

# 无线传感器网络 APIT 定位算法及其改进

周四清<sup>1,2</sup>, 陈锐标<sup>2</sup>

(1. 暨南大学经济学院, 广州 510632; 2. 暨南大学信息科学技术学院, 广州 510632)

**摘要:** 针对无线传感器网络中 APIT 定位算法定位覆盖率不足的问题, 提出一种改进的 IAPIT 算法, 并从不同的锚节点比例、节点通信半径以及同一锚节点比例等方面比较 2 种算法的性能, 仿真实验结果表明, IAPIT 定位算法在定位覆盖率上有明显提高。

**关键词:** 无线传感器网络; 定位; APIT 算法; IAPIT 算法; 覆盖率

## APIT Location Algorithm & Its Improvement in Wireless Sensor Network

ZHOU Si-qing<sup>1,2</sup>, CHEN Rui-biao<sup>2</sup>

(1. College of Economics, Jinan University, Guangzhou 510632;

2. College of Information Science & Technology, Jinan University, Guangzhou 510632)

**【Abstract】** Aiming at the problems of lower location coverage rate for APIT location algorithm in Wireless Sensor Network(WSN), an improved algorithm called IAPIT is proposed. The performances of these two algorithms are compared in terms of different anchor ratios, different communication radius of nodes and the same anchor ratio node. Simulation experimental results show this improved algorithm has higher location coverage rate.

**【Key words】** Wireless Sensor Network(WSN); location; APIT algorithm; IAPIT algorithm; coverage rate

### 1 概述

定位技术是无线传感器网络多种关键支撑技术之一<sup>[1]</sup>, 在传感器网络实际应用工程项目中, 利用传感器节点获得包含绝对地理位置或者相对参照系位置的感知数据具有十分重要的意义。

无线传感器网络的定位问题分为 2 类: (1)无线传感器网络对自身传感器节点的定位; (2)利用传感器节点对外部目标的跟踪定位, 而节点自身的定位是实现跟踪定位外部目标的基础<sup>[1]</sup>。

节点自身定位主要有 4 个方面的意义: (1)确定发生感知数据的具体地理位置或者相对某一个参照系的位置, 并为进一步跟踪定位进入无线传感器网络的外部目标打下基础。(2)可以节约节点的功耗, 延长无线传感器网络的生命周期。研究表明: 在节点的感知、传输和计算等行为中, 传输信息是最消耗能源的因素。当只对某一局部区域的感知数据感兴趣时, 就可以只让该区域中的节点工作, 而其他节点处于睡眠状态, 这样可以降低节点的能耗, 提高整个网络的生命周期。(3)可选择最短的路径传输信息, 以使在提高路由效率的同时, 减少信息传递次数。(4)进行网络管理, 利用传感器节点传回的位置信息构建网络拓扑图, 并实时统计网络覆盖情况, 对节点密度低的区域及时采取必要措施。

考虑到成本、体积和功率等因素的制约, 目前发展相对成熟的全球定位系统 GPS 并不适合无线传感器网络节点的定位。因此, 研究开发适合无线传感器网络的定位算法具有广泛的实际应用背景和重要的理论价值。

本文主要针对 APIT<sup>[2]</sup>定位算法在节点定位覆盖率方面存在的问题, 提出一种改进的 IAPIT 算法。

### 2 APIT 定位算法

#### 2.1 APIT 算法的基本原理

APIT 定位算法的基本思想是: 假设未知节点具有获取锚节点信息的能力, 通过锚节点的通信传播过程, 未知节点获得所有邻居锚节点的信息, 包括锚节点的位置信息和到达信号的能量信息(传播过程能量有损耗)。对于每个未知节点, 假设其邻居锚节点数目为  $n$ , 则每次从这  $n$  个锚节点中取出 3 个为一组, 判断以它们为顶点的三角形是否包含该未知节点(采用三角形相似内点测试法), 穷尽所有的锚节点组合, 有  $C_n^3$  种。对所有包含该未知节点的三角形求重合区域(采用格子扫描算法), 最后求出该重合区域的质心位置(将所有小格子求出各自质心位置, 再取算术平均值), 即为该未知节点的估计位置。

#### 2.2 APIT 算法流程

通过仿真分析可知, APIT 算法存在以下 2 个主要问题:

(1)在网络覆盖区域的中部, 未知节点的邻居锚节点有很多, 但是可能没有一个由 3 个锚节点组成的三角形可以包含该未知节点, 这就造成这些未知节点在运行定位算法后成为未确定节点; (2)在网络覆盖区域的边缘地带, 未知节点拥有比较少的邻居锚节点, 这样更加容易造成它们无法满足 APIT 定位条件, 甚至有些节点的邻居锚节点数少于 3 个, 它们都将

**基金项目:** 国家社科基金资助项目(06BJL066); 暨南大学引进优秀人才科研启动基金资助项目(51205068)

**作者简介:** 周四清(1964—), 男, 副教授、博士后, 主研方向: 智能信息处理; 陈锐标, 硕士研究生

**收稿日期:** 2008-11-10 **E-mail:** zsqhu2004@163.com

成为未确定节点。APIT 算法的流程如图 1 所示。

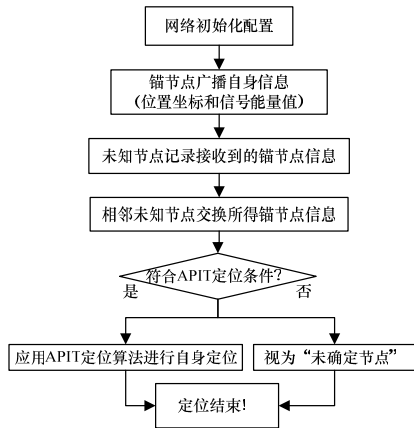


图 1 APIT 算法流程

定位覆盖率体现节点的有效利用比例，节点的定位覆盖率过低，相当于有许多传感器节点提供的感知数据不可信，或者说没有价值。没有位置信息的感知数据信息是毫无用处的，特别是当节点规模较大时，定位覆盖率的低下意味着有很多节点在做无用功。为改进 APIT 算法，提高节点的定位覆盖率，同时保持良好的定位精度，使其适用于较大规模网络的应用，具备较强的扩展性，本文提出 IAPIT 定位算法。

### 3 APIT的改进算法IAPT

#### 3.1 IAPIT算法的基本原理

在深入研究 APIT 算法的基础上提出 IAPIT 算法，它的全称是 Improved Approximate PIT Test。算法的主要思路是将 3 边测量法以及几何上的由已知 2 点在辅助条件下求解 2 圆交点的方法融入到 APIT 算法中。对于符合 APIT 定位条件的节点，按照 APIT 算法进行定位，对于不符合 APIT 定位条件的节点，作进一步的处理，提高定位覆盖率。当邻居锚节点数大于等于 3 个时，采用 3 边测量法进行定位；当邻居锚节点数为 2 个时，由几何上求 2 圆交点的方法求出 2 交点，再依据其他条件判断哪一个交点为节点的估计位置。

#### 3.2 IAPIT算法流程

IAPIT 算法流程如图 2 所示。

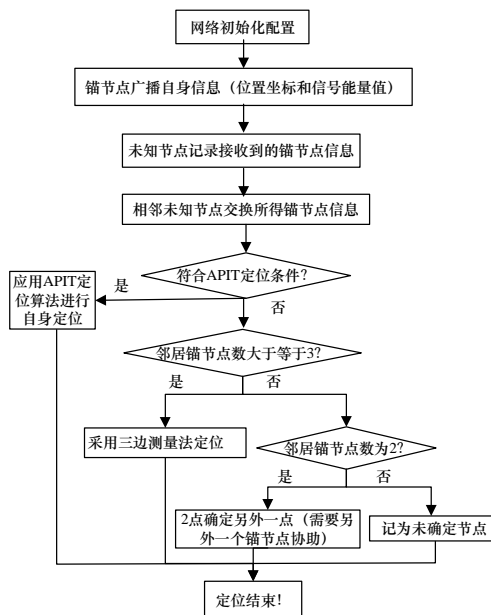


图 2 IAPIT 定位算法流程

## 4 IAPIT算法与APIT算法仿真分析

### 4.1 仿真平台的搭建

本文采用的仿真软件是 Visual C++<sup>[3]</sup>与 Matlab<sup>[4]</sup>，每次仿真结果都是通过运行算法 50 次，然后取平均值得到的。仿真所采用的各种参数如下：

(1) 传感器节点部署的网络区域是 40 m×40 m 的正方形区域。

(2) 节点是随机部署在网络区域，节点的通信半径取为  $R=10$  m，无论是锚节点或者是未知节点，它们的通信半径都是一样的。

(3) 节点总数有 100, 150, 200, 300 等，节点的锚节点数也各不相同，这样做的目的是仿真新算法对于各种不同配置的网络有什么不同特性，以期找到比较适合应用该算法的网络配置。

### 4.2 APIT与IAPIT仿真分析

图 3 和图 4 分别为在同一锚节点比例与不同锚节点比例情况下，APIT 与 IAPIT 的定位覆盖率及定位精度变化情况。

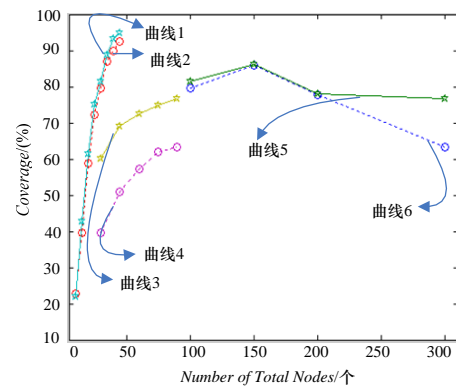


图 3 APIT 与 IAPIT 的定位覆盖率

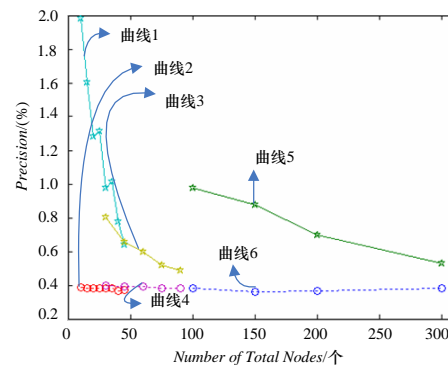


图 4 APIT 与 IAPIT 的定位精度

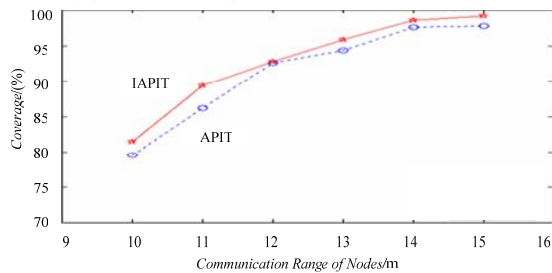
对图 3、图 4 的说明如下：

(1) 图 3 中的曲线 1 和曲线 2 分别为不同锚节点比例下，节点总数为 100 时，IAPIT 和 APIT 算法的定位覆盖率；曲线 3 为不同锚节点比例下，节点总数为 300 时，IAPIT 算法的定位覆盖率；曲线 4 为不同锚节点比例下，节点总数为 400 时，APIT 算法的定位覆盖率；曲线 5 和曲线 6 分别为锚节点比例为 30% 时，IAPIT 和 APIT 算法的定位覆盖率。

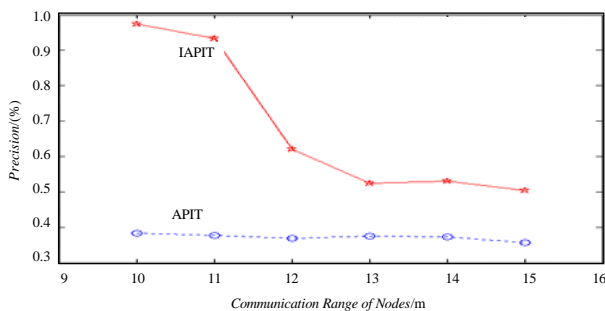
(2) 图 4 中的曲线 1 和曲线 2 分别为不同锚节点比例下，节点总数为 100 时，IAPIT 和 APIT 算法的定位精度；曲线 3 和曲线 4 分别为不同锚节点比例下，节点总数为 300 时，IAPIT 和 APIT 算法的定位精度；曲线 5 和曲线 6 分别为锚节点比例为 30% 时，IAPIT 和 APIT 算法的定位精度。

可以看出,在一定网络区域内,随着节点平均连通度的增加,IAPIT 算法在提高定位覆盖率的同时,定位精度也得到较好的改善。

图 5 为这两种算法在节点通信半径分别取 10 m~15 m 时定位覆盖率和定位精度的变化情况。



(a)APIT 和 IAPIT 定位覆盖率



(b)APIT 和 IAPIT 定位精度

图 5 不同节点通信半径下的定位覆盖率和定位精度

从图 5(a)可知,IAPIT 的定位覆盖率比 APIT 好一些,且随着通信半径的增加,IAPIT 的上升势头明显比 APIT 强。图 5(b)则说明两者的定位精度都随着通信半径的增加而得到改善,且 IAPIT 的改善幅度很明显,同时还能看出,当节点通信半径达到 13 m 以上时,两者的定位精度很难再得到明显改善。该仿真结果说明,当网络部署一定时,通信半径的增加对改善 IAPIT 的定位精度较明显,但通信半径增加到一定范围后,IAPIT 的定位精度基本维持不变。

## 5 结束语

通过仿真对比得出以下结论:

(1)在节点总数少于 200 的情况下,IAPIT 算法的定位覆盖率比 APIT 算法稍好一点,提高约 5%,但定位精度差很多,APIT 算法的定位精度基本恒定在 0.4%附近,而 IAPIT 算法的误差较大。

(2)随着节点总数的增加,即随着节点连通度的增加,IAPIT 的性能远胜于 APIT,且改善的幅度非常明显。这说明 IAPIT 的扩展性要好,而且比较适合中大型规模网络的应用。

(3)IAPIT 不适合应用于对定位精度要求太高的场合。由仿真可见,IAPIT 的定位精度一般都大于 0.5%,在锚节点比例为 30%的情况下,其定位精度下降到 0.49%。虽然随着锚节点比例的增长其定位精度可以进一步下降,但这会使网络的部署成本大幅度增加。

(4)IAPIT 的性能受制于未知节点的邻居锚节点数为 2 的情形。该情况可以大幅度降低 IAPIT 的定位精度,这在节点数较少的时候尤其突出。所以,如何找到更好的定位位置将是进一步提升 IAPIT 性能的关键。

(5)定位精度与定位覆盖率是一对不可完全调和的矛盾。提高定位覆盖率是以牺牲定位精度为代价的。但通过网络配置环境的优化,可以降低这种不调和现象。

## 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] Blum B M, Stankovic J A, Abdelzaher T. Range-free Localization Schemes for Large Scale Sensor Networks[C]//Proc. of the 9th Annual Int'l Conf. on Mobile Computing and Networking. San Diego, CA, USA: [s. n.], 2003.
- [3] 谭浩强. C++程序设计[M]. 北京: 清华大学出版社, 2004.
- [4] 张平. MATLAB 基础与应用简明教程[M]. 北京: 北京航空航天大学出版社, 2001.

编辑 陈文

(上接第 86 页)

网络协议包头的处理利用了 3 个 PicoBlaze 实现:

(1)由于只需要支持远程启动的功能,因此只实现简单的 ARP(无 ARP 缓存)和 ICMP(echo)这 2 个请求,并且 BOOTP 请求对系统的负载不大,完全可以利用一个 PicoBlaze 实现,程序空间使用率超过 90%。同时利用一张存有 IP-MAC 地址表的 ROM 与 PicoBlaze 相连,作为 BOOTP 分配 IP 所用。

(2)TFTP 应用层的网络协议需要相对复杂的处理过程,并且当多台客户端同时启动进行文件传输时,对服务器的负荷较大,因此,单独作为一条通路实现。将 MAC 层、UDP/IP 层的处理合并为一个 PicoBlaze 完成,TFTP 层用另一个 PicoBlaze 完成。

当需要支持启动的客户端较多时,为了增加系统的处理性能可以增加 TFTP 处理通路的并行条数,处理过程完全可以移植,只需要修改接收调度的模块。

## 5 结束语

FPGA 具有很好的并行特性,网络协议具有分层结构,文中介绍的多个 PicoBlaze 软核的处理结构能够很好地适合网络复杂处理需要,PicoBlaze 简单的特点使得普通资源的

FPGA 上也能集成更多的微处理器。怎样充分发挥挖掘出该结构的特点,是否能够通过集成相当多的微小处理器也达到高速处理的效果,仍然需要进一步的研究。

## 参考文献

- [1] Chapman K. PicoBlaze 8-bit Embedded Microcontroller User Guide for Spartan-3, Virtex-II and Virtex-II PRO FPGAs[EB/OL]. (2005-11-21). <http://www.xilinx.com/bvdocs/userguides/ug129.pdf>.
- [2] Ravindran K, Satish N, Jin Y, et al. An FPGA-based Soft Multiprocessor System for IPv4 Packet Forwarding[C]//Proc. of the 15th Field Programmable Logic and Applications. Tampere, Finland: [s. n.], 2005: 487-492.
- [3] Aritz S L. Analysis of the FSMs Implementation with Mini-microprocessors in FPGAs[C]//Proc. of the IEEE International Symposium on Industrial Electronics. Vigo, Spain: [s. n.], 2007: 2290-2294.
- [4] 秦思林,张盛兵,周昔平. 网络处理器并行度研究[J]. 计算机工程, 2007, 33(18): 116-118.
- [5] Sollinsl K. The TFTP Protocol (Revision 2)[S]. RFC 1350, 1992.

编辑 任吉慧

