. \|L_i - L_j\| \le R, \tag{26}$$
$$(\forall j,\ IsNeighbor_{i,j} = 1)$$
$$\|L_i - L_q\| > R, \tag{27}$$
$$(\forall q \text{ within two hops},\ IsNeighbor_{i,q} = 0)$$

where L_i represents the coordinates of unknown node i, L_i is i's estimated position by last iteration (during the first iteration $\overline{L}_i$ is node i's estimated position by original DV-Hop algorithm), R is the communication range, L_j and L_q represent the coordinates of a neighbor node j and a non-neighbor node q, respectively. It should be noted that, if j and q correspond to unknown nodes, L_j and L_q will be assigned with their estimated positions obtained from last iteration.

The number of constraints in the optimization problem is calculated as follows. If node i has J_i neighbor nodes, then (8) consists of J_i inequations. On the other hand, if there are T_i non-neighbor nodes which are 2 hops away from node i, then (30) consists of T_i inequations. Thus there are $J_i + T_i$ constraints in the above optimization problem.

To solve the optimization problem, similar to CCDV-Hop algorithm, we first equivalently transform all the distance to the square of distance, then the original optimization problem becomes

$$\min_{L_i} \|L_i - \overline{L}_i\|^2 \tag{28}$$

$$s.t. \|L_i - L_j\|^2 \le R^2, \tag{29}$$
$$(\forall j,\ IsNeighbor_{i,j} = 1)$$
$$\|L_i - L_q\|^2 > R^2. \tag{30}$$
$$(\forall q \text{ within two hops},\ IsNeighbor_{i,q} = 0)$$

It can be observed that the objective function in (28) and the inequation constraints in (29) are all convex, while all the inequation constraints in (30) are not convex. Similar to CCDV-Hop algorithm, we can derive the solution of the original optimization problem through multiple iterations of SCA. And in every SCA iteration, we need also to transform each non-convex inequation constraints in (30) to a linear inequation based on first-order Taylor expansion. The transformation of (30) in the n_{th} iteration will be illustrated in the following.

Since there is only one variable L_i on the left side of (30), the first-order Taylor expansion of the left side of (30) will be performed upon L_i. At the n_{th} SCA iteration, the Taylor expansion point is set to the solution at the $(n-1)_{\text{th}}$ SCA iteration, denoted as $L_i(n-1)$. Thus the first-order Taylor expansion of $\|L_i - L_q\|^2$ at $L_i = L_i(n-1)$ can be expressed

as

$$\begin{aligned}&\|L_i - L_q\|^2\\ &\approx \|L_i(n-1) - L_q\|^2\\ &\quad + (L_i - L_i(n-1)) \cdot \frac{\partial(\|L_i(n-1) - L_q\|^2)}{\partial L_i}\\ &= \|L_i(n-1) - L_q\|^2\\ &\quad + 2(L_i(n-1) - L_q)^T \cdot (L_i - L_i(n-1)).\end{aligned} \tag{31}$$

Substituting (31) into the left of (30), we can transform each non-convex constraint in (30) into a linear constraint as

$$\begin{aligned}&2(L_i(n-1) - L_q)^T \cdot (L_i - L_i(n-1)) +\\ &\|L_i(n-1) - L_q\|^2 > R^2.\\ &(\forall q \text{ within two hops, } IsNeighbor_{i,q} = 0)\end{aligned} \tag{32}$$

To solve the new optimization problem comprising of (28), (29) and (32), we can still use CVX. The corresponding solution denoted as $L_i(n)$ represents the estimated position of node i at the n_{th} SCA iteration. Then in the $(n+1)_{\text{th}}$ iteration, $L_i(n)$ will be used as the Taylor expansion point to transform (30). The aforementioned process at each SCA iteration will repeat multiple times. Then the acquired solution of the optimization problem will be taken as node i's current estimated position to be used at the first step in the next outer-iteration. The whole procedure can be summarized as Algorithm 2.

Algorithm 2 DCDV-Hop algorithm

1. Given the distribution of nodes and the radio range R;
2. Determine the 2-hop connectivity of each unknown node i;
3. Initialize $\overline{L}_i$ as the estimated position by original DV-Hop algorithm;
4. Repeat outer-iterations;
   - position exchange: unknown node i receives estimated positions of the nodes within 2 hops;
   - position calculation: Repeat inner iterations to solve (28, 29, 30);
     1) transform (30) to (31) by the first-order Taylor expansion at the n_{th} iteration;
     2) solve the new optimization problem comprised of (28), (29) and (31) by CVX;
     3) obtain the optimal solution $L_i(n)$;
     4) use $L_i(n)$ and $L_q(n)$ for the Taylor expansion at next iteration;

     Until the number of inner-iterations > preset threshold.
5. Until the number of outer-iterations > preset threshold;
6. Return $L_i^* = L_i(n)$.

The computational complexity of our DCDV-Hop algorithm is derived as follows. For DCDV-Hop algorithm, the CVX tools convert our convex problem comprised of (28), (29), (32) into a SOCP problems as

$$\min_{(L_i,z)} \ z \tag{33}$$

$$s.t.$$

$$\|\sqrt{w_i}(L_i - \overline{L}_i), \frac{z-1}{2}\| \le \frac{z+1}{2}, \tag{34}$$

$$\|L_i - L_j\| \le R, (\forall j,\ IsNeighbor_{i,j} = 1) \tag{35}$$

$$a_q L_i + b_q > R^2, (\forall q \text{ within two hops}, IsNeighbor_{i,q} = 0) \tag{36}$$

where the parameters a_q and b_q can be obtained from (32).

The number of constraints in (34) is 1, the number of constraints in (35) is J_i, while the number of constraints in (36) is T_i. So $N_{cons} = 1 + J_i + T_i$. Since the above problem has 3 scalar optimization variables, n equals 3. From the constraints, it can be found that $n_1 = 2,\ n_2 = n_3 = ... = n_{J_i+1} = 2,\ n_{J_i+2} = ... = n_{j_i+T_i+1} = 1$.

Since the complexity of solving a SOCP problem is $\mathcal{O}(\sqrt{N_{cons}} n^2 \sum_k n_k)$, the complexity of DCDV-Hop algorithm is calculated as

$$Comp\_DCDV\text{–}Hop = \mathcal{O}(9\sqrt{J_i+T_i+1}(2J_i+T_i+2)). \tag{37}$$

Given a sensor network, the above complexity can be simplified to $\mathcal{O}((J_i + T_i)^{1.5})$. Since $J_i + T_i$ should be much smaller than N, the complexity of DCDV-Hop algorithm is much smaller than that of CCDV-Hop algorithm.

IV. Simulation Results and Analysis

In this section, we first discuss the localization accuracy of our proposed CCDV-Hop and DCDV-Hop algorithms by two specific examples, then investigate the average performance of our algorithms through Monte Carlo simulations by using Matlab. Our algorithms are compared to not only the original DV-Hop algorithm [17] but also some latest algorithms such as DV-maxHop [25], RAS DV-Hop [26] , GOS DV-Hop [26] and an improved DV-Hop [27].

A. Simulation parameters

The two specific examples correspond to a small scale network which has 6 nodes and a middle scale network with 30 nodes. As for the example of small scale network, the main simulation parameters are listed in Table I. Sensor nodes are uniform-randomly distributed in a 40m×40m region. The node density defined as the average number of nodes per 400m^2 is 1.5.

TABLE I: Simulation parameters in small scale network

Parameters	Values
Network area	40m×40m
Node distribution density	1.5 per 400m^2
Communication range in ideal channel	20m
Total number of sensor nodes	6
Number of anchor nodes	3

As for the example of middle scale network, the main simulation parameters are listed in Table II. Sensor nodes are uniform-randomly distributed in a 60m×60m region with the node density of 1 per 400m^2.

As for the general case, in order to obtain the average performance of concerned algorithms, the simulation is executed ten thousand times. Each time a number of sensor nodes are uniform-randomly distributed in a 100m×100m region, anchor nodes are randomly selected from all those nodes, and then the

TABLE II: Simulation parameters in middle scale network

Parameters	Values
Network area	60m×60m
Node density (low)	1 per 400m^2
Communication range in ideal channel	20m
Total number of sensor nodes	9
Number of anchor nodes	3

unknown nodes use the aforementioned algorithms to estimate their positions. The average localization error, computation time and network traffic of each concerned algorithm will be compared and analyzed.

B. Simulation results

The localization results of the small-scale network example is shown in Fig. 1. In the example, A_1, A_2 and A_3 are anchor

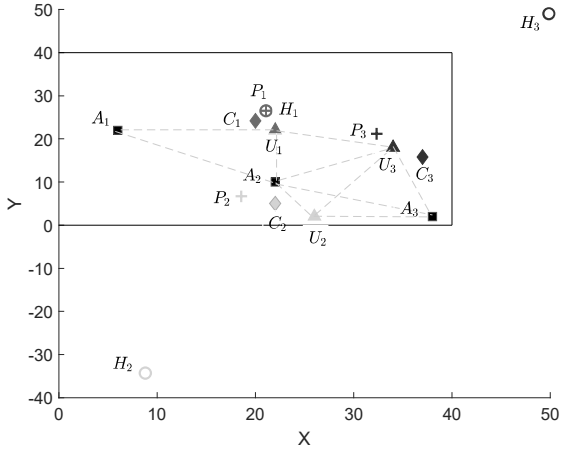


Fig. 1. Example of small scale network

nodes whose coordinates are (6, 22), (22, 10) and (38, 2), respectively. U_1, U_2 and U_3 are unknown nodes whose real positions are (22, 22), (26, 2) and (34, 18), respectively. H_1 (21.07, 26.52), H_2 (8.79, -34.3) and H_3 (49.86, 49.03) are the estimated positions of the unknown nodes by the original DV-Hop algorithm. From Fig. 1, we can see that the connectivity between H_1 and H_3 does not satisfy their real connectivity, neither does the connectivity between H_2 and A_2, nor the connectivity between H_3 and A_2. Then our CCDV-Hop and DCDV-Hop algorithms can be used to adjust the localization results of DV-Hop. P_1 (21.07, 26.52), P_2 (18.55, 6.68) and P_3 (32.33, 21.18) are the localization results by our DCDV-Hop algorithm, while C_1 (20, 24.2), C_2 (22, 5) and C_3 (37, 15.8) are the localization results of our CCDV-Hop algorithm. The localization results by RAS DV-Hop is the same as DV-Hop, while the localization results of the three unknown nodes by GOS DV-Hop are respectively (23.45, 25.61), (11.3, -32.67) and (46.86, 45.7). Thus the average localization errors of DV-Hop, RAS DV-Hop, GOS DV-Hop, our DCDV-Hop and our CCDV-Hop are respectively 26.55m, 26.55m, 22.41m, 5.68m and 3.90m. So given the localization error by DV-Hop as the reference, the localization accuracy has been improved

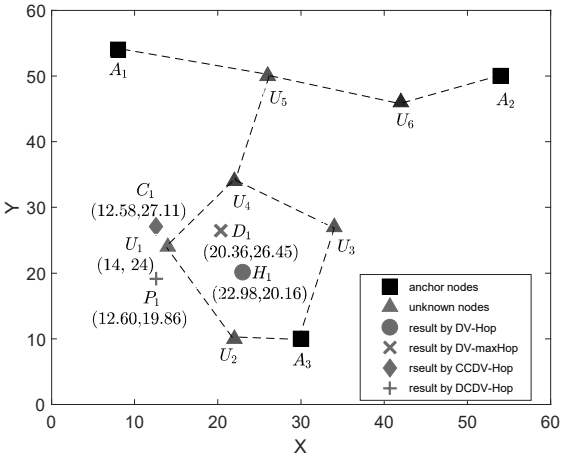


Fig. 2. Example of middle scale network

by 15.6%, 78.6% and 85.3% in GOS DV-Hop, our DCDV-Hop and our CCDV-Hop, respectively. Our CCDV-Hop is found to be more accurate than our DCDV-Hop, because the connectivity between any two nodes are all utilized in CCDV-Hop.

Similarly the localization results of the middle-scale network example are shown in Fig. 2. Since it is lengthy to present the localization results of all unknown nodes, only one unknown node is taken as an example. In Fig. 2, U_1 is the real location of the unknown node, H_1 is the localization result by DV-Hop, P_1 is the localization result by our DCDV-Hop, and C_1 is the localization result by our CCDV-Hop. The localization errors by the original DV-Hop and DV-maxHop are 9.76m and 6.82m, respectively. But the localization errors by our DCDV-Hop and CCDV-Hop are much smaller, i.e., 4.36m and 3.42m, respectively.

In general cases, the average localization errors by each concerned algorithm is shown in Fig. 3. The ratio of anchor nodes to all nodes is fixed as 20%. It can be observed that when the density of nodes increases, the localization errors of all concerned algorithms decrease. The reason is that with an increasing node distribution density, the network has more nodes which can not only produce more accurate distance-per-hop but also bring more connectivity information. It can also be observed that the localization errors of both DCDV-Hop and CCDV-Hop are much smaller than other algorithms. The reason is that these two proposed algorithms endeavour to make the localization results conform to the connectivity between nodes, while this consistency of localization results and connectivity has not been considered by other algorithms.

Fig. 4 shows how the average localization errors change with different ratios of anchor nodes. The density of nodes is set to 0.005 per m^2. We can observe that the localization errors of all concerned algorithms decrease with the ratio of anchor nodes. Two reasons are given in the following. First, when the number of anchor nodes increases, the average hop count between each unknown node and each anchor node becomes smaller, so is the estimation error of their distance. Second, more anchor node can provide each unknown node with more

supplementary information which is reliable, determinate and helpful for its localization. It can also be observed that localization errors of our DCDV-Hop and CCDV-Hop algorithms are much smaller than the other algorithms.

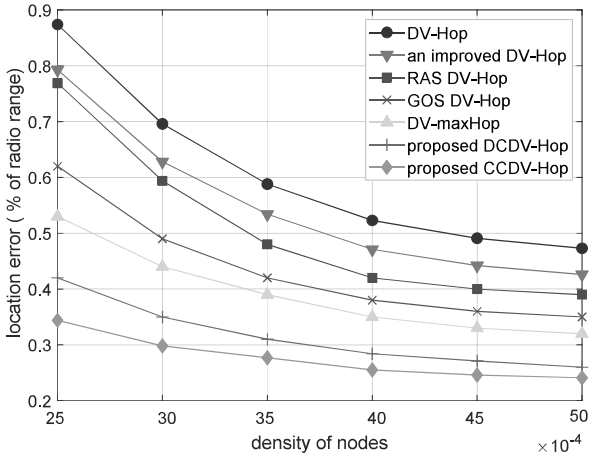


Fig. 3. Localization error with different densities of nodes

In order to evaluate the computational complexity of the concerned algorithms, the computation time consumed by the algorithms are simulated and shown in Fig. 5. The total number of nodes is set to 100. The computer used for the simulation has a 3.3GHz CPU and a 4GB RAM. It can be observed that when the ratio of anchor nodes increases, the computation time of each algorithm becomes larger. The reason is that each algorithm estimates the position of an unknown node based on the positions of anchor nodes as well the connectivity or hop count between the unknown node and each anchor node. It can also be observed that the computation time of CCDV-Hop is much higher than other algorithms. The reason is that in CCDV-Hop, the central node is responsible for the whole computation task, i.e., that node has to estimate the positions of all unknown nodes by itself.

In order to evaluate network cost of the concerned algorithms, the number of transmitted packets by the algorithms are simulated and shown in Fig. 6. The ratio of anchor nodes is set to 10%. Since RAS DV-Hop, GOS DV-Hop and DV-maxHop focus on the computational optimization at the third step while sharing the same Step 1 and Step 2 as the original DV-Hop, their communication costs mainly come from the network-wide broadcast of anchor information during the first two steps. However, at the third step, in order to obtain the connectivity between nodes, our CCDV-Hop and DCDV-Hop algorithms require additional data transmissions.

The approximate number of additional transmissions in our two proposed algorithms will be estimated in the following. As for our CCDV-Hop algorithm, additional transmissions are needed for each unknown node to send its neighbor list to the central node. If only one unknown node sends its neighbor list to the center node, the number of transmitted packets is supposed to equal the minimum hop counts between the unknown node and the center node, because the intermediate nodes on the shortest path will do nothing but relay the neighbor list. However, since all unknown nodes are required

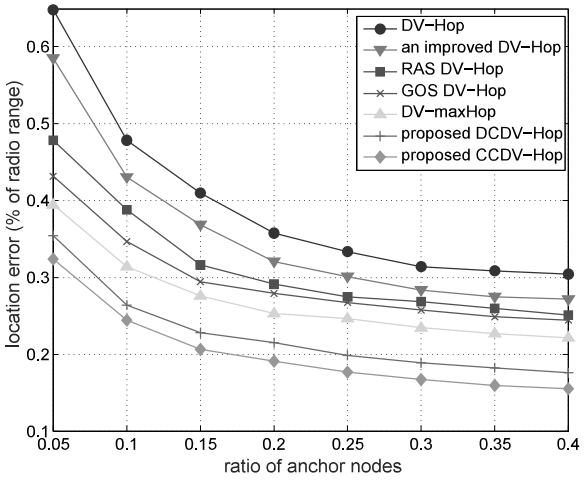


Fig. 4. Localization error with different ratios of anchor nodes

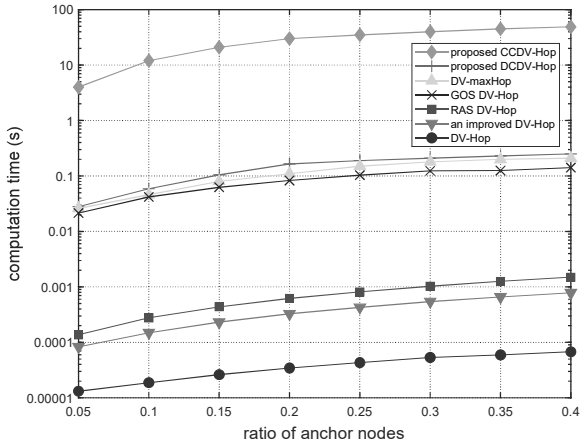


Fig. 5. Computation time in different ratio of anchor nodes (for CCDV-Hop, it is time consumed by the central node; for other algorithms, it is average time consumed by each node)

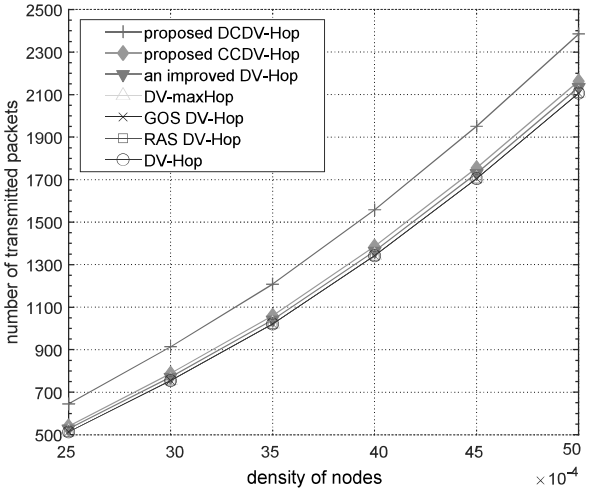


Fig. 6. The number of transmitted packets in different density of nodes (the number of outer iterations in DCDV-Hop is 5)

to send their lists to the center node, in one transmission an unknown node can not only relay the neighbor list it received from another unknown node, but also send its own neighbor list. In other words, when one unknown node sends a packet to one neighbor node (toward the center node), this packet contains a combined neighbor list which contains both its own neighbor information and the neighbor information received from other unknown nodes. Therefore, each unknown node only needs to send the combined neighbor list to one neighbor node. Then if this neighbor node is an anchor node, it just relays the list toward the center node; otherwise, this neighbor unknown node will first adds its own neighbor information into the list and then relay the list toward the center node. As a result, if N denotes the number of nodes in the network, the number of additional transmitted packets in CCDV-Hop is about N-1, since the centre node is always the receiver. So we have

$$C_{CCDV-Hop} = N - 1\,. \tag{38}$$

Although the build of the table of neighbors for each unknown node also requires communication between nodes, the communication can be well designed so that it will not introduce additional transmissions. For example, the build of the table of neighbors can be incidentally fulfilled at the first step of DV-Hop, because at the first step all nodes help each anchor node to broadcast its position throughout the network.

The simulation results shown in Fig. 6 can also verify the derived result in (38). For example, when the density of nodes is 0.0025 per m^2, the number of nodes is 25. Then according to (38), the number of additional transmitted packets is about 25, similar to the simulation result in Fig. 6, which is around 27 (27=540-513, where 540 and 513 are the number of transmitted packets by CCDV-Hop and DV-maxHop, respectively).

As for our DCDV-Hop algorithm, at each iteration, additional transmissions are required for each unknown node to send its estimated position to the nodes within 2 hops. Similarly, using the transmission strategy in CCDV-Hop, each unknown node broadcasts a combined packet that contains its own estimated position and the estimated positions it received from its neighbors. So the number of additional transmitted packets in DCDV-Hop is about

$$C_{DCDV-Hop} = num_{itera} \times N, \tag{39}$$

where num_{itera} denotes the number of iterations and N is the number of nodes. The simulation results in Fig. 6 can also help to verify the derived result in (39). For example, when the density of nodes is 0.0025 per m^2, num_{itera} is set to 5, according to (39) the number of additional transmitted packets is about $5 \times 25 = 125$, similar to the simulation result in Fig. 6, which is around 132 (132=645-513, where 645 and 513 are the number of transmitted packets by DCDV-Hop and DV-maxHop, respectively). We can observe that the overall network cost of DCDV-Hop is larger than that of CCDV-Hop. The reason is that DCDV-Hop requires multiple iterations, while at each iteration the updated position (estimated at last iteration) of each unknown node needs to be transmitted to the nodes within 2 hops.

The transmit power can influence the localization accuracy. All nodes in the network are assumed to have the same transmit power and the same communication range. If the transmit power increases, the communication range of each node also increases and then each node may have more neighbor nodes. When the transmit power increases from a low level to a medium level, the number of neighbor nodes around one node also increases from a small number to some medium number (e.g., from 1 to 3). In this case, the localization accuracy can be improved by the increase of transmit power, because the new-added neighbor nodes can help to localize each unknown node. But if the transmit power is high, around each node there are already many neighbor nodes. In this case, the increase of transmit power may have limited effect on accuracy improvement, as mentioned in [31].

V. Conclusion

Although DV-Hop localization scheme has its distinguishing capability to localize unknown nodes which have less than 3 or even no neighbor anchors, its localization accuracy is not so satisfactory. We find that the localization results by existing DV-Hop based algorithm are inconsistent with the real connectivity between nodes. So in this paper we propose two algorithms to rehabilitate the connectivity. First a Centralized Connectivity-based DV-Hop (CCDV-Hop) algorithm is proposed to optimize the accuracy of DV-Hop localization by taking the real connectivity between any two nodes as the constraints. SCA (sequential convex approximation) technique and Taylor-expansion based approximation are employed to solve the corresponding optimization problem. Then in order to obtain near-optimal localization performance in distributed networks, a algorithm with lower complexity is proposed, namely Distributed Connectivity-based DV-Hop (DCDV-Hop) algorithm. Not including the connectivity of all nodes, the constraints in the proposed DCDV-Hop algorithm only consider the real connectivity within two hops. Simulation results show that with higher complexity, the proposed algorithms can achieve much better accuracy than other DV-Hop based methods.

References

[1] F. Xiao, Z. Wang, N. Ye, R. Wang, and X. Y. Li, "One more tag enables fine-grained rfid localization and tracking," *IEEE/ACM Transactions on Networking*, vol. 26, no. 1, pp. 161–174, Feb 2018.

[2] H. Zhu, F. Xiao, L. Sun, R. Wang, and P. Yang, "R-ttwd: Robust device-free through-the-wall detection of moving human with wifi," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1090–1103, May 2017.

[3] Y. Zhao, X. Fan, C. Xu, and X. Li, "Er-crlb: An extended recursive cramĺerĺcrao lower bound fundamental analysis method for indoor localization systems," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1605–1618, Feb 2017.

[4] J. Cota-Ruiz, P. Rivas-Perea, E. Sifuentes, and R. Gonzalez-Landaeta, "A recursive shortest path routing algorithm with application for wireless sensor network localization," *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4631–4637, June 2016.

[5] L. Shangguan, Z. Yang, A. X. Liu, Z. Zhou, and Y. Liu, "Stpp: Spatial-temporal phase profiling-based method for relative rfid tag localization," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 596–609, Feb 2017.

[6] C. Wu, Z. Yang, Z. Zhou, Y. Liu, and M. Liu, "Mitigating large errors in wifi-based indoor localization for smartphones," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6246–6257, July 2017.

[7] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2409–2419, March 2018.

[8] F. Despaux, K. Jaffres-Runser, A. van den Bossche, and T. Val, "Accurate and platform-agnostic time-of-flight estimation in ultra-wide band," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2016, pp. 1–7.

[9] J. Tiemann and C. Wietfeld, "Scalable and precise multi-uav indoor navigation using tdoa-based uwb localization," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Sept 2017, pp. 1–7.

[10] L. Gui, M. Yang, H. Yu, J. Li, F. Shu, and F. Xiao, "A cramer-rao lower bound of csi-based indoor localization," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2814–2818, March 2018.

[11] F. Shu, Y. Qin, T. Liu, L. Gui, Y. Zhang, J. Li, and Z. Han, "Low-complexity and high-resolution doa estimation for hybrid analog and digital massive mimo receive array," *IEEE Transactions on Communications*, vol. 66, no. 6, pp. 2487–2501, June 2018.

[12] L. Gui, T. Val, A. Wei, and R. Dalce, "Improvement of range-free localization technology by a novel DV-hop protocol in wireless sensor networks," *Ad Hoc Netw.*, vol. 24, no. PB, pp. 55–73, Jan. 2015.

[13] R. K. Patro, "Localization in wireless sensor network with mobile beacons," in *2004 23rd IEEE Convention of Electrical and Electronics Engineers in Israel*, Sept 2004, pp. 22–24.

[14] Z. Zhong and T. He, "Rsd: A metric for achieving range-free localization beyond connectivity," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 11, pp. 1943–1951, Nov 2011.

[15] V. Vivekanandan and V. W. S. Wong, "Concentric anchor beacon localization algorithm for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2733–2744, Sept 2007.

[16] J. F. Huang, G. Y. Chang, and G. H. Chen, "A historical-beacon-aided localization algorithm for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 6, pp. 1109–1122, June 2015.

[17] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1, pp. 267–280, Jan 2003.

[18] K. Chen, Z. hua Wang, M. Lin, and M. Yu, "An improved dv-hop localization algorithm for wireless sensor networks," in *IET International Conference on Wireless Sensor Network 2010 (IET-WSN 2010)*, Nov 2010, pp. 255–259.

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

[20] L. Gui, T. Val, and A. Wei, "Improving localization accuracy using selective 3-anchor DV-Hop algorithm," in *2011 IEEE Vehicular Technology Conference (VTC Fall)*, Sept 2011, pp. 1–5.

[21] S. Kumar and D. K. Lobiyal, "An advanced DV-Hop localization algorithm for wireless sensor networks," *Wireless Personal Communications*, vol. 71, no. 2, pp. 1365–1385, Jul 2013.

[22] Y. Zhang and Z. Zhu, "A novel dv-hop method for localization of network nodes," in *2016 35th Chinese Control Conference (CCC)*, July 2016, pp. 8346–8351.

[23] J. Zhang and L. Zhang, "A modified dv-hop localization algorithm based on communication radius dynamic adjustment," in *2017 9th International Conference on Advanced Infocomm Technology (ICAIT)*, Nov 2017, pp. 260–264.

[24] H. Huang, H. Chen, S. Cheng, and F. Li, "An improved dv-hop algorithm for indoor positioning based on bacterial foraging optimization," in *2016 8th International Conference on Wireless Communications Signal Processing (WCSP)*, Oct 2016, pp. 1–5.

[25] F. Shahzad, T. R. Sheltami, and E. M. Shakshuki, "DV-maxHop: A fast and accurate range-free localization algorithm for anisotropic wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2494–2505, Sept 2017.

[26] L. Gui, X. Zhang, Q. Ding, F. Shu, and A. Wei, "Reference anchor selection and global optimized solution for dv-hop localization in wireless sensor networks," *Wireless Personal Communications*, vol. 96, no. 4, pp. 5995–6005, Oct 2017.

[27] S. Shen, B. Yang, K. Qian, Y. She, and W. Wang, "On improved dv-hop localization algorithm for accurate node localization in wireless sensor networks," *Chinese Journal of Electronics*, vol. 28, no. 3, pp. 658–666, 2019.

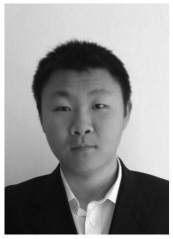
[28] J. Li, D. Wang, and Y. Wang, "Security dv-hop localisation algorithm against wormhole attack in wireless sensor network," *IET Wireless Sensor Systems*, vol. 8, no. 2, pp. 68–75, 2018.

[29] X. Zhou, W. Cai, R. Chen, L. Gui, F. Shu, J. Lin, S. Zhang, and Y. Zhang, "Secrecy energy efficiency optimization for miso swipt systems," *Physical Communication*, vol. 28, pp. 19 – 27, 2018.

[30] R. B. A. Lewis, "Lectures on modern convex optimization: Analysis, algorithms, and engineering applicationsby a. ben-tal; a. nemirovski," *Siam Review*, vol. 44, no. 4, pp. 731–734.

[31] A. Behnad, X. Wang, L. Hanzo, and T. J. Willink, "Connectivity-based centroid localization using distributed dense reference nodes," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6685–6689, July 2018.

Linqing Gui received the Ph.D. degree in information science from INSA-Toulouse, Toulouse, France, in 2013. He is now an Associate Professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. His main research interests are wireless communication and internet of things.



Yang Zhou received the B.S. degree from Nanjing University of Science and Technology, JiangSu, China, in 2014. He is currently pursuing the M.S. degree at the School of Electronic and Optical Engineering, Nanjing University of Science and Technology. His main research interests are wireless communication and internet of things.

Fu Xiao (M'12) received the Ph.D. degree in computer science and technology from the Nanjing University of Science and Technology, Nanjing, China, in 2007. He is currently a Professor and a Ph.D. Supervisor with the School of Computers, Nanjing University of Posts and Telecommunications. His main research interests are wireless sensor networks and mobile computing. He is a member of the IEEE Computer Society and the Association for Computing Machinery.

Feng Shu received the Ph.D., M.S., and B.S. degrees from the Southeast University, Nanjing, in 2002, XiDian University, Xi'an, China, in 1997, and Fuyang Normal University, Fuyang, China, in 1994, respectively. From Oct. 2003 to Oct. 2005, he is a post-doctor researcher with the National Key Mobile Communication Lab at the Southeast University. From Sept. 2009 to Sept. 2010, he is a visiting post-doctor at the University of Texas at Dallas. In October 2005, he joined the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China, where he is currently a Professor and supervisor of Ph.D and graduate students. He is also with Fujian Agriculture and Forestry University and awarded with Mingjian Scholar Chair Professor in Fujian Province. His research interests include wireless networks, wireless location, and array signal processing. He has published about 200 papers, of which more than 100 are in archival journals including more than 40 papers on IEEE Journals and 70 SCI-indexed papers. He holds six Chinese patents. He serves as an Editor for IEEE Access.

Thierry Val obtained his Ph.D. in computer science at Blaise Pascal University, Clermont-Ferrand, France, in 1993. In 1994, he became a lecturer at the University of Toulouse, where he currently teaches networks and computing systems. He obtained his HDR in 2002. He is now a professor for the University of Toulouse at the Blagnac Institute of Technology. He was sub-manager of the LATTIS laboratory, where he managed a research activity on wireless local networks and related protocols. He is now a member of IRIT-CNRS laboratory of Toulouse in RMESS team. His current research focuses on wireless networks in smart homes, DL-IoT (Device Layer-Internet of Things).