

DV-Hop 定位算法误差分析与优化^①

李敬兆¹, 孙睿², 谭大禹²

¹(安徽理工大学 电气与信息工程学院, 淮南 232001)

²(安徽理工大学 计算机科学与工程学院, 淮南 232001)

摘要: 传感器节点在森林、水下等复杂环境下进行数据采集时, 由于信号强度与信号传输速度受到障碍物或传输介质的干扰, 影响了基于信号信息的定位算法的测量精度. 同时考虑到节点的成本和节点的动态性, DV-Hop 定位算法有较强的适用性和实用性. 分析了影响 DV-Hop 算法定位精度的因素, 并在此基础上提出了一个度量节点离散程度的公式, 给出了一个该算法的优化方案, 仿真表明优化后的算法有更高的定位精度.

关键词: 无线传感器网络; 定位; DV-Hop

Error Analysis and Improvement of DV-Hop Location Algorithm

LI Jing-Zhao¹, SUN Rui², TAN Da-Yu²

¹(School of Electrical and Information Engineering, Anhui University of Science and Technology, Huainan 232001, China)

²(School of Computer Science and Engineering, Anhui University of Science and Technology, Huainan 232001, China)

Abstract: When the sensor nodes collect data in the forest or water environment, the signal intensity and the signal transmission speed are affected by the obstacles or transmission media, which affects the measurement accuracy of localization algorithm based on the signal information. Taking into account the cost of nodes and the dynamic nature of nodes, DV-Hop positioning algorithm has excellent applicability and practicality. This paper analyses the factors influencing the accuracy of DV-Hop algorithm, proposes a formula for measuring the degree of uniformity of nodes, optimized the algorithm. The simulation results show that the optimized algorithm has higher positioning accuracy.

Key words: wireless sensor network(wsn); positioning; DV-Hop

传感器节点采集数据时, 获取节点信息是一项必不可少的重要环节. 例如在环境监测的应用中, 需要获取采集节点的位置信息以便反映出该区域环境信息, 又如在目标追踪与监视的应用中, 其目的就是为了获取该目标的位置信息. 另一方面, 利用传感器节点的位置信息可以对路由进行优化, 根据节点传回的位置信息构造出整个传感器网络的网络拓扑, 从而加强对网络的管理. 因此, 研究传感器节点定位对传感器的部署和监控都有着重要意义.

随着无线传感器网络的发展, 关于定位的新算法和优化方案不断涌现. 目前已有许多算法能够解决无线传感器网络中的定位问题, 但每种算法都有各自的适用场景, 在定位算法的选择上, 必须对具体需求做

出权衡. 当传感器在一些复杂环境下工作时, 传播信号受到周围不确定因素影响, 会产生反射、散射、衍射等现象, 同时信号强度的衰减速度也随介质的改变而改变, 再加上障碍物的干扰, 这就使得依靠信号强度和信号传输速度的定位算法不能满足实际需求. DV-Hop 定位算法是一个基于非测距的经典算法, 有着实现简单, 相对误差较小等优点, 同时在节点成本与功耗方面, 相比于基于测距的定位算法有很大优势. 但由于该算法使用节点之间的平均跳距估算节点间的实际距离, 当节点分布不均匀时, 该算法的定位精度大大降低. 同时连通度较低的信标节点与三边定位误差也影响了定位精度. 本文对该算法进行改进, 通过仿真对算法有效性进行验证.

① 基金项目:国家自然科学基金(61170060);安徽省学术与技术带头人学术科研活动(2015D046);安徽省高等学校优秀拔尖人才资助项目(gxbjZD2016044)

收稿时间:2016-07-25;收到修改稿时间:2016-09-23 [doi:10.15888/j.cnki.csa.005738]

1 DV-Hop定位算法与三边定位

DV-Hop 算法是由 Dragos Niculescu 等人提出的, 其算法最大的特点就是无需依靠传感器信号信息, 而是依靠路由跳数实现定位. 由于路由信息可从节点网络层直接获取, 该算法无需测距模块从而降低了节点成本, 同时不依赖于传感器信号信息, 从而有较高的自适应性. 该算法通过获得信标节点这些先验信息和未知节点到信标节点的最小跳数, 估算出未知节点的位置. 具体算法分为以下三个步骤:

1) 计算各个节点之间的最小跳数

首先每个信标节点将自身的位置信息以广播的形式发送给周围的节点, 其广播的数据包中包含一个记录跳数的字段记为 h , 初始值为 0. 接收到该信息的节点只保留到相同信标节点的最小跳数值, 并将该数值加 1, 向周围的邻居节点继续广播. 通过节点不断的广播, 网络中的所有节点都将记录下该节点与每个信标之间所经过的最少节点个数, 即最小跳数. 未知节点 u 与信标节点 s 之间的最小跳数记为 $t_{u,s}$.

2) 估算平均跳距

通过上一步骤节点的广播, 每一个信标节点都可以获取其它信标节点的位置信息以及它们之间的最小跳数, 基于以上两个信息可估算出节点间进行通讯时每一跳所经过的实际距离, 记为 $HopSize$. 计算平均跳距的公式如式(1)所示.

$$HopSize = \frac{\sum_{i \neq j} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j} h_{i,j}} \quad (1)$$

其中 (x_i, x_j) , (y_i, y_j) 为信标节点 i, j 坐标, $h_{i,j}$ 为信标节点 i 与 j 之间的最小跳数.

3) 计算未知节点位置

设 $d_{u,s}$ 为估算的未知节点 u 到信标节点 s 的距离, 计算公式如式(2)所示.

$$d_{u,s} = HopSize \times t_{u,s} \quad (2)$$

当未知节点得到周围三个信标的距离信息后, 根据三边定位原理即可计算出未知节点的实际方位. 三边定位求解原理如图 1 所示.

设信标节点 A_1, A_2, A_3 的坐标分别为 (x_i, y_i) , 其中 $(i=1,2,3)$, 未知节点的坐标为 (x, y) , 未知节点与三个信标的距离分别为 $d_i (i=1,2,3)$, 那么存在以下关系, 如式(3)所示.

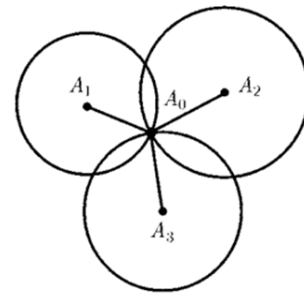


图 1 三边定位求解图

$$\begin{cases} \sqrt{(x-x_1)^2 + (y-y_1)^2} = d_1 \\ \sqrt{(x-x_2)^2 + (y-y_2)^2} = d_2 \\ \sqrt{(x-x_3)^2 + (y-y_3)^2} = d_3 \end{cases} \quad (3)$$

由上式可计算出未知节点的具体坐标, 从而实现对该节点的定位.

2 误差分析

2.1 节点分布的均匀程度对定位精度的影响

DV-Hop 算法由于依赖于跳数进行距离估算, 其对平均跳距的估算决定着定位的精度, 在一些节点分布均匀的情况下, DV-Hop 具有良好的定位精度. 但是在实际运用中, 节点可能呈非均匀分布, 即某个区域内节点分布密集, 另外的区域内节点分布稀疏. 分析算法计算公式可知, 得出的平均跳距 $HopSize$ 的值具有唯一性, 无法随着节点分布的疏密而改变. 在节点分布密集的环境下, 计算得到的平均每跳距离相比于节点稀疏的环境的平均每跳距离较大, 这是因为节点越密集, 它们之间的通信路径就越近似于一条直线, 如图 2 所示, 且计算得出的平均每跳距离近似于节点自身的通信半径(平均每跳距离一定不大于通信半径).

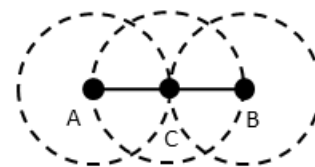


图 2 理想情况下的节点拓扑

节点 A 与节点 B 进行通信时, 针对该算法的最理想情况, 即存在节点 C , 使得节点 C 与节点 A 和 B 的距离恰好等于节点的通信半径. 该网络拓扑情况下使用该算法计算得出的平均跳距就为通信半径, 计算得

出节点 A 与 B 的距离也与实际相等. 在节点分布稀疏区域, 节点 A 与 B 的通信可能经过若干个节点, 这将造成计算得出的平均每跳距离小于通信半径, 所以全部节点都使用相同的平均跳距进行距离计算将影响定位精度.

2.2 连通度较低的信标节点对定位精度的影响

连通度较低的信标节点会造成定位误差, 如图 3 所示的网络拓扑环境.

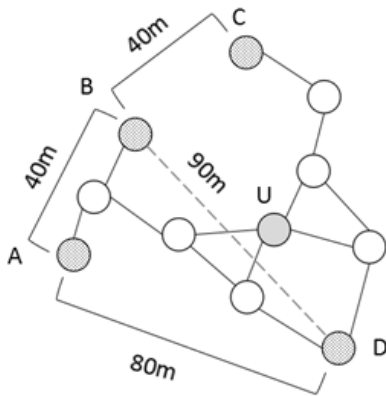


图 3 节点网络拓扑图

信标节点 A, B, C, D 与待定位节点 U, 其余为普通节点, 由于地形或节点运动原因, 节点 B 与节点 C 之间存在障碍物, 造成信标节点 C 处于整个网络边缘. 位于区域边界的节点 C 只能接收到来自某一侧的信息, 使得该节点无法利用全面的信息进行定位计算. 如果使用信标节点 A、B、C 计算得到的平均每跳距离约为 10m, 可观察到, 实际情况中 B 与 C 的距离只是不到两个通信距离, 而 B 与 C 之间的跳数达到了 6 跳, 使用该平均每跳距离计算 A 与 B 的距离为 20 m. 这与真实距离相差了一倍, 若使用该平均跳距计算未知节点位置必然会造成较大误差. 造成该影响的原因为信标节点 C 的连通度较低处于网络边缘, 信标节点 C 的通信半径下仅有 1 个节点. 如果排除 C 节点, 使用信标节点 A、B、D 来估算平均每跳距离则误差较小.

2.3 三边定位计算公式对定位精度的影响

根据三边定位原理可知, 节点定位时通过周围三个信标节点的位置信息来计算自身位置, 所以选择不同的信标计算得到的位置信息也不同. 一般情况下, 对这三个信标节点的选择采用就近原则, 即选择距离未知节点最近的三个信标进行位置的计算, 这样可以减少定位的误差. 但实际情况下, 由平均跳距计算得

出的节点间距离与实际距离存在偏差, 三边定位求解图中的三个圆的交汇处是一块区域, 而不是一个点, 如图 4 所示.

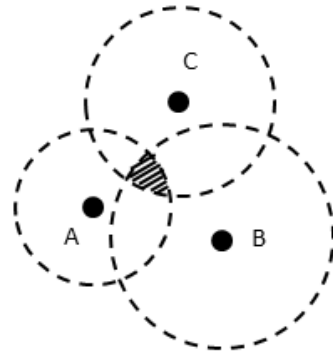


图 4 三边定位误差分析

已知待定位节点到信标 A, B, C 的距离, 分别以信标位置为圆心, 这三个距离为半径作圆, 形成图 4 中的阴影区域. 与之对应, 对三边定位方程组求解, 解的范围(即未知节点的坐标)便在该阴影内. 这就导致只能判断出未知节点在该区域中, 不能确定该节点的具体位置. 通过选择周围的信标节点参与计算, 采用最小二乘法求解可有效避免上述情况.

3 算法改进与仿真

在无线传感器网络中, 各个节点之间的距离与各个节点的平均距离之差反应着整个网络中所有节点的离散程度, 离散程度越小则该区域中节点越均匀, 反之该区域中节点越离散. 借鉴方差对无线传感器网络中的节点离散程度进行度量.

设无线传感器网络中节点 i 与节点 j 之间的距离为 $d_{i,j}$, 则一个边长为 w 的方形区域内节点离散程度 S 计算公式如式(4)所示.

$$S = \sqrt{\frac{\sum_{i \neq j} (d_{i,j} - \bar{d})^2}{C_n^2}} \tag{4}$$

式中 \bar{d} 表示各个节点之间的平均距离, n 表示无线传感器网络中的节点个数. \bar{d} 的计算公式如式(5)所示.

$$\bar{d} = \frac{\sum_{i \neq j} d_{i,j}}{C_n} \tag{5}$$

使用 Matlab 对无线传感器网络中节点进行仿真, 构造节点分布不均匀环境, 节点分布如图 5 所示.

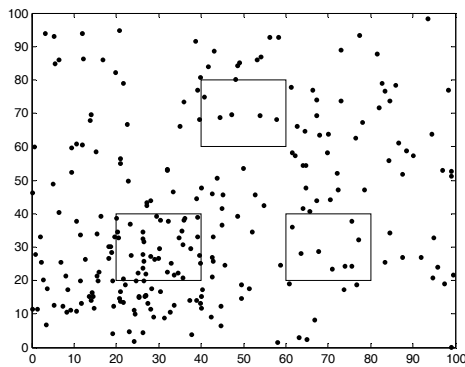


图 5 节点分布图

对整个区域内和三个方形区域分别计算其离散程度, 通过 50 次仿真节点不均匀分布的情况, 得出各区域节点的离散程度如图 6 所示.

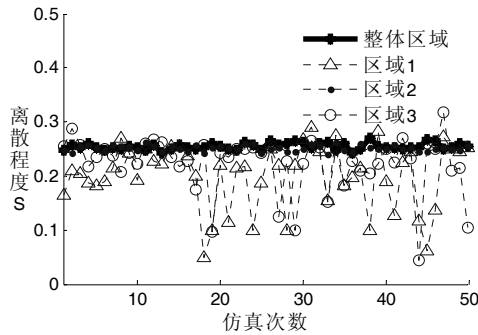


图 6 各区域节点离散程度

由以上仿真数据可看出, 三个方形区域内节点的离散程度大体上比全体节点的离散程度低很多, 通过实验数据汇总计算多次仿真各区域离散程度的平均值得到的结果如表 1 所示.

表 1 各区域节点离散程度表

区域	整体	区域 1	区域 2	区域 3
离散值	0.25	0.17	0.23	0.20

从表中数据可观察到, 由于节点分布不均整体区域的离散程度值最大, 取该整体部分区域计算得到的平均离散程度值均小于整体区域的平均离散程度值, 所以节点离散程度大的区域可通过区域划分的方式划分成多个节点分布相对均匀的区域.

因此, 在节点广播阶段, 通过控制数据包转发次数, 进行局部定位将提高定位精度. 仅选取待定位节点周围的节点进行计算, 忽略掉远离待定位节点的节

点, 因为待定位节点周围的节点环境可以近似成均匀分布的节点环境, 同时节点的泛洪不会扩散至整个传感器网络, 节约了功耗和网络负担.

针对连通度低的信标情况, 在进行平均每跳距离计算时, 应尽量避免连通度较低的信标节点, 信标节点可首先通过广播确定通信范围内的邻居节点个数, 如果该个数极少则自身休眠不参与平均跳距地计算任务.

针对三边定位误差问题, 选用若干个邻居信标对节点进行定位计算. 这若干信标坐标代入后组成的方程组使用最小二乘法进行求解. 设待定位节点坐标为 (x, y) , 周围有 k 个邻居信标, 坐标表示为 (x_i, y_i) 未知节点到信标的距离为 d_i , 其中 $i=1, 2, 3, \dots, k$. 代入两点距离公式后组成如式(6)所示的超定方程组.

$$\begin{aligned}
 d_1 &= \sqrt{(x-x_1)^2 + (y-y_1)^2} \\
 d_2 &= \sqrt{(x-x_2)^2 + (y-y_2)^2} \\
 &\dots \\
 d_k &= \sqrt{(x-x_k)^2 + (y-y_k)^2}
 \end{aligned}
 \tag{6}$$

化简移项后如式(7)所示.

$$\begin{aligned}
 x^2 + y^2 - 2x_1x - 2y_1y &= d_1^2 - x_1^2 - y_1^2 \\
 x^2 + y^2 - 2x_2x - 2y_2y &= d_2^2 - x_2^2 - y_2^2 \\
 &\dots \\
 x^2 + y^2 - 2x_kx - 2y_ky &= d_k^2 - x_k^2 - y_k^2
 \end{aligned}
 \tag{7}$$

取

$$A = \begin{bmatrix} 1, -2x_1, -2y_1 \\ 1, -2x_2, -2y_2 \\ \dots \\ 1, -2x_k, -2y_k \end{bmatrix}, \quad b = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 \\ d_2^2 - x_2^2 - y_2^2 \\ \dots \\ d_k^2 - x_k^2 - y_k^2 \end{bmatrix},$$

$$X = [x^2 + y^2, x, y]^T$$

方程组(7)可化简成 $AX = b$ 的形式, 利用最小二乘法可得 $X = (A^T A)^{-1} A^T b$, 从而解出未知节点坐标.

同时未知节点与某个信标节点距离越近, 该信标节点越能反映出未知节点周围的网络拓扑情况, 使用该信标节点计算的平均跳距越接近于实际情况下未知节点周围节点的跳距.

基于以上分析, 其优化后的算法如下所示:

1) 待定位节点广播自身信息.

待定位节点进行广播, 在原有的广播的信息中增加一个跳数上限 T 的字段, 防止该消息扩散至整个网络. 这样待定位节点周围的信标节点都存有到该待定

节点的最小跳数。

2) 确定信标。

接收到定位信息的信标节点，通过广播自身信息(报文跳数限制为 1 跳)获取自身周围节点个数，并与自身设定的阈值 K 进行比较，大于阈值则参与计算平均跳距，否则不参与计算。

3) 估算未知节点广播范围内节点的平均每跳距离。

4) 计算待定位节点到各个信标的距离，使用最小二乘法计算待定位节点坐标。

优化后的算法流程图如图 7 所示。

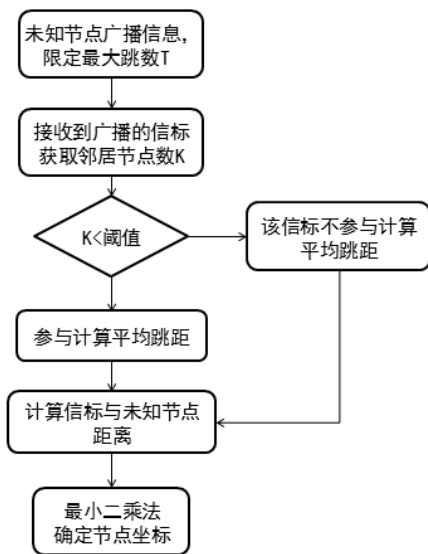


图 7 算法流程图

为验证优化后算法的有效性，使用 Matlab 对该算法进行仿真实验，200 个传感器节点非均匀分布在 100×100 的矩形区域内，其中信标节点占总节点数的 30%，节点的通信半径设置为 15，低连通度信标的阈值判定设置为 1，对单个节点进行定位计算误差后分析得出优化后算法的定位精度受广播最大跳数 T 的变化而产生波动，如图 8 所示。

由图中数据可观察到相对误差随着最大跳数 T 的增大呈 V 字型走势，这是因为当 T 取较小的值时未知节点发送的数据包无法到达信标或者参与定位的节点过于稀少导致误差较大，随着可传输跳数的增加，参与定位的节点变多，误差降至最低，当最大跳数继续增大时，由于节点的非均匀分布性导致了定位误差增大，在该环境下最大跳数设为 3 时，定位误差最小，此时为最优最大跳数。

对该网络拓扑环境中的所有节点进行定位仿真，通过调整最大跳数与低连通度信标的阈值后达到最优的定位效果，如图 9 所示。

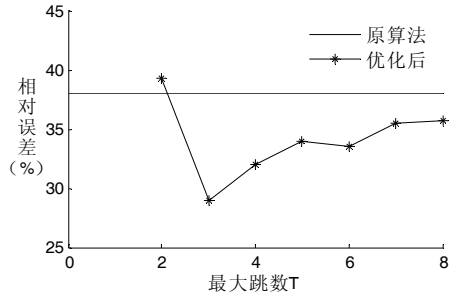


图 8 最大跳数与定位误差关系图

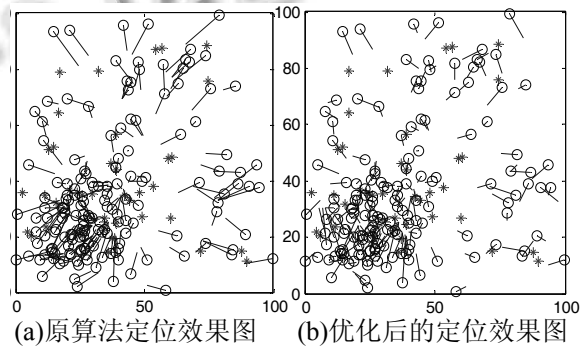


图 9 定位效果图

图 9(a)为原算法定位效果图，图 9(b)为优化后的定位效果图。图中圆标为未知节点的实际位置坐标，星标为信标节点的实际位置坐标，连接圆标线段的另一端为算法定位位置坐标，则每条线段长度代表了每个未知节点定位的误差，线段越长定位误差越大。图 9(b)线段明显短于图 9(a)线段，表明优化后的算法定位精度比原算法高。

针对非均匀分布的节点定位优化，一些学者提出采用未知节点到各个信标的跳数对平均跳距进行加权处理的方法提高定位精度，即与未知节点相距跳数越小的信标在计算时赋予更大的权值，该方法在实际应用中也获得了良好的定位效果，本文对其相关算法进行仿真后得出如下数据。

图 10 为三种算法在节点非均匀分布环境下的定位误差与信标所占比例的关系图，随着信标节点的增加，所有算法的定位误差均有下降趋势，优化后的两个算法较原算法定位精度提高较为明显，以信标所占比例 18%为例，原算法相对误差为 45.8%，加权 DV-Hop 算法为 38.7%，本文算法为 35.1%。

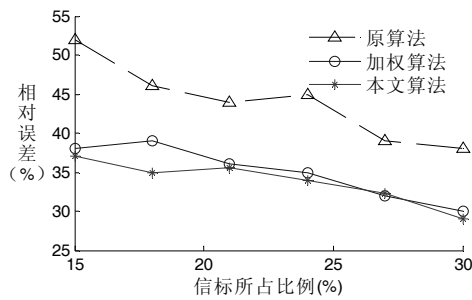


图10 信标比例与相对误差关系图

本文算法主要针对节点离散程度影响因素进行改进,通过改变节点离散程度,即调整节点分布情况,使用上述三种算法对节点进行定位,计算得出节点定位的相对误差如表2所示。

表2 不同离散程度下各定位算法的定位误差(%)表

算法	离散值 S			
	0.25	0.3	0.35	0.4
原算法	43.8	48.7	56.4	66.1
加权算法	36.6	39.3	43.5	46.5
本文算法	35.4	39.1	41.9	45.2

表中数据反映出各算法的定位误差都随节点离散程度的增加而增加,加权算法和本文算法在节点离散程度增大的情况下均能抑制定位误差的类指数式增长,本文算法的定位误差在不同离散值下均小于其他两种算法的定位误差。

由于本文算法在定位中采取区域划分的原则,数据包只能扩散至节点周边区域,相比于传统算法洪泛整个网络的做法,大大降低了节点的能耗和系统的通信开销。

4 总结

本文首先介绍了 DV-Hop 定位算法,分析了该算

法在特定环境下影响定位误差的因素,并针对实际应用环境,将原有的算法进行了改进,提高了定位精度的同时也节约了网络资源与系统开销。算法分析得到的数据和仿真实验数据表明,该算法在节点密集程度不同的环境下可保证较高的定位精度,同时优化后的算法可根据实际应用环境调整最大跳数及低连通信标阈值达到最佳的定位效果,具备可用性及适用性。

参考文献

- 1 孙利民,李建中,陈渝.无线传感器网络.北京:清华大学出版社,2005.
- 2 黄俊杰,刘士兴,干小宇,尹坤.基于节点密度的改进型 DV-Hop 算法.电子科技,2015,8:67-70.
- 3 王书聪.无线传感器网络分布式定位算法研究.计算机技术与发展,2008,11:62-65.
- 4 赵灵锴,洪志全.基于无线传感器网络的 DV-Hop 定位算法的改进.计算机应用,2011,5:1189-1192.
- 5 张晓龙,解慧英,赵小建.无线传感器网络中一种改进的 DV-Hop 定位算法.计算机应用,2007,11:2672-2674.
- 6 余磊,余成波,祝松健,等.基于跳距修正加权 DV-Hop 的 WSN 定位算法.压电与声光,2013,6:899-902.
- 7 Doremami F, Javadi HHS, Farahi A. A new distributed weighted multidimensional scaling algorithm for localization in wireless sensor networks. International Journal of Computer Science & Engineering Survey, 2011, 2(1): 39-64.
- 8 Hu Y, Li X. An improvement of DV-Hop localization algorithm for wireless sensor networks. Telecommunication Systems, 2013, 53(1): 13-18.
- 9 Li S, Ding X, Yang T. Analysis of five typical localization algorithms for wireless sensor networks. Wireless Sensor Network, 2015, 7(4): 27-33.