

# 无线传感器网络分布式定位算法研究

王书聪

(南京邮电大学 计算机学院, 江苏 南京 210003)

**摘 要:**定位技术是传感器网络中的关键技术之一。重点介绍 DV-Hop, ROCRSSI 两种分布式非测距定位算法及具体实现过程, DV-Hop 算法以增加网络通信开销为代价获得了较高的容错性和覆盖率; ROCRSSI 算法简单、部署成本低, 但对周围环境和锚节点密度依赖性较大。从开销、容错性、定位精度、覆盖率等方面对两种算法的具体性能做了评价比较。

**关键词:**无线传感器网络; 节点定位; DV-Hop; ROCRSSI

**中图分类号:** TP301.6

**文献标识码:** A

**文章编号:** 1673-629X(2008)11-0062-04

## Research on Distributed Localization Algorithm in Wireless Sensor Network

WANG Shu-cong

(College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

**Abstract:** Node localization is one of the most critical technologies in wireless sensor networks. Mainly introduced the concrete processes of DV-Hop and ROCRSSI localization algorithms. DV-Hop algorithm had higher coverage and fault-tolerant, while increased the communication overhead; ROCRSSI algorithm was simple and had lower cost, but relied on environment and the density of the anchor nodes. Finally evaluated the performances of the two algorithms from overhead, accuracy, coverage, fault-tolerant and so on.

**Key words:** wireless sensor network; node localization; DV-Hop; ROCRSSI

### 0 引 言

无线传感器网络集中了微机电技术、嵌入式计算技术、分布式信息处理技术和无线通信技术,孕育出一种全新的信息获取和处理模式。在无线传感器网络的具体应用中,节点通常被随机部署在不同的环境中执行各种检测任务,以自组织的方式相互协调工作,节点的位置信息对传感器网络的检测活动至关重要<sup>[1]</sup>。节点定位是无线传感器网络进行目标识别、监控、跟踪等众多应用的前提,也是无线传感器网络研究中的热点问题之一。由于传感器节点本身存在资源有限、随机部署、通信易受环境干扰甚至失效等特点,传统的 GPS 定位<sup>[2]</sup>也受到节点成本、体积、功耗等因素的限制,节点定位技术已经成为传感器网络的一个重要的研究方向。到目前为止,已经提出了许多针对传感器网络节点自身的定位算法。重点介绍 DV-Hop, ROCRSSI 两种分布式非测距定位算法,并对两种算法的性能做了

具体评价。

### 1 DV-Hop 定位算法

#### 1.1 三边测量定位

其主要原理是:当未知节点获得三个或者是三个以上锚节点的距离时,执行三边测量定位。如图 1 所示,  $M$  为未知节点,  $A, B, C$  为锚节点。假设节点  $M$  获得与锚节点  $A, B, C$  的距离分别是  $d_1, d_2, d_3$ , 则在测距精确的情况下,采用三边定位理论可以判定节点  $M$  一定在以点  $A, B, C$  为圆心, 以距离  $d_1, d_2, d_3$  为半径的三个圆的交点处, 从而确定节点  $M$  的坐标。

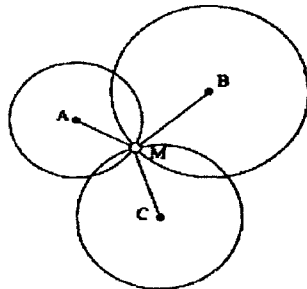


图 1 三边测量定位示意图

假设在二维环境中,  $M$  坐标为  $(x, y)$ ,  $A, B, C$  的

收稿日期: 2008-02-28

基金项目: 江苏省高校自然科学基金资助项目(05KJD520144); 江苏省“六大人才高峰”基金资助项目(06-E-044)

作者简介: 王书聪(1983-), 女, 河南周口人, 硕士研究生, 研究方向为无线传感器网络、信息安全。

坐标分别为  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ , 则

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

则解得节点  $M(x, y)$  的坐标为:

$$\begin{bmatrix} x \\ y \end{bmatrix} = 2x \begin{bmatrix} x_1 - x_3 & y_1 - y_3 \\ x_2 - x_3 & y_2 - y_3 \end{bmatrix}^{-1} \cdot \begin{bmatrix} x_1^2 - x_3^2 + y_1^2 - y_3^2 + d_1^2 - d_3^2 \\ x_2^2 - x_3^2 + y_2^2 - y_3^2 + d_2^2 - d_3^2 \end{bmatrix}$$

把三边测量定位推广到多边测量的情况如下: 设未知节点坐标为  $A(x, y)$ , 锚节点坐标为  $L_1(x_1, y_1), \dots, L_k(x_k, y_k)$ , 未知节点到锚节点的距离分别为  $d_1, \dots, d_k$ , 则可建立线性方程组, 表示形式  $Cx = b$ 。

$$\text{其中 } C = 2x \begin{bmatrix} x_1 - x_k & y_1 - y_k \\ \vdots & \vdots \\ x_{k-1} - x_k & y_{k-1} - y_k \end{bmatrix}, x = \begin{pmatrix} x \\ y \end{pmatrix},$$

$$b = \begin{pmatrix} d_1^2 - d_k^2 - x_1^2 + x_k^2 - y_1^2 + y_k^2 \\ d_{k-1}^2 - d_k^2 - x_{k-1}^2 + x_k^2 - y_{k-1}^2 + y_k^2 \end{pmatrix}$$

建立线性方程组后通过最小二乘法求解未知节点的位置估计:  $\hat{x} = (C^T C)^{-1} C^T b$ 。

## 1.2 DV-Hop 算法

DV-Hop 算法<sup>[3]</sup>是美国路特葛斯大学的 Niculescu 等人首次为了避免对节点间距离的直接测量而提出的。该定位算法采用了传感器网络节点能量及通信距离有限, 只能够与邻节点交换信息, 在网络中以多跳的方式传递数据来进行通信的特性。

其基本思想是: 将未知节点到参考节点之间的距离用网络平均每跳距离和两者之间最短路径跳数乘积表示, 进而使用三边测量法确定节点位置。DV-Hop 定位算法由三个阶段组成:

(1) 首先利用典型的距离矢量交换协议, 使网络内的所有节点获得到距离锚节点的跳数。

为了获得节点间的跳数, 锚节点向所有的邻居节点广播一个包含其位置信息的数据包, 接收到该数据包的节点将跳数加 1 后继续转发, 这样每一个节点即可获得到锚节点的最小跳数。

(2) 其次, 获得到其他锚节点的位置和隔跳距离后, 用锚节点计算网络平均每跳距离。

平均每跳距离  $C_i$  可以采用下式计算。其中  $(x_i, y_i)$  为附近锚节点的坐标,  $h_{i,j}$  为本节点到附近锚

节点的跳数。

$$C_i = \frac{\sum \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum h_{ij}} \quad (i \neq j, \forall j)$$

采用可控泛洪法把这个平均距离作为一个校验值广播至整个网络, 使节点仅仅接收第一个校验值而丢弃后来的, 从而保证绝大多数节点从最近的锚节点接收校验值。这样节点根据跳数和校验值计算到锚节点的距离。在大规模传感器网络中也可以为数据包设置一个 TTL 门限域来减少通信量。

(3) 最后, 当未知节点获得到 3 个或者是 3 个以上不共线锚节点的距离时, 利用三边测量法对未知节点实施定位。

例如: 如图 2 所示, 已知锚节点  $L_1$  与  $L_2, L_3$  之间的距离和跳数。 $L_2$  计算得到校正值 (即平均每跳距离)  $(40 + 75)/(2 + 5) = 16.42$ 。在本例中, 假设  $A$  从  $L_2$  获得校正值, 则它与 3 个锚节点之间的距离分别为  $L_1: 3 \times 16.42, L_2: 2 \times 16.42, L_3: 3 \times 16.42$ , 然后使用三边测量定位法确定节点  $A$  的位置。

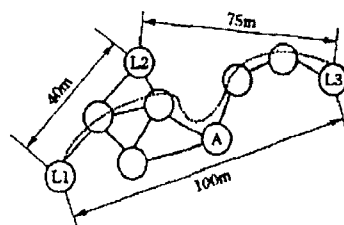


图 2 DV-Hop 定位算法示例图

综上所述, 该算法包含锚节点和未知节点两个定位模块, 各自具体的定位流程分别如图 3、图 4 所示。

可以从设置门限跳数  $L$ 、改善网络拓扑结构、修正地形影响、不良节点的处理等方面对算法进行改进来提高定位精度。

DV-Hop 算法引入跳数的概念, 用跳数来间接估算节点距离, 避免了对节点间距离的直接测量, 在网络拓扑较为规则时仍是一种较好的节点定位算法。

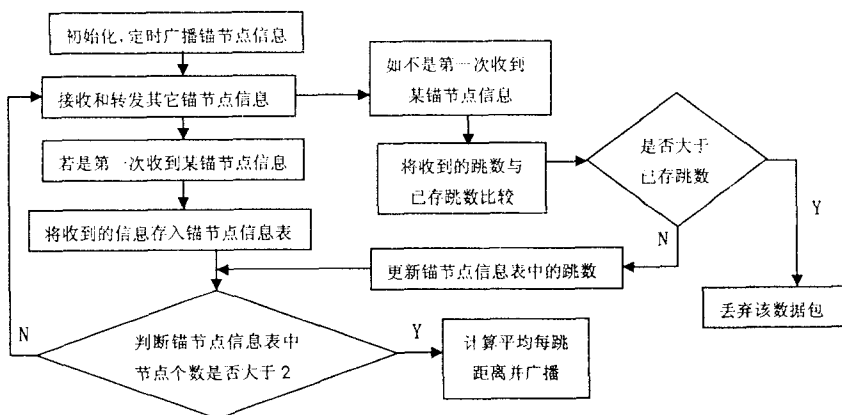


图 3 锚节点模块流程图

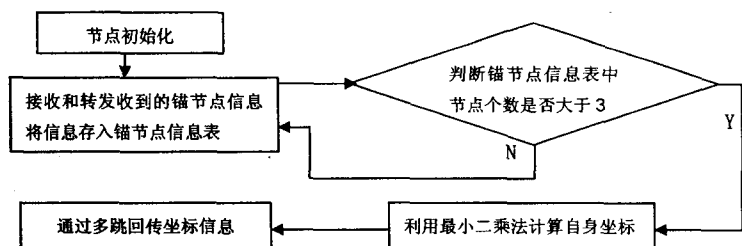


图4 未知节点模块流程图

## 2 ROCRSSI 算法

ROCRSSI(Ring Overlapping based on Comparison of Received Signal Strength Indicator)节点定位算法<sup>[4]</sup>属于分布式非测距定位算法。该算法的可行性基于以下假设：在方向的某个范围内，随着发送者和接收者之间距离的增加，接收者接收到的信号强度单调递减。

算法的基本思想是：网络中每个未知节点比较接收到的RSSI大小求出某些环形区域，进一步求出这些环形区域的相交部分并以其质心作为自身坐标。例如：图5所示，假设锚节点A发送信标信号后，如果未知节点S，锚节点B、C接收该信标信号的接收信号强度指示分别为 $RSSI_{AS}$ 、 $RSSI_{AB}$ 、 $RSSI_{AC}$ ，如果有： $RSSI_{AB} > RSSI_{AS} > RSSI_{AC}$ ，则 $d_{AC} > d_{AS} > d_{AB}$ ，因此S可知它必在以A为中心，以 $d_{AB}$ 为内半径并以 $d_{AC}$ 为外半径的环形区域内。类似地，S同样可以求出分别以B、C为中心的环形区域。求出这些环形区域的相交部分，并以相交区域的质心作为S自身坐标，如图6所示。

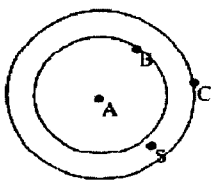


图5 ROCRSSI 算法示例图

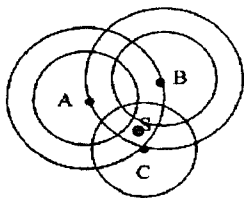


图6 ROCRSSI 算法示例图

ROCRSSI算法可分为两个阶段：RSSI传播阶段和位置估计阶段。

(1) RSSI传播阶段：首先，每个锚节点广播固定数量的信标信息。在这个阶段中，它的每个邻居节点持续地对接收信号强度采样。然后，所有邻居节点计算接收到的该锚节点信号强度的平均值。若邻居节点为

未知节点，则它只需保存该平均值；若邻居节点为锚节点，则它需要广播一条RSSI消息，包含自身ID、坐标、参考锚节点ID及接收信号强度的平均值。任何接收到该RSSI消息的未知节点记录该消息。

(2) 位置估计阶段：当未知节点经过RSSI传播阶段接收到足够信息后，它就可以利用这些信息求出它所在的每一个环形区域。计算这些环形区域的交集，并以相交区域的质心作为自身坐标。

利用ROCRSSI算法，未知节点S的定位过程可描述如下，其中： $S_s$ 表示未知节点S的邻居锚节点集合； $S_A$ 表示锚节点A的邻居锚节点集合； $R$ 表示包含未知节点S的环形区域的集合； $RSSI_{AB}$ 表示B接收到锚节点A的信标信号时的接收信号强度指示。

1) 令 $R = \{\}$ ;

2) 若 $S_s$ 非空，取出下一个元素A，并将其分为两部分 $S_{A1}$ 和 $S_{A2}$ 。其中 $S_{A1}$ 中的每个元素k满足： $RSSI_{Ak} > RSSI_{AS}$ ， $S_{A2}$ 中的每个元素h满足： $RSSI_{Ah} < RSSI_{AS}$ ，否则转4；

3) 若 $S_A$ 为空或 $S_{A2}$ 为空，转2；否则：

若 $S_{A2}$ 中的元素j具有最大 $RSSI_{Aj}$ ，则求出A与j之间的距离 $d_2$ ；

若 $S_{A1}$ 为空，令 $d_1 = d_2$ ；否则若 $S_{A1}$ 中的元素i具有最小 $RSSI_{Ai}$ ，则求出A与i之间的距离 $d_1$ 。

令r表示以A为中心，以 $d_1$ 为内半径，以 $d_2$ 为外半径的环形区域，将r加入集合R。转2；

4) 计算R中所有环形区域的交集 $A_0$ ；

5) 以 $A_0$ 的质心作为S的坐标。

ROCRSSI算法并非实际测距，而是利用未知节点与接收到的邻近锚节点RSSI大小的比较来定位自身。但考虑到无线信道易受障碍物的反射、绕射等作用 and 无线电异向传播特性的影响，使得算法的定位精度依赖于网络锚节点密度和周围部署环境，容错性较差。该算法需要每个锚节点广播两次信息，网络通信开销较大。但ROCRSSI算法简单，无需额外的硬件开销，网络成本较低，仍不失为一种理想的定位方案。

## 3 结论

针对无线传感器网络本身的特性，在二维环境中分别从通信开销、计算开销、定位精度、容错性、网络成本、覆盖速度、覆盖率几个方面来对DV-Hop、ROCRSSI两种定位算法性能进行理论评价，结果如表1所示。

表 1 DV-Hop 算法和 ROCRSSI 算法性能评价表

性能评价指标	DV-Hop 算法	ROCRSSI 算法
定位方式	分布式	分布式
测距方式	非测距	非测距
计算开销	$(18K - 17 + 2^3/3)F$	$2C(C + 4)F + \Delta$
通信开销	$2AN$	$A(1 + C)$
定位精度	适中(约 33%)	依赖锚节点密度和环境
容错性	不受测距误差影响,较高	较差
网络成本	较低	一般
覆盖速度	与网络规模有关	较慢
覆盖率	较高	与锚节点密度有关

注:  $N$  为网络节点数,  $A$  为网络锚节点数,  $F$  为单个 flop 的能量消耗,  $C$  为网络平均连通度,  $K$  为参与一次多边测量定位的锚节点个数。

## 4 结束语

DV-Hop 算法利用节点间大量的冗余信息来定位,是以通信代价换取的锚节点数量减少,要进一步优化锚节点的部署来提高对网络拓扑结构变化的适应能力和获得更高的覆盖率。该算法仅适合应用在各向同性网络中。在各向异性网络中,可以采用与 Exposure 模型<sup>[5]</sup>、分布式自散步算法<sup>[6]</sup>、基于虚拟力的算法<sup>[7]</sup>等相结合的方法来调整网络拓扑结构,提高网络覆盖率。ROCRSSI 定位算法在锚节点部署、定位精度和通信开销等方面仍有待改进的地方,基于 RSSI 与 DV-Hop 算法相结合来对节点实施定位也是一种较好的定位思路。

通过对上述 DV-Hop 算法和 ROCRSSI 算法两种非测距定位算法性能的评价,可以看到目前并没有一种算法是最优的,每个算法都有自己的应用条件和适用

范围,因此在实际应用中还应采取“均衡策略”选取合适的满足特定需求的定位方案。

## 参考文献:

- [1] 孙利民,李建中,陈渝,等. 无线传感器网络[M]. 北京:清华大学出版社,2005.
- [2] Bulusu N, Heidemann J, Estrin D. GPS-less low-cost outdoor localization for very small devices[J]. IEEE Personal Communications, 2000(10): 28-34.
- [3] Niculescu, Nath B. Ad-hoc positioning systems[C]//Proc. of the 2001 IEEE Global Telecommunications Conf. San Antonio: IEEE Communications Society, 2001: 2926-2931.
- [4] Liu Chong, Wu Kui, He Tian. Sensor Localization with Ring Overlapping Based on Comparison of Received Signal Strength Indicator[C]// Proceedings of Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference. Fort Lauderdale, Florida, USA: IEEE Press, 2004: 516-518.
- [5] Seapahn M, Farinaz K, Qu Gang, et al. Exposure in wireless ad-hoc sensor networks[C]// Proceedings of the 7th annual international conference on mobile computing and networking. Rome, Italy: ACM Press, 2001.
- [6] Heo N, Varshney P K. A distributed self spreading algorithm for mobile wireless sensor network[J]. Wireless Communications and Networking, 2003(3): 1597-1602.
- [7] Iyengar K C S S, Qi Hairong, Cho Eungchun. Grid coverage for Surveillance and target location in distributed sensor networks[J]. IEEE Transactions on Computers, 2002, 51(12): 1448-1453.

(上接第 61 页)

程对象,编译,链接,产生服务器端应用程序。

5) 在编写客户端应用程序时,添加对公共接口的引用,设计相关的界面和编写调用服务的代码,编译,链接,产生客户端应用程序。

.NET 的基本构件单元是以程序集(Assembly)的形式存在,因而通过上述的操作所得到的构件,其在部署、版本的控制和更新以及重用上,都提供了极大的方便。客户端通过获取服务器端的对象,来获得服务端的方法,这样保证了客户端与服务器端的相关对象的松散耦合,同时也优化了通信性能。

## 5 结束语

构件技术是软件复用的核心,随着 Internet 网络的广泛应用,对于分布式构件的要求也越来越高。而.NET 技术集合了过去相关开发平台的优点,其使得开发多语言、跨平台、跨区域、可复用的分布式构件成为

可能。文中所提出的基于 .NET 平台的分布式软件开发方法,能够很好地适应今后系统的演化,在开发同一领域的应用软件时,会大大缩短开发的生命周期,降低开发上的费用,减轻维护工作,从而很好地满足了系统的开发要求,对于用构件化开发方法开发分布式系统,具有一定的指导意义。

## 参考文献:

- [1] 李延春,晏敏. 软件构件技术的现状和未来[J]. 计算机工程与应用, 2003, 39(31): 86-96.
- [2] 王志坚,费玉奎,姜渊清. 软件构件技术及其应用[M]. 北京:科学出版社,2005.
- [3] 刘宪凯,张维石,罗武军. 基于 CORBA 的软件构件开发方法研究[J]. 计算机工程与应用, 2001, 37(19): 151-153.
- [4] Lowy J. Programming .NET Components[M]. Gravenstein Highway North Sebastopol: O'Reilly Media, 2005.
- [5] Barnaby T. .NET 分布式编程—C# 篇[M]. 黎媛,王小锋译. 北京:清华大学出版社,2004.