

锚节点稀疏的传感器网络节点自定位算法

刘明¹, 王婷婷¹, 周自波²

(1. 华中师范大学计算机科学系, 武汉 430079; 2. 华中师范大学网络教育学院, 武汉 430079)

摘要: 针对 Euclidean 算法中定位精度及覆盖率受锚节点密度影响较大的问题, 提出一种改进的分布式节点自定位算法。该算法将初始定位精度较高的节点升级为锚节点, 未知节点根据更新的锚节点位置信息循环求精, 并通过估计坐标值的方差来控制循环求精过程中的循环次数。仿真实验显示, 改进定位算法在锚节点密度较低的情况下能有效提高定位精度和覆盖率, 明显降低了对锚节点密度的依赖程度。
关键词: 无线传感器网络; 定位; 分布式; 循环求精

Self-localization Algorithm for Sensor Networks of Sparse Anchors

LIU Ming¹, WANG Ting-ting¹, ZHOU Zi-bo²

(1. Department of Computer Science, Central China Normal University, Wuhan 430079;
2. School of Network Education, Central China Normal University, Wuhan 430079)

【Abstract】 To address the problem that anchor ratio has a strong impact on localization error and coverage in Euclidean algorithm, this paper proposes an improved distributed localization algorithm. This method uses high localization accuracy nodes as new anchor nodes. According to the information, other nodes raise the localization accuracy by iterative refinements. And it controls the circulation times with the variance of coordinates. Simulation results show that, compared with Euclidean algorithm, improved localization algorithm can enhance the localization accuracy efficiently when the anchor ratio is lower. It is obvious that the influence of anchor ratio on the localization accuracy and coverage is less.

【Key words】 wireless sensor networks; localization; distributed; iterative refinements

1 概述

无线传感器网络是当前国际上倍受关注的、由多学科高度交叉的新兴研究领域。这种网络系统具有广阔的应用前景, 如国防军事、环境监测、交通管理、医疗卫生、反恐抗灾等领域。对于大多数应用, 无线传感器网络中的节点需要知道自身的物理位置, 才能详细说明“在什么位置或区域发生了特定事件”。传感器节点自身定位不但是实现对外部目标定位和追踪的技术基础, 另一方面, 了解传感器节点位置信息还可以提高路由效率, 为网络提供命名空间, 向部署者报告网络的覆盖质量, 实现网络的负载均衡以及网络拓扑的自配置^[1]。而人工部署和为所有网络节点安装 GPS 接收器都会受到成本、功耗、扩展性等问题的限制, 甚至在某些场合可能根本无法实现, 因此必须采用一定的机制与算法实现 WSN 的节点自身定位。

本文在继承 Euclidean 算法计算与锚节点相隔 2 跳的未知节点位置的思想的基础上, 提出了改进的定位算法, 将部分高精度节点升级为锚节点, 未知节点通过循环求精来减小测距误差影响。在循环求精过程中设置一个门限值, 以限制循环次数。笔者做了仿真实验以验证改进算法的性能, 对仿真结果进行分析, 并提出了进一步的研究方向。

2 Euclidean 定位算法

Euclidean 定位算法^[2-3]中基于围绕在锚节点周围的节点的局部几何拓扑, 给出了计算与锚节点相隔 2 跳的未知节点位置的方法。如图 1 所示, 未知节点 B、C 与锚节点 L 相邻; 节点 A 与 B、C 相邻, 对于四边形 ABCL, 所有边长和一个对角线 BC 的长度已知, 根据几何原理可计算出 AL 的长度。当未知节点获得与 3 个或更多锚节点距离后进行自身定位。AL

的计算公式如下:

$$\cos \angle ACB = \frac{d_{AC}^2 + d_{BC}^2 - d_{AB}^2}{2d_{AC}d_{BC}}$$

$$\cos \angle BCL = \frac{d_{BC}^2 + d_{CL}^2 - d_{BL}^2}{2d_{CL}d_{BC}}$$

$$d_{AL}^2 = d_{AC}^2 + d_{CL}^2 - 2d_{AC}d_{CL} \cos \theta$$

其中, 当未知节点 A 与 L 位于 BC 的异侧时, $\theta = \angle BCL + \angle ACB$; 当 A 与 L 位于 BC 的同侧时, $\theta = \angle BCL - \angle ACB$ 。

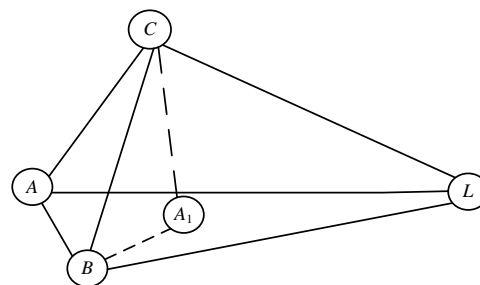


图 1 Euclidean 算法示意图

该算法适用于网络拓扑不规则的网络, 由于数据包只传送 2 跳, 因此该算法通信开销较小, 同时具有适当的计算开销和定位精度。文献[3]中的实验显示, 随着锚节点密度的降低, 定位误差也急剧降低, 可见该算法的定位精度和覆盖率受锚节点密度的影响较大。由此, 提出了改进的定位算法。

基金项目: 国家社会科学基金资助项目(07BYY033)

作者简介: 刘明(1967-), 男, 副教授、博士, 主研方向: 计算机网络, 嵌入式系统; 王婷婷, 硕士; 周自波, 工程师

收稿日期: 2009-06-30 **E-mail:** wttwj1@yahoo.com.cn

3 改进的定位算法

3.1 改进算法思想

考虑到无线传感器网络成本的因素,锚节点数量往往受到一定的限制,每个节点 2 跳范围内很难满足有 3 个以上的锚节点参与定位,要得到高精度和高覆盖率的定位需要较多的锚节点数量。笔者提出在不增加锚节点数量的情况下,将部分定位精度较高的节点升级为锚节点,增加未知节点定位中的锚节点数目,从而提高定位覆盖率。如图 2 所示,在未知节点 U_0 的 2 跳范围, U_2 根据到 3 个锚节点 L_1 , L_2 和 L_3 的距离,利用三边测量法确定自己的位置坐标,然后根据相隔 1 跳和 2 跳的锚节点数目,确定可升级为锚节点。如图 3 所示, U_2 升级为锚节点。而未知节点 U_0 只能获得到 L_1 和 L_2 2 个锚节点的距离,因而无法实现自身定位。但在图 3 中,由于节点 U_2 升级为锚节点, U_0 可获得到 L_1 , L_2 和 U_2 3 个锚节点的距离,从而可利用三边测量法确定自己的位置坐标。

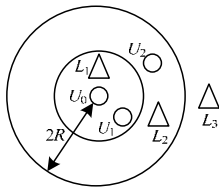


图 2 U_2 初始定位示意图

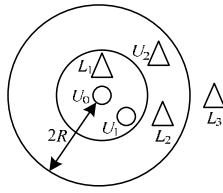


图 3 U_2 升级为锚节点示意图

一些已定位的节点也可根据新增加的锚节点重新进行位置估计,从而提高定位精度。但将部分高精度节点升级为锚节点虽然缓解了锚节点稀疏问题,但同时引进了一个误差累积的问题。笔者详细指定了未知节点接收位置估算并升级为锚节点的条件,使低精度节点不参与运算,以减少误差累积的影响。

现有的分步求精定位算法,如 Two-phase positioning^[4], n-hop multilateration primitive 等通过循环求精来减小测距误差的影响,实现较高定位精度。并且在循环求精过程中设置一个门限值,以消除循环次数无法估计的缺点,同时也是在保证一定的定位精度和覆盖率的条件下,尽量减少节点的通信量和计算量,以延长网络的生命周期。

3.2 改进的算法描述

未知节点具体的定位过程分为以下 3 个步骤:

(1) 初始定位

首先锚节点广播一个消息给邻居节点,广播消息形式为 $M: \{ID, (x,y), RAT, HOP\}$

其中, RAT 为节点的等级,代表了该节点的定位精度,按精度由高到低分别为 $anchor$, $anchor1$, $anchor2$ 和 $common$, $anchor$ 表示在网络部署时已知位置的锚节点, $anchor1$ 和 $anchor2$ 表示定位精度较高而升级的锚节点, $common$ 表示根据锚节点和升级的锚节点而定位出自身位置的普通节点; HOP 为消息传播的跳数,初始值为 1,当未知节点向其邻居节点转发收到锚节点的广播消息时,将 HOP 加 1,对于 $HOP=2$ 的锚节点信息,节点不再转发出去。通过 RSSI 测量节点间点到点距离,然后利用 Euclidean 算法计算出到相隔 2 跳的锚节点的距离。当未知节点获得到 3 个或更多于 3 个不在同一直线上的锚节点的距离时,分别采用三边测量法或极大似然估计法计算出自身位置坐标。

(2) 确定是否升级为锚节点

在节点获得多个锚节点的信息中,距锚节点跳数较小且锚节点数目较多的未知节点,其定位精度较高,所以依据 RAT

的值来确定节点是否升级为锚节点。令 RAT 值等于 $anchor$ 的个数为 $anchornum$, RAT 值等于 $anchor1$ 的个数为 $anchor1num$, RAT 值等于 $anchor2$ 的个数为 $anchor2num$ 。

当 $anchornum \geq 3$ 时,令该节点的 RAT 值为 $anchor1$ 。

当 $anchornum + anchor1num \geq 3$ 时,令该节点的 RAT 值为 $anchor2$ 。

否则,令该节点的 RAT 值为 $common$,即该节点不能升级为锚节点。

升级为锚节点的节点向网络中广播 M 消息。

(3) 循环求精

该阶段本质上是一个迭代过程,节点通过接收到步骤(2)中升级锚节点信息来更新自身位置坐标,同时在升级锚节点 2 跳内的其他锚节点也会接收到此信息,再次计算自己的位置坐标后,向网络中广播 M 消息。类似此过程,进入下一轮循环。假设每个节点都保存自身最新的 N 次坐标值,从实验中发现,当循环求精一段时间后,估计值趋于一个稳定值,此时即可停止循环。

由于随机变量的方差是刻画其取值的分散程度,若将节点坐标 (\hat{x}, \hat{y}) 中的 \hat{x} 和 \hat{y} 视为随机变量 X 和 Y ,则其方差 $D(X)$ 和 $D(Y)$ 就表示节点多次定位的坐标值的变化大小和稳定程度。随机变量 X 和 Y 的方差公式如下:

$$D(X) = \frac{1}{N} \sum_{i=1}^N (\hat{x}_i - \bar{X})^2$$

$$D(Y) = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - \bar{Y})^2$$

其中, N 为节点的位置坐标个数,一般取 3~5 个坐标值即可。 \bar{X} 和 \bar{Y} 分别代表 N 个坐标值的平均值。随机变量的方差取值区间为 $[0, +\infty)$,当 N 个样本值均相同时,样本的方差为 0;方差越大,表示随机变量样本值波动越大。设置一个门限值 η ,用来表示节点的 N 个位置坐标趋于一致的程度。根据实验的数据测得, $\eta \in [1, 3]$ 可表示可接受的坐标值离散程度的范围。当 N 个 X 坐标值和 N 个 Y 坐标值的方差均小于 η ,则认为节点坐标已接近真实位置值,停止迭代过程,算法结束。

另外考虑到一种特殊情况,当未知节点的邻居节点和相隔 2 跳节点中, RAT 值为 $anchor$, $anchor1$ 和 $anchor2$ 的锚节点的总数小于 3 时,该未知节点向网络中广播请求邻居节点和 2 跳邻居节点的位置信息的信息,此时 $common$ 节点也将参与定位计算,以保证整个网络节点定位的覆盖率。

4 算法仿真

在仿真实验中,采用以下参数衡量定位算法的性能。

定位精度:误差值(估测位置 and 实际位置之间的距离)与节点无线射程(通信半径)的比例表示。将误差值设为

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (\hat{x}_i - \bar{x}_i)^2 + (\hat{y}_i - \bar{y}_i)^2}{N}}$$

其中, N 为网络中的总节点数; (\hat{x}_i, \hat{y}_i) 为定位算法的定位结果; (\bar{x}_i, \bar{y}_i) 为节点的真实坐标。

覆盖率:可实现定位未知节点数与总未知节点数的比值。

仿真分析的网络环境设置如下:设置 200 个传感器节点随机均匀分布在 700 m×700 m 正方形区域内,每个节点最大无线通信半径设为 100 m,通过调节节点的无线通信半径来调节网络平均连通度, RSSI 测距误差为 20%, $\eta=3$ 。本实验数据来自 50 次仿真实验的平均值。

图 4 描述了 Euclidean 和改进算法的定位精度(纵坐标)随锚节点密度(横坐标)的增加而变化的情况。从图中可以看出,

当锚节点密度小于 20% 时,改进算法比 Euclidean 算法的定位精度提高 7%~20%,这是因为需要定位的节点在升级锚节点为参考点的情况下进行循环求精定位,减少了测距误差的影响,使其定位结果更加精确,同时限制精度不高的节点不参加定位运算,也提高了定位精度。

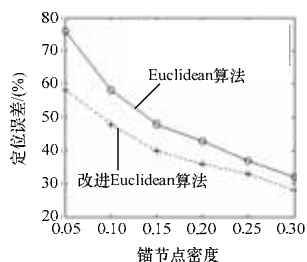


图 4 网络定位精度比较

图 5 描述了 Euclidean 和改进算法的定位覆盖率随锚节点密度的增加而变化的情况。与 Euclidean 定位算法相比,改进算法的覆盖率明显提高了,尤其是在锚节点密度较低的情况下,改进算法的优势更为明显。

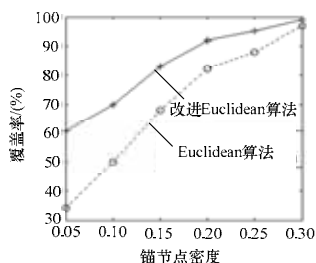


图 5 网络覆盖率比较

如图 5 所示,当锚节点密度小于 20% 时, Euclidean 算法的定位覆盖率随锚节点密度的减小而迅速降低,而改进算法

的定位覆盖率随锚节点密度的减小变化较小。可见改进算法的定位性能对锚节点密度的依赖程度有所减小,这是由改进算法的定位机制决定的,将一部分定位精度较高的节点升级为锚节点,增加了参考节点的数量,从而有更多的未知节点进行定位,提高了网络定位覆盖率。

5 结束语

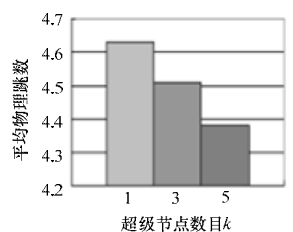
本文设计了一种基于 Euclidean 的完全分布式算法。将部分高精度节点升级为锚节点的方法,其他未知节点根据锚节点更新的位置信息,通过循环求精定位运算以提高定位精度。仿真证明该算法提高了网络节点定位的覆盖率和定位精度,明显减小了定位精度及定位覆盖率对锚节点密度的依赖程度。在优化的过程中,由于发送升级的锚节点信息和节点位置循环求精而增加了通信数据量,从而增加了通信开销和计算开销,因此下一步的工作将考虑定位精度、定位覆盖率与通信开销折中的方案。

参考文献

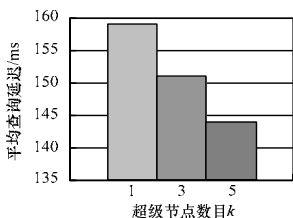
- [1] Cerpa A, Estrin D. Ascent: Adaptive Self-configuring Sensor Network Topologies[J]. IEEE Transactions on Mobile Computing, 2004, 3(3): 272-285.
- [2] Niculescu D, Nath B. Ad-hoc Positioning Systems(APS)[C]//Proc. of the IEEE Global Telecommunications Conference. San Antonio, USA: IEEE Communications Society, 2001: 2926-2931.
- [3] Niculescu D, Nath B. DV Based Positioning in Ad Hoc Networks[J]. Journal of Telecommunication Systems, 2003, 22(1-4): 267-280.
- [4] Savarese C, Rabaey J, Langendoen K. Robust Positioning Algorithms for Distributed Ad-hoc Wireless Sensor Networks[C]//Proc. of the USENIX Technical Annual Conference. Monterey, Canada: [s. n.], 2002: 317-327.

编辑 顾逸斐

(上接第 118 页)



(a) k 取不同值时物理跳数的分布



(b) k 取不同值时延迟的分布

图 4 k 取不同值时物理跳数和延迟的比较

5 结束语

本文提出的基于物理拓扑的双向搜索 Chord 路由算法分别从路由长度、节点之间的延迟、缓存查询结果等几个方面来减少查询时间。通过将物理上相邻近的节点组织成一个簇从而减少延迟。针对 Chord 中查询采用单向搜索的不足,充

分利用节点逆时针方向上的路由信息,可以把 Chord 上查询的平均路由长度减少。而超级节点的引入进一步减轻了网络的负担,但同时也增加了维护的开销,在将来的工作中将考虑改进的方法。

参考文献

- [1] Stoica I, Morris R, Karger D, et al. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications[C]//Proc. of ACM SIGCOMM'01. San Diego, California, USA: [s. n.], 2001: 17-32.
- [2] 邹东尧, 宋俊德. 一种基于物理网络拓扑的高效 Chord 模型[J]. 计算机工程, 2008, 34(6): 127-129.
- [3] Padmanabhan V N, Subramanian L. An Investigation of Geographic Mapping Techniques for Internet Hosts[C]//Proc. of ACM SIGCOMM'01. San Diego, CA, USA: [s. n.], 2001: 173-185.
- [4] Krishnamurthy B, Wang Jia. On Network-aware Clustering of Web Clients[C]//Proc. of ACM SIGCOMM'00. Stockholm, Sweden: [s. n.], 2000: 97-110.
- [5] Ratnasamy S, Handley M, Karp R, et al. Topologically-aware Overlay Construction and Server Selection[C]//Proc. of INFOCOM'02. New York, USA: [s. n.], 2002: 1190-1199.
- [6] Yang B, Hector G M. Designing a Super-peer Network[C]//Proc. of the 19th International Conference on Data Engineering. Bangalore, India: IEEE Computer Society, 2003: 49-60.

编辑 任吉慧