

# 基于无线传感器网络的算法研究

## ——AC-HOP 算法改进

Version : 1.0.0

2010-07-08

### DOCUMENT HISTORY

Ed.	Version	Author	Change
1	1.0.0	侯毅光	Initial

班级：软设 2 班

联系电话：18912798657

E-Mail: [hyg1943@mail.ustc.edu.cn](mailto:hyg1943@mail.ustc.edu.cn)

# 目 录

1 项目概述.....	3
1.1 研究背景.....	3
1.2 定位算法研究意义.....	3
1.3 定位算法评判标准.....	3
2 项目需求分析.....	4
3 系统设计描述.....	5
3.1 概论.....	5
3.2 经典算法举例.....	5
3.3 对算法的改进.....	7
4 系统测试.....	9
4.1 编写目的.....	9
4.2 算法测试.....	10
4.2.1 算法测试环境.....	10
4.2.2 定位算法测试指标.....	11
4.2.3 测试范围: .....	11

4.3 测试报告.....	11
4.3.1 测试用例.....	11
4.3.2 测试事项.....	15
4.3.2.1.1 测试进度安排.....	15
4.3.2.1.2 分析测试.....	15
5 项目及课程总结.....	16
6 参考资料.....	17

## 1 项目概述

### 1.1 研究背景

随着微机电系统，片上系统，无线通信和低功耗嵌入式技术的飞速发展，集成了传感器，微机电系统和网络三大技术而成的无线传感器网络是一种全新的信息获取和处理技术。并以其低功耗，低成本，分布式和自组织的特点带来了信息感知的一场变革。简而言之，无线传感器网络就是由部署在检测区域内大量的廉价微型传感器节点组成，通过无线通信方式组成了一个多跳自组织网络。

无线传感器网络具有非常广阔的应用前景。美国商业周刊和 MIT 技术评论在预测未来结束发展中，分别将无线传感器网络列为 21 世纪最具影响力的 21 项技术和改变世界的十大技术之一。无线传感器网络与塑料电子学，仿生人体器官并称为全球未来的三大高科技产业。

传感器节点定位是无线传感器网络应用研究的一个非常重要的方面。基于传感器节点成本低的要求，GPS 信号定位方法则被淘汰掉了。由于无线传感器网络存在网络自组织性，节点分布随机性以及网络拓扑的不确定性等特点，致使定位方法必须在分布性，自适性，可扩展性方面有比较杰出的性能。通过比较目前已经提出的多种定位算法可以发现，基于锚节点位置的节点定位算法在以上均具有明显的优势。根据定位过程中是否对节点相互距离进行测量，定位算法分为非基于测距的定位算法与基于测距的定位算法，二者相互比较，非基于测距的定位算法具有成本低，功耗小，抗测距噪声能力强和硬件设备简单易行等优势，定位精度不及基于测距的定位算法是其劣势，但在当今无线传感器网络大量应用的领域，这个精度足以接受，因此非基于测距的定位算法今年倍受关注。目前已发展的非基于测距的定位算法主要有质心算法，凸规划算法，DV-Hop 算法，Amorphous 算法，MDS-MAP 算法。

### 1.2 定位算法研究意义

须知道节点感知数据的发生位置才有应用价值，这是应用的直接要求。

无线传感器网络的很多通信协议是在已知节点位置的基础上运行的。另外基于节点的已知位置可优化网络运行期间的值守调度机制，使网络中冗余节点不定期地轮休以延长寿命。

### 1.3 定位算法评判标准

**定位精度：** 节点定位技术首要的评价指标就是定位精度，一般用误差值与节点无线射程的比例表示。用二维网格划分定位区域，精度为网格大小。

**规模：** 不同的节点定位系统或算法也许可在园区内、建筑物内、一层建筑物或仅仅是一个房间内实现定位。另外，给定一定数量的基础设施或在一段时间内，一种技术可以定位多少目标也是一个重要的评价指标。

**锚节点密度：** 锚节点的费用会比普通节点高两个数量，关系到系统成本。

**节点密度：** 关系到网络通信质量和系统成本。

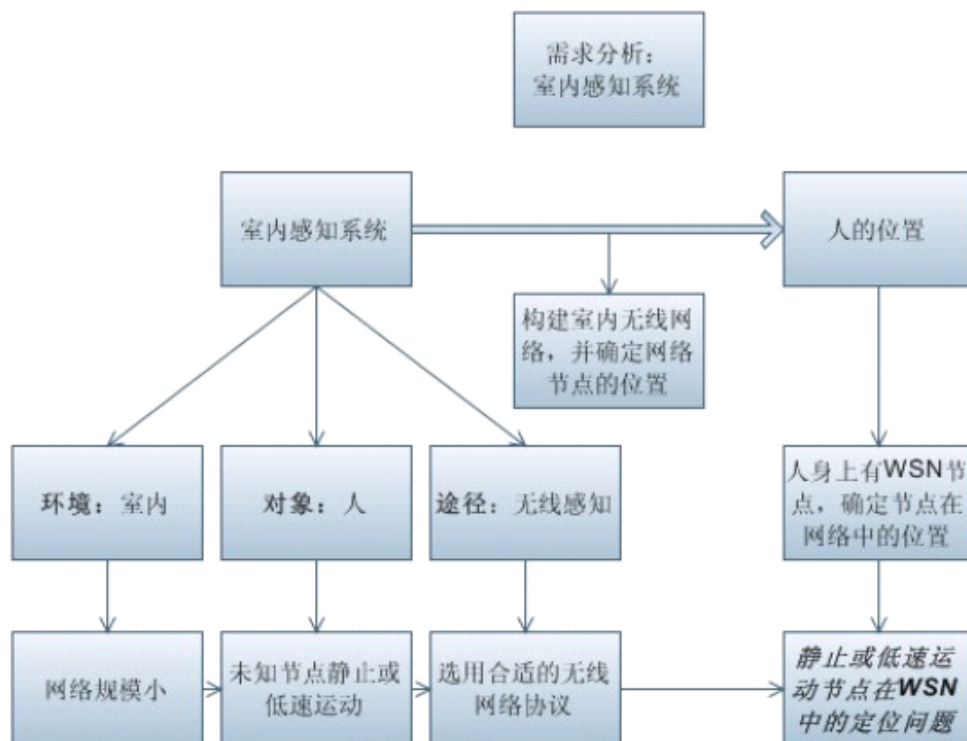
**容错性和自适应性：** 线传感网络本身特点决定。

**功耗、代价（时、空成本）。**

## 2 项目需求分析

建立室内环境的感知系统，可以确定室内环境中人的位置。

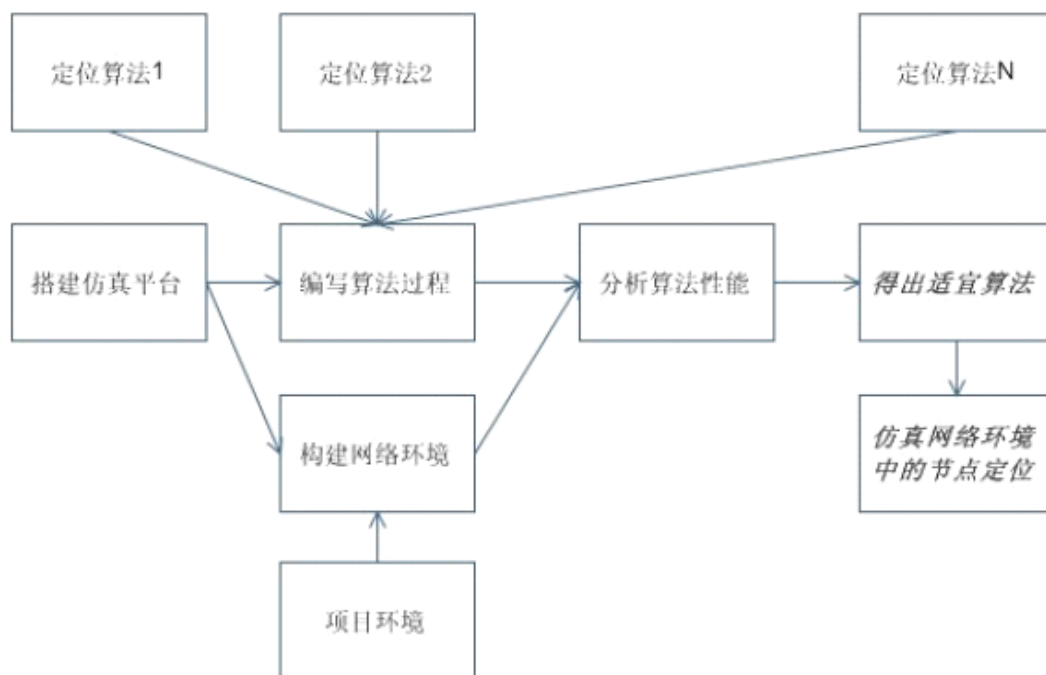
实现的途径可以通过构建室内无线网络，确定网络中结点的位置，已达到定位的目的。但是室内又有网络规模小，结点密度低，受干扰多，应用场景复杂等特点的影响。



### 3 系统设计描述

#### 3.1 概论

解决方案：首先调研已有的定位算法，找出与应用需求相近的算法。然后分析算法的优缺点，结合应用环境提出改进。



#### 3.2 经典算法举例

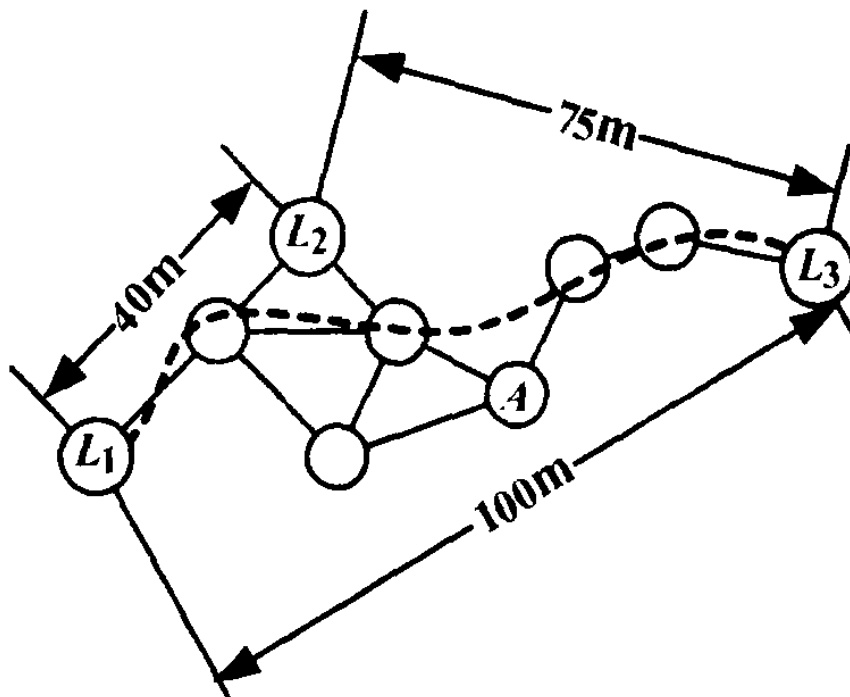
调研分析结果是 AC-HOP 算法最适合此应用场景。以下是 DV-Hop 算法介绍：

美国 Rutgers University 的 Niculescu 等人利用距离矢量路由和 GPS 定位原理提出的一系列分布式定位算法，合称为自组织定位系统。包括六种定位算法：DV-Hop 算法，DV-Distance 算法，DV-Euclidean 算法，DV-Coordinate 算法，DV-Bearing 算法和 DV-Radial 算法。六者中应用最成功，使用最广泛的当属 DV-Hop 定位算法。DV-Hop 算法和质心算法一样，整个定位过程不依赖于昂贵的测距，不同的是 DV-Hop 算法利用了多跳锚节点信息来参与未知节点的定位，其定位覆盖率远远高于质心算法。

DV-Hop 算法由三个阶段组成。第一阶段，使用典型的距离矢量交换协议，通过节点间的信息交换，使网络中所有节点获得与锚节点之间的跳距数。第二阶段，在获得其他锚节点的位置信息和跳数信息后，锚节点计算网络平均每跳距离，然后将其作为一个跳距校正值广播至网络中。

并且这个校正值是采用可控泛洪法传播的，即未知节点获得的校正值是第一个达到它的校正值。在大型网络中，可以通过为数据包设置一个 TTL 域来减少信息量以提高网络质量。在接收到校正值后，未知节点根据跳数信息和跳距信息估计其与附近锚节点的欧式距离。第三阶段，当未知节点获得与至少三个锚节点的欧式距离后，可以通过三角测量定位法获取未知节点的坐标，达到定位目的。

下面举例说明：



上图为由 9 个节点组成的传感器网络结构图。其中， $L_1, L_2, L_3$  为 3 个已知锚节点，其余为未知节点。下面以未知节点 A 为例来阐述 DV-Hop 算法。

第一阶段，锚节点广播包含自身位置信息与跳数的数据包。接收节点记录各锚节点至它的最小跳数，然后跳数加 1 转发给邻居节点。网络中的所有未知节点都记录下其至各锚节点的最小跳数。对于 A 节点来说，它至锚节点  $L_1, L_2, L_3$  的最小跳数分别是 3, 2, 3 跳。

第二阶段，每个锚节点根据记录的其他锚节点的位置信息及相互跳数来估算属于该锚节点的平均每跳距离：

$$HS_i^D = \frac{\sum_{j \neq i} d_{i,j}}{\sum_{j \neq i} h_j} \sim \sim (1)$$

式 1 中的  $d_{i,j}$  是标号为 i 和 j 的锚节点间的欧式距离，由于锚节点位置是已知的，这个很容易得到。

$h_j$  表示参考锚节点 i 至锚节点 j 间的跳数。从图中数据可以得到，属于 3 个锚节点的

估计每跳距离分别是  $HS_1^D = (40+100) / (2+6) = 17.5$ ， $HS_2^D = (40+75) / (2+5) = 16.42$ ，

$HS_3^D = (75+100) / (5+6) = 15.91$ 。对于 A 节点来说， $L_2$  锚节点与其跳数最小，最先收到  $L_2$

锚节点的平均每跳距离估计值，则 A 节点与三个锚节点间的距离分别是： $L_1 - 3 \times 16.42$ ， $L_2 - 2 \times 16.42$ ， $L_3 - 3 \times 16.42$ 。

第三阶段，利用三角测量定位法确定节点 A 的位置。假设某未知节点 u，并已测得其与 n 个锚节点的距离， $(x_i, y_i)$  为第 i 个锚节点坐标， $d_i$  为 u 至第 i 个锚节点的距离。

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 = d_2^2 \\ \vdots \\ (x_n - x)^2 + (y_n - y)^2 = d_n^2 \end{cases} \sim \sim (2)$$

上图为计算节点 u 坐标的方程组，二元二次方程组，理论上最少需要三个锚节点坐标即可保证方程组能求出唯一解。

### 3.3 对算法的改进

DV-Hop 算法具有良好的分布性和可扩展性，但定位性能却受到网络拓扑的约束，在网络节点分布非均匀情况下，节点位置估计精度不高，下面我们就此提出对算法的改进。

#### 一、基于最小均方误差准则球的平均每跳距离

$$HS_i^D = \frac{\sum_{j \neq i} d_{i,j}}{\sum_{j \neq i} h_j} \sim \sim (1)$$

上式为传统方法计算平均每跳距离公式，此公式是基于无偏估计准则来计算的，即下式恒等于零。



$$\frac{1}{N-1} \sum_{j \neq i} (d_{i,j} - HS_i^D \times h_j) \sim \sim (3)$$

其中 N 为锚节点个数，传统方法得到的估计误差均值为零，然而，根据参数估计理论，作为估计子误差的代价函数，使用均方误差比使用方差更为合理。于此，我们提出基于最小均方误差准则来计算平均每跳距离，即利用下式最小化来确定估计值。

$$\frac{1}{N-1} \sum_{j \neq i} (d_{i,j} - HS_i^N \times h_j)^2 \sim \sim (4)$$

式（4）取最小值时，其关于  $HS_i^N$  的导数为零，即：

$$HS_i^N = \frac{\sum_{j \neq i} h_j \times d_{i,j}}{\sum_{j \neq i} h_j^2} \sim \sim (5)$$

这是基于最小均方误差准则的平均每跳距离估计值。在上例中，属于锚节点  $L_1$ ， $L_2$ ， $L_3$  的估计每跳距离分别是：

$$HS_1^N = \frac{2 \times 40 + 6 \times 100}{2^2 + 6^2}$$

$$HS_2^N = \frac{2 \times 40 + 5 \times 75}{2^2 + 5^2}$$

$$HS_3^N = \frac{5 \times 75 + 6 \times 100}{5^2 + 6^2}$$

## 二、基于自适应共线度阈值的锚节点选择

通过观察方程组（2）我们可以假想如果锚节点分布是线性分布时情况会怎么样。此方程组能否求出解？如果能求出解，在锚节点共线情况下，略微的测距误差就足以导致较大的位置估计

误差, 这样的定位精度能否接受? 于此我们提出一种锚节点筛选方法以过滤掉部分将导致误差偏大的锚节点。

首先介绍共线度概念:

$$DC = \frac{2\sqrt{3} \times h_{\min}}{3 \times l_{\max}} \sim (6)$$

首先指出共线度是基于三角形的, 上式中  $l_{\max}$  表示三角形最长边,  $h_{\min}$  表示与此边对应的三角形高。很容易得到共线度取值范围是【0, 1】当三点共线时, 共线度值为零; 当三角形是等边时, 共线度值为 1。

尽管通过锚节点的坐标可以很容易的计算出它们间的拓扑关系, 但是仅仅考虑锚节点间的拓扑关系是不够的, 在未知节点的位置估计过程中, 还要考虑锚节点与未知节点间的拓扑关系。因此我们设置共线度阈值来约束锚节点的选择。

通过 OMNET 的仿真 DV-Hop 算法我们得到, 未知节点与锚节点的跳距越大时, 距离估计误差越大。详细的说, 5 跳间距的距离估计误差是 1 跳间距的距离估计误差的 3 倍, 分别是节点通信距离的 0.2 倍与 0.6 倍。如此说来综合考虑设定的共线度阈值及位置节点距离锚节点三角形的最小距离阈值, 可以发现共线度阈值分别和未知节点与锚节点三角形的位置关系以及锚节点之间的拓扑关系有关。即共线度阈值并不是一个可以固定计算出的值, 它是随网络拓扑变化而变化的。因此, 锚节点共线度阈值的建立是自适应的。

我们的改进是根据 AC-HOP 算法固有的缺陷提出的, 由于 AC-HOP 算法原本与应用场景有完美的契合, 例如非基于距离的测距降低结点成本, 节点密度低, 对于低速移动结点定位等等。改进后的算法继承了这一优点, 并通过锚节点选取的改进和获得步长方法的改进, 增加了最后结果的确信度。

## 4 系统测试

### 4.1 编写目的

软件测试就是利用测试工具按照测试方案和流程对产品进行功能和性能测试, 甚至根据需要编写不同的测试工具, 设计和维护测试系统, 对测试方案可能出现的问题进行分析和评估。执行测试用例后, 需要跟踪故障, 以确保开发的产品适合需求。测试计划是描述测试目的、范围、方法和软件测试的重点等的文档。详细地测试计划可以帮助测试项目组之外的人了解为什么和怎样验证产品。如何规划整个项目周期的测试工作; 如何将测试工作上升到测试管理的高度都依赖于测试计划的制定, 测试计划成为测试工作的赖以展开的基础。

软件测试计划是指导测试过程的纲领性文件，包含了产品概述、测试策略、测试方法、测试区域、测试配置、测试周期、测试资源、测试交流、风险分析等内容。借助软件测试计划，参与测试的项目成员，尤其是测试管理人员，可以明确测试任务和测试方法，保持测试实施过程的顺畅沟通，跟踪和控制测试进度，应对测试过程中的各种变更。

## 4.2 算法测试

### 4.2.1 算法测试环境

本次测试是在 OMNET 4.0++平台下进行的测试。利用这个平台可以成功模拟设计需求中的网络环境，即低锚节点密度下的室内网络环境。

#### 1. OMNET 4.0++环境的搭建

##### 1) Java 虚拟机的安装

下载J2SDK 的安装文件：jdk-6u14-windows-i586.exe，下载之后安装好J2SDK；安装完之后，设置环境变量。

我的电脑---属性---高级---环境变量，添加变量

(假定你的jdk 安装在C:\Program Files\Java\jdk1.6.0\_14)：

JAVA\_HOME=C:\Program Files\Java\jdk1.6.0\_14

classpath=.;%JAVA\_HOME%\lib\dt.jar;%JAVA\_HOME%\lib\tools.jar;

path=%JAVA\_HOME%\bin

##### 2) OMNET++的安装

###### a) 拷贝omnet 4++文件夹到目标位置

###### b) GNU编译

```
./configure
```

```
Make
```

###### c) 执行实例

```
cd samples/dyna
```

```
./dyna
```

###### d) 运行环境

```
Omnetpp
```

###### e) 重新编译库

```
./configure
```

```
make clean
```

```
make
```

## 2. 配置网络

结点区域  $50m \times 50m$  的矩形区域

结点数量 100 个

结点密度  $0.4/m^2$ ；室内 20 平米左右的房间大概有 8 个结点，符合需求密度

结点分布 结点生成  $\pi$  分布

结点通信半径 分别为 15m, 20m, 25m, 30m, 意义在于对不同结点密度下定位仿真

综述，考察在  $50m \times 50m$  区域内随机设置 100 个节点 (包括参考节点与未知节点) 的情形，研究在不同的通信半径与参考节点比例条件下节点定位的性能。对不同节点的通信半径及参考节点比例，分别进行了 100 次网络仿真试验。

### 4.2.2 定位算法测试指标

可定位节点比例：指通过定位算法成功实现位置估计的未知节点数量占网络中所有未知节点数量的百分比。

定位误差：( localization error, LE)指的是通过定位算法得到的未知节点的估算位置与实际位置的偏差, 这种偏差可以用两者之间的欧氏距离衡量。定位误差的大小能最直接说明算法的有效性。

### 4.2.3 测试范围：

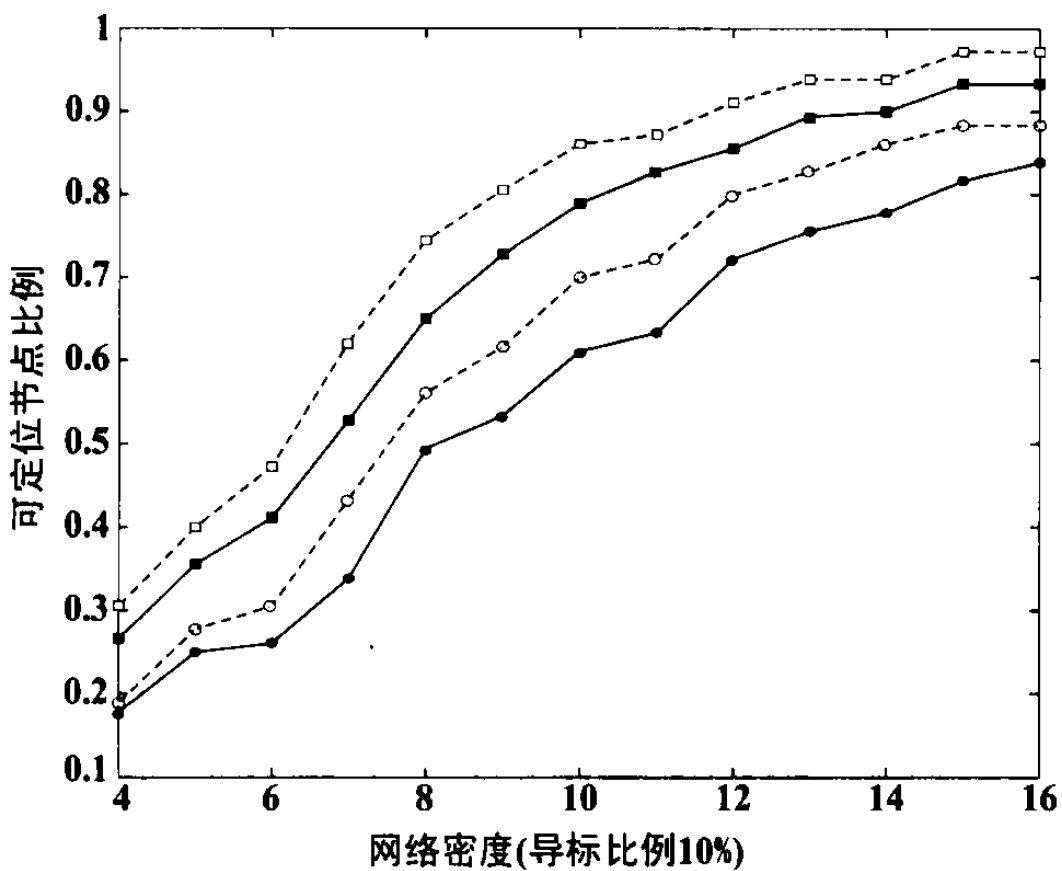
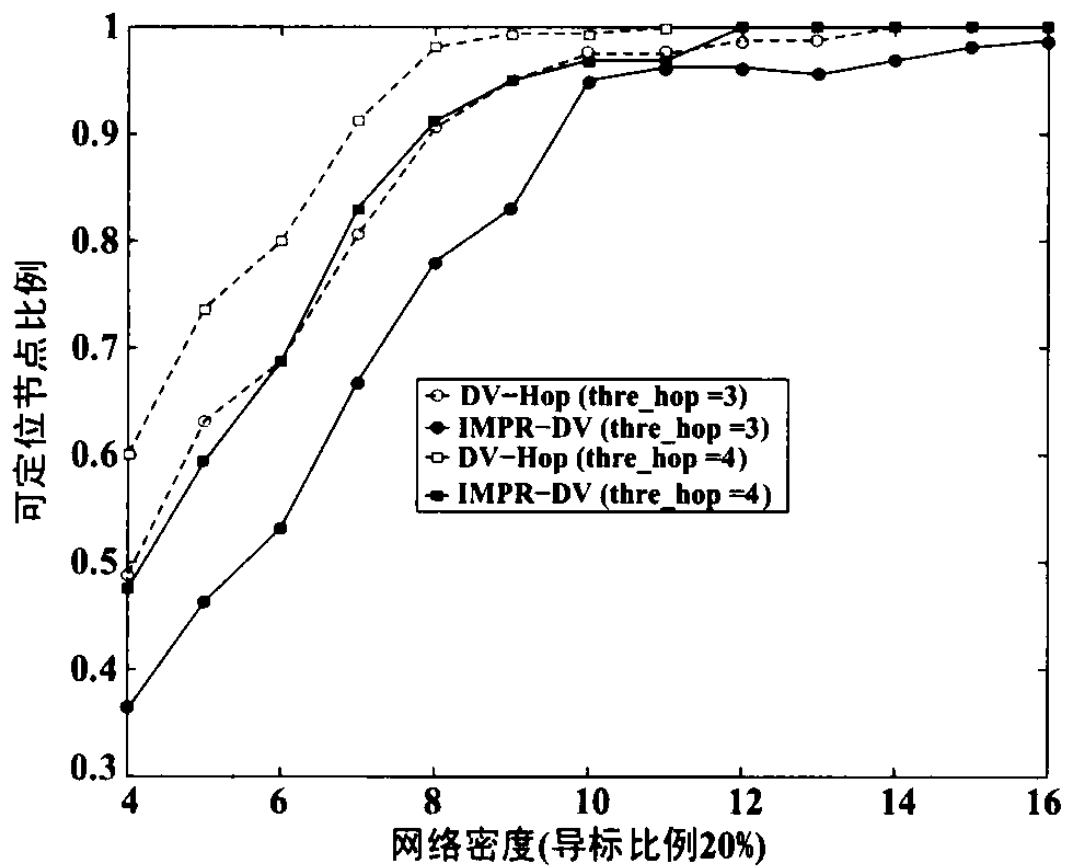
1. DV-Hop 与 IMPR-DV 在 3 个锚节点，导标比例 10% 情况下的可定位节点比例比较。
2. DV-Hop 与 IMPR-DV 在 3 个锚节点，导标比例 20% 情况下的可定位节点比例比较。
3. DV-Hop 与 IMPR-DV 在 4 个锚节点，导标比例 10% 情况下的可定位节点比例比较。
4. DV-Hop 与 IMPR-DV 在 4 个锚节点，导标比例 20% 情况下的可定位节点比例比较。
5. DV-Hop 与 IMPR-DV 在 8 个锚节点，导标比例 10% 情况下的定位平均误差比较。
6. DV-Hop 与 IMPR-DV 在 8 个锚节点，导标比例 20% 情况下的定位平均误差比较。

## 4.3 测试报告

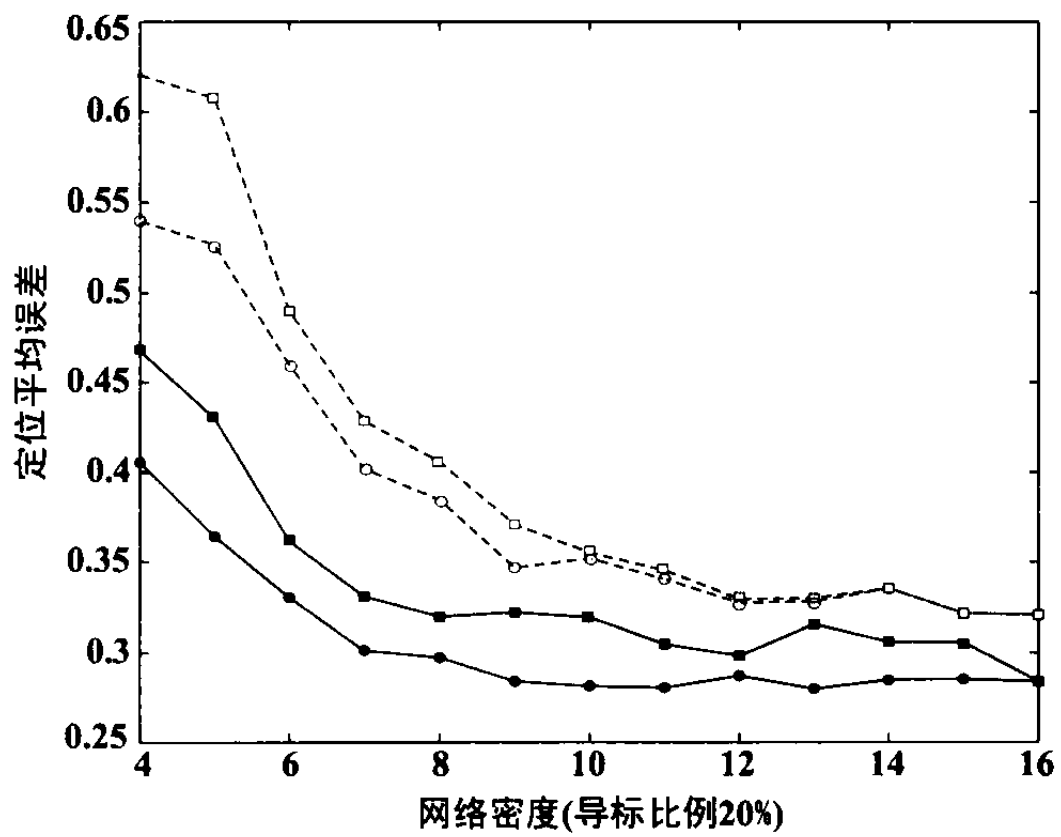
### 4.3.1 测试用例

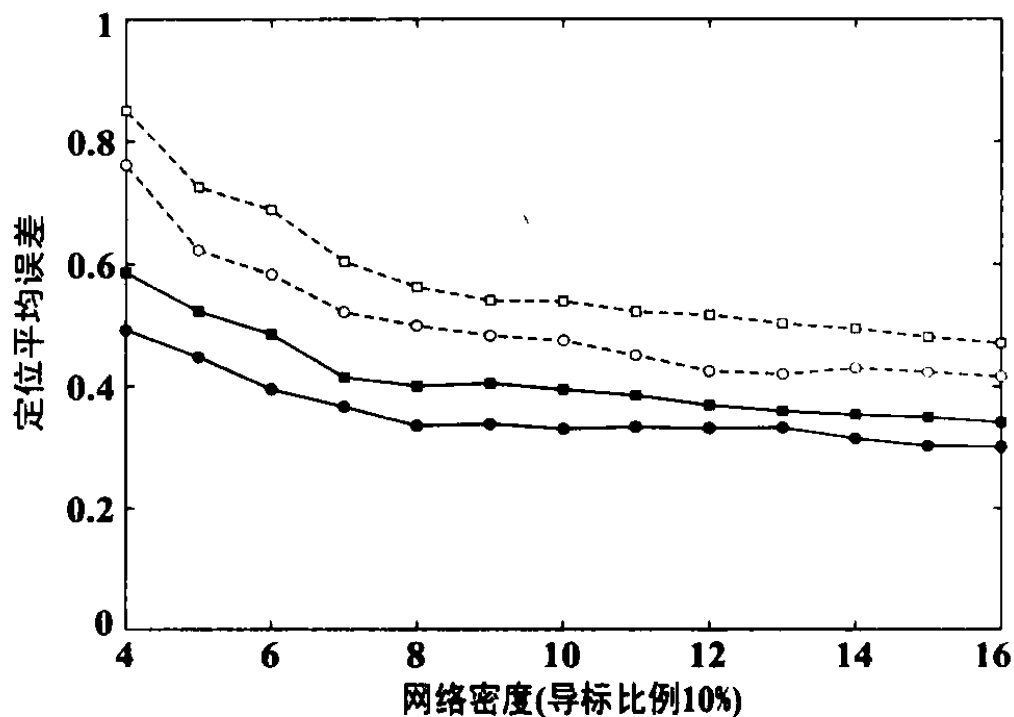
为了提高 DV-Hop 算法在随机分布无线传感器网络中的定位性能，我们对 DV-Hop 算法的距离估计和定位方面提出了采用最小均方误差准则取代无偏估计准则的策略，在锚节点选取方面引入

了共线度概念，提出基于自适应共线度阈值的锚节点选择策略。仿真实验表明改进算法在定位误差方面比传统 DV-Hop 算法更具有优势，可以明显提高无线传感器网络中的节点定位精度。



下图是定位平均误差示意图：





#### 4.3.2 测试事项

##### 4.3.2.1.1 测试进度安排

1. 制定计划：根据测试大纲、需求、说明书编制《测试计划》 1--2 个工作日
2. 安装部署环境：进行测试设计并部署、安装测试环境 2--3 个工作日
3. 设计测试用例：根据《测试计划》中人员安排，时间进度，编写测试用例 需要 4 个工作日
4. 执行功能测试用例：根据《使用说明书》、《需求规格说明书》、《测试用例》，执行测试 需要 5——7 个工作日
5. 编制功能测试问题报告：生成《测试问题报告》并通过评审，提交给开发人员 与 4. 同步完成
6. 回归测试：得到开发人员的反馈后，进行回归测试

##### 4.3.2.1.2 分析测试

可定位节点比例是指通过定位算法成功实现位置估计的未知节点数量占网络中所有未知节点数量的百分比。



定位误差 (localization error, LE) 指的是通过定位算法得到的未知节点的估算位置与实际位置的偏差, 这种偏差可以用两者之间的欧氏距离衡量。定位误差的大小能最直接说明算法的有效性。

在网络密度低于 10 锚节点比例为 20% 以及锚节点比例为 10% 的情况下, 改进算法的定位平均误差与传统 DV-Hop 算法比降低程度为 10-45%, 改进算法比传统算法在平均误差方面表现出更好的性能。

在导标节点比例为 20% 且网络密度低于 10 的情况下以及在导标节点比例为 10% 且网络密度从 4 到 16 变化的情况下, 改进算法的可定位节点比例相比 DV-Hop 低于 5~10%。

总体上, 改进的算法比原算法在平均误差方面表现出了更好的性能, 尤其是在网络密度较为稀疏及导标节点密度较小的情况下, 优越性更为明显。

总体上, 改进的算法比原算法在平均误差方面表现出了更好的性能, 尤其是在网络密度较为稀疏及导标节点密度较小的情况下, 优越性更为明显。

## 5 项目及课程总结

提到项目总结我就先介绍下我们小组的成员组成以及项目题目确定的背景吧。首先是我自己, 本科在科大学了五年的数学, 考研失败之后系里教学秘书的介绍下来到了科大软院。来苏州之前我可以说是几乎不了解软件设计, 甚至连 c, java 这样的语言都很陌生, 更不用说自己做一个软件了。而小组另一位成员曹玉斌的情况和我也是非常的相似, 区别只是他本科是在物理系学了物理。有了如此背景我们在选题时也就很自然的挑选了无线传感器网络的应用这一结合了许多算法完善与硬件设施的题目。在项目计划制定之前我们了解到学校的吕松武老师正是研究这一领域, 并且学校实验室也有相关实验器材。在请教果几位老师之后, 我们初步定下了项目的目标与阶段性的步骤。时间飞逝, 转眼到了新的一学期, 踌躇满志的我们正向着我们的既定目标迈进, 在最后测试硬件的过程中我们倍受打击, 学校实验室的传感器较低的灵敏度和较大的误差让我们无所适从。理想与实践的相差是巨大的, 我们知道了我们无法将计划里硬件部分完成下去, 于是项目变成了单纯的算法改进与仿真测试, 很难想象这是一个软件工程的题目。在这个项目的完成过程中我学到了很多, 不仅仅是技术层面的, 更多的是如何做一个软件项目, 如何才能做的最好。都说做人做事是相通的, 在经历过第一个长时间项目的不顺利后, 我受益匪浅。

在本学期学习孟老师的高级软件工程课程期间, 我脑子里想的最多的一句话就是如果是在做软件工程项目之前或者期间学习这门课, 将会有多少弯路时可以避免的, 又有多少步骤我能够做的更好。所谓智者虑远, 见微知著, 现在的后悔是毫无意义的, 想一想自己学到了什么, 又有什么地方能改进才是。学习孟老师的课有很多很多的收获, 要说最大的还是老师将我们做的软件工程项目与课程联系起来, 让我们能够非常细致的回顾了自己曾经做过的项目, 将一个我们再熟悉不过的项目完整的呈现在我们面前, 让我们体会到自己曾经做的与标准有着多大的差距, 我们应如何去完善自己的项目。高级软件工程本来是一门理论性极强的课程, 是很多沉醉于代码的牛人不怎么愿意去课堂聆听的一门课。但就是这样一门枯燥的工程被孟老师讲成了一门实用性极强, 让人有很大欲望去参与互动的一门课程。最后如果一定要提什么意见的话, 希望以后在课堂上能

引入一些非常经典的实例，让大家去分析解剖这些精粹，让我们能够站在巨人的肩膀上去遨游知识的海洋。

## 6 参考资料

1、ADAPTATION AND MOBILITY IN WERELESS INFORMATION SYSTEMS by Randy H. Katz from IEEE Personal Communications Magazine First Quarter 1994\_Volume 1, Number 1

2、END-TO-END ARGUMENTS IN SYSTEM DESIGN by J.H. Saltzer, D. P. Reed and D. D. Clark\* M. I. T Laboratory for Computer Science

3、基于最小跳数的无线传感器网络能量自适应路由算法 著：于磊磊、蔡乔林、王春雷

4、无线传感器网络方向性分区路由算法 著：刘云路、柴乔林、赵晋

5、蚁群网络中后效问题的研究 著：王新伟、刘春雨

6、鲁棒的机器人门特卡罗定位算法 著：武二永、项志宇、刘济林

Simulation Wireless Sensor Networks with OMNeT++ by C. Mallanda, A. suri, V. Kunchakarra, S. S. Iyengar, R. Kannan and A. Durresi Sensor Network Research Group, Department of Computer Science, Louisiana State University, Baton Rouge, LA