

基于角度阈值的改进型 DV-Hop 定位算法

杨磊, 张政保, 谢桂海, 明亮, 王新锋

(军械工程学院计算机工程系, 石家庄 050003)

摘要: 针对 DV-Hop 定位算法中部分参与定位的锚节点位置近似成直线而引起的定位性能较低的问题, 提出一种基于角度阈值的锚节点选择策略, 通过设置角度阈值来筛选锚节点以提高 DV-Hop 的定位性能。仿真结果证明, ATDV-Hop 算法在可定位节点比例上比 DV-Hop 算法提高了 24.05%, 在定位精度上降低了 8.79%。

关键词: 普适计算; 无线传感器网络; 定位; DV-Hop 算法; ATDV-Hop 算法

Improved DV-Hop Positioning Algorithm Based on Angle Threshold

YANG Lei, ZHANG Zheng-bao, XIE Gui-hai, MING Liang, WANG Xin-feng

(Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050003)

Abstract In order that the coordinate of some anchor nodes which join in positioning may be close to a line, the position capability of DV-Hop is lower. This paper puts forward a positioning algorithm based on angle threshold, which can set the angle threshold to filter out the anchor nodes and improve DV-Hop's positioning capability. Compared with DV-Hop, simulation shows that the improvement of ATDV-Hop on the proportion of enable-positioning nodes is 24.05% and the positioning precision is 8.79%.

Key words ubiquitous computing; wireless sensor network; positioning; DV-Hop algorithm; ATDV-Hop algorithm

无线传感器网络(WSN)^[1]是普适计算^[2]思想的一种典型实验环境, WSN的定位技术为普适计算提供了重要的位置上下文信息, 已成为普适计算领域的研究热点。WSN的定位技术主要有基于测距技术(Range-Based)和无须测距技术(Range-Free)2种^[3]。其中, Range-Based需要测量节点之间的实际距离或角度关系, 而Range-Free技术无须测量距离和角度信息, 仅根据网络连通性等信息即可实现。Range-Based虽然在定位精度上有可取之处, 但并不适用于低功耗、低成本的应用领域。当定位误差小于传感器节点无线通信半径的40%时, 定位误差对路由性能和目标追踪精确度的影响不会很大^[4], 因此, Range-Free定位方案倍受关注。比较典型的Range-Free定位算法有DV-Hop^[5-6]、凸规划、MDS-MAP、Amorphous、APIT和质心定位算法等^[3-4]。本文把需要定位的节点称为未知节点, 已知位置、协助未知节点定位的节点称为锚节点(anchor node)。某节点通信半径内的所有其他节点, 称为该节点的邻节点。

1 DV-Hop 定位算法

1.1 DV-Hop 的定位过程

DV-Hop 定位算法是美国路特葛斯大学的 Dragos Niculescu等人利用距离矢量路由(distance vector routing)和GPS定位原理提出的6种分布式定位算法(APS)^[5-6]之一。该算法的核心思想是: 第1阶段先利用典型的距离矢量交换协议, 使网络中所有节点获得距锚节点的跳数, 每个节点维持一个表 $\{X_i, Y_i, h_i\}$, 并且只与它的邻节点交换更新信息, 保证网络中所有节点记录下的是到锚节点的最小跳数 h_i 。然后, 其中一个锚节点计算与其他锚节点的距离, 由此估计出平均每跳的距离, 并把它作为校正值(correction)广播给其他邻节点。当节点收到这个校正值之后, 丢弃后来得到的校正值, 这样就保证大部分节点都是从离它最近的锚节点获得校正

值, 利用这个校正值节点就可以估计出它与锚节点的距离了。最后可以使用三边定位算法估计出未知节点的位置, 三边定位算法描述见1.2节。

第2阶段中校正值(平均每跳的距离) c_i 由式(1)得出, 其中 (X_i, Y_i) 为锚节点 i 的坐标, h_j 为锚节点 i 与 $j(j \neq i)$ 之间的跳数。

$$c_i = \frac{\sum_{j \neq i} \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}}{\sum_{j \neq i} h_j} \quad (1)$$

经过第1阶段和第2阶段, 能够计算出锚节点 N_1, N_2, N_3 之间的距离和跳数, 如图1所示, 3个锚节点 $N_1 \sim N_2, N_2 \sim N_3, N_3 \sim N_1$ 之间的距离分别为40 m, 75 m和100 m, 对于未知节点 A , 由于 N_2 离它的跳数最小, 因此 A 应该从 N_2 获取校正值 c_2 。由式(1)得 $c_2 = (40+75)/(2+5) = 16.42$ m, 那么3个锚节点与 A 之间的距离分别为 $N_1: 3 \times 16.42$ m, $N_2: 2 \times 16.42$ m, $N_3: 3 \times 16.42$ m。最后利用三边测量法估计出 A 的位置。

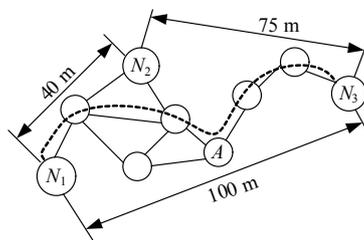


图1 DV-Hop 定位算法的一个实例

基金项目: 国家自然科学基金资助项目(60372042)

作者简介: 杨磊(1982-), 男, 硕士研究生, 主研方向: 智能检测技术, 普适计算; 张政保, 教授、博士; 谢桂海, 教授、博士生导师; 明亮、王新锋, 博士研究生

收稿日期: 2007-11-20 **E-mail:** yang1982lei@sina.com

1.2 三边测量定位算法

三边测量法^[4]只需 3 个锚节点就可以估计出未知节点的位置, 可以大大减少节点之间的计算和通信功耗, 因此, 在实际应用中倍受关注。设 3 个锚节点 A, B, C 的坐标分别为 $(x_a, y_a), (x_b, y_b), (x_c, y_c)$ 以及它们到未知节点 D 的距离分别是 d_a, d_b, d_c , 设未知节点 D 的坐标为 (x, y) , 那么有公式:

$$\begin{cases} \sqrt{(x-x_a)^2+(y-y_a)^2}=d_a \\ \sqrt{(x-x_b)^2+(y-y_b)^2}=d_b \\ \sqrt{(x-x_c)^2+(y-y_c)^2}=d_c \end{cases} \quad (2)$$

由式(2)可以计算出未知节点 D 的坐标为

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2(x_a-x_c) & 2(y_a-y_c) \\ 2(x_b-x_c) & 2(y_b-y_c) \end{bmatrix}^{-1} \begin{bmatrix} x_a^2-x_c^2+y_a^2-y_c^2+d_c^2-d_a^2 \\ x_b^2-x_c^2+y_b^2-y_c^2+d_c^2-d_b^2 \end{bmatrix} \quad (3)$$

2 角度阈值的锚节点选择及 ATDV-Hop 定位算法

2.1 影响 DV-Hop 算法定位性能的因素

由于 WSN 节点是随机抛洒出去的^[1], 因此用于定位的锚节点位置是随机的。由三边定位算法可知, 3 个点共线时, 式(2)中的 3 个方程可以约去 1 个, 那么式(3)就不成立了, 所以式(2)得出的 (x, y) 值就会有 2 个结果。同时由于 DV-Hop 算法中式(1)得出的 c_i 值本身就有一定误差, 因此当用于定位的 3 个锚节点近似成直线时, 会给定位结果带来很大误差。如图 2 所示, 由于锚节点 A, B, C 近似成直线, 对于未知节点 D 的估计就出现了严重错误。同时笔者的仿真实验显示当锚节点较密集时约有 30% 的未知节点在定位时选择的锚节点类似于图 2 所示情形。若选择离未知节点较远的锚节点 E 进行定位, 由于锚节点 B, C, E 的位置相对较分散, 因此对节点 D 的定位效果较好, 基于此笔者提出了基于角度阈值的锚节点选择策略, 用它来选择位置关系较好的锚节点组用以改善定位效果。

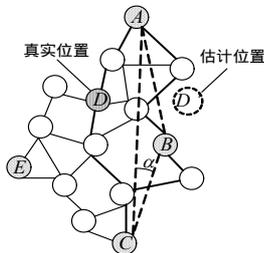


图 2 锚节点位置关系对估计结果的影响示意图

2.2 角度阈值选择策略的理论依据

由于锚节点本身的位置关系严重影响定位结果, 因此在某些情况下必须放弃某些跳数较小的锚节点, 转而去寻找跳数稍大、离未知节点较远的锚节点重新定位。放弃哪些锚节点进而去选择哪些锚节点是角度阈值选择策略的出发点。从图 2 可以看出, 锚节点 A, B, C 近似成直线, 表现为 $\triangle ABC$ 的某些内角过大或过小。在图 2 中, 由于 $\angle ABC$ 接近于 180° , 因此给定位结果带来了很大的影响。

由于锚节点的坐标是已知的, 根据余弦定理可得到三角形内角的余弦值。对于图 2 中 $\triangle ABC$ 的任意一个内角 α , 根据余弦函数的性质可以得出 $|\cos \alpha|$ 越接近于 1, $\triangle ABC$ 的 3 个点越接近于重合。因此, 本文在此设置一个角度阈值 $T(0 < T < 1)$, 当锚节点所构成的 $\triangle ABC$ 的 3 个内角的余弦绝对值都不大于阈值 T 时, 说明 3 个锚节点符合定位要求, 可以进行定位。如果其中有 1 个内角的余弦绝对值大于阈值 T 时, 说明 3 个锚节点接近共线而不能定位, 须重新选择锚节点。在重新选择锚节点时, 仍然是从与未知节点跳数较小的锚节

点开始选取, 当选取的锚节点与未知节点之间的跳数差的绝对值较大时(本文实验中选择 3 跳)应该放弃一个离未知节点跳数最小的锚节点, 进而选择比它大 1 跳的锚节点, 以保证 $\triangle ABC$ 的内角不至于过大或过小。

2.3 ATDV-Hop 定位算法的实现步骤

基于角度阈值的锚节点选择策略对 DV-Hop 进行了改进, 把改进后的算法称为 ATDV-Hop(Angle Threshold DV-Hop) 定位算法, 其实现步骤如下:

Step1 计算未知节点与锚节点之间的最小跳数, 这一步与 DV-Hop 算法的第 1 步相同^[5-6];

Step2 设置角度阈值 $T(0 < T < 1)$, 注意 T 应该靠近 1;

Step3 从锚节点中选取与未知节点跳数最小的 3 个节点 N_1, N_2, N_3 构成定位 $\triangle ABC$;

Step4 根据余弦定理计算 $\triangle ABC$ 的 3 个内角的余弦绝对值;

Step5 比较 $\triangle ABC$ 的 3 个内角的余弦绝对值与阈值 T 的大小, 只要有 1 个角度的余弦绝对值大于阈值 T 转到 Step 6, 若 3 个角的余弦绝对值都小于阈值 T 则转到 Step 7;

Step6 重新选择锚节点 N_1, N_2, N_3 , 并回到 Step 4;

Step7 根据三边测量定位算法估计未知节点的位置坐标。

3 仿真实验

选择阈值 $T=0.8660$, 即定位三角形内角在 $\pi/6 \sim 5\pi/6$ 之间, 节点无线射程为 $R=15\text{ m}$, 在 $100\text{ m} \times 100\text{ m}$ 的平面区域内随机分布 200~500 个节点进行仿真实验, 对 DV-Hop 与 ATDV-Hop 算法进行了比较。

表 1 列出了部分节点的定位结果。从中可以看出 ATDV-Hop 算法比 DV-Hop 算法在定位性能上有了明显改善, 尤其是对于 DV-Hop 定位结果误差较大的节点(如节点 90)更是有了较大改善。

表 1 2 种算法定位结果对比表(部分节点)

节点序号	$X_{\text{实际}}$	$Y_{\text{实际}}$	$X_{\text{估计(DV-Hop)}}$	$Y_{\text{估计(DV-Hop)}}$	$X_{\text{估计(ATDV-Hop)}}$	$Y_{\text{估计(ATDV-Hop)}}$
41	94	30	93.000 0	27.000 0	91.006 0	30.762 1
81	72	28	57.000 0	23.000 0	70.500 0	27.500 0
90	38	70	-933.834 9	-67.274 1	38.520 6	72.953 5
109	74	28	57.000 0	23.000 0	70.500 0	27.500 0

如图 3 所示, ATDV-Hop 算法和 DV-Hop 算法随着总节点数量的增多, 可定位未知节点比例也随之增多, 在节点密集型网络中 ATDV-Hop 的可定位节点比例可达到 95%。仿真结果显示在可定位节点比例上 ATDV-Hop 算法比 DV-Hop 算法提高了 24.05%。

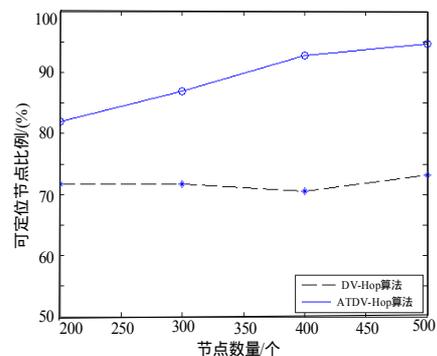


图 3 可定位节点比例关系分析图

(下转第 100 页)