

一种基于成簇优化的无线传感器网络非均匀分簇算法

陈 涛¹ 罗永健¹ 肖福刚¹ 史德阳¹ 张 莎²

(西安通信学院 西安 710106)¹ (新疆军区联勤部指挥自动化工作站 乌鲁木齐 830002)²

摘 要 针对大规模无线传感器网络中靠近基站的簇头需要转发大量数据而能量消耗较快的特点,提出了一种基于成簇优化的非均匀分簇算法。该算法通过对构建的传感器网络模型进行非均匀分层,然后各层独立展开簇的组建来实现非均匀分簇,在簇头选举阶段依据节点剩余能量及节点离层中间线距离的加权和优化簇头选举方法。分簇完成后,簇头调整通信距离,构建簇头间动态的多跳路由。仿真实验结果表明,新算法生成的簇头数目稳定,拓扑结构合理,与 LEACH 算法和 EEUC 算法相比能较好均衡簇头的能耗,延长网络寿命。

关键词 无线传感器网络,非均匀分簇,簇头,梯度

中图法分类号 TP393 文献标识码 A

Uneven Clustering Algorithm Based on Clustering Optimization for Wireless Sensor Networks

CHEN Tao¹ LUO Yong-jian¹ XIAO Fu-gang¹ SHI De-yang¹ ZHANG Sha²

(Xi'an Communications Institute, Xi'an 710106, China)¹

(Xinjiang Military Area Command Automation Workstation, Urumqi 830002, China)²

Abstract In consideration of the characteristic that cluster heads close to the base station consume the larger energy in wireless sensor networks using multi-hop communication model, an uneven clustering algorithm based on clustering optimization was presented in this paper. The constructed network model is divided into various non-uniform layers, and then each layer completes unequal clustering by clustering independently. During the course of cluster heads election, the proposed algorithm optimizes the cluster heads election method according to the weighting sum of the node's residual energy and distance to layer's middle line. After completion of clustering, cluster heads adjust their communication distance in order to build dynamic multi-hop routing among cluster heads. Simulation results show that the proposed algorithm can generate the stable number of cluster heads and reasonable topology, and also prolong the network lifetime by balancing the energy consumption of nodes effectively compared with LEACH and EEUC.

Keywords Wireless sensor network, Uneven clustering, Cluster heads, Grade

1 引言

通过均衡无线传感器网络中节点的负载来延长网络寿命,是无线传感器网络研究的重要课题之一,其常用方式是分簇方法。分簇方法是把整个网络节点分成不同的簇,然后选举簇头对簇内节点进行管理维护,并周期性进行簇头轮换和簇重组。由于无线传感器节点通信距离小,通常无法直接和基站进行通信,信息必须通过多跳路由才能传递到基站。在多跳通信模式下,靠近基站的节点由于要传递外侧节点向基站发送的信息,会消耗相对较多的能量,如果担任簇头节点,还需接收融合簇内其他节点的数据,造成更大能量消耗。这些节点极有可能过早耗尽自身能量而死亡,产生由网络分割引起的“热区”问题。

为了解决无线传感器网络多跳路由中的“热区”问题,So-ro 等提出了非均匀分簇的思想^[1]。该文中所假设的网络拓扑是环绕汇聚点的两层同心圆环,内圆环中的簇头由于靠近

汇聚点,需要承担数据转发的任务,因此通过减小所在簇的节点数来降低簇内处理中消耗的能量,以便为簇间数据转发预留更多能量。但文中所考虑的网络模型是一个异构网络,簇头作为超级节点,将自始至终担任簇头,并且位置是事先计算好的,故该方法对于一般网络的适用性较差。李成法等文献^[2]中提出了一种能量高效的非均匀分簇算法(EEUC),该算法首先按一定概率选出部分节点成为候选簇头,各候选簇头依据其与汇聚节点的距离设定竞选半径,再通过比较候选簇头的剩余能量确定最终的簇头,但是该算法只是尽量平衡能耗,并未从理论上保证能耗的均衡性,且没有给出参数的最优求解方法^[3]。文献^[4]对 EEUC 算法中的网络分层模型进行了改进,但是在成簇阶段网络中部分节点有可能会被遗漏,算法成簇效果不能保证,且未详细提及簇头选举方法。文献^[5]提出的多跳非均匀分簇算法在簇头选举阶段仍采用 LEACH 算法^[6]中随机产生簇头的方法,这种算法无法解决可能出现的簇头分布集中、分簇不合理的情况。为此,本文提

本文受国家自然科学基金面上项目(61179002,61102160),陕西省自然科学基金基础研究计划资助项目(2011JM8030)资助。

陈 涛(1988—),男,硕士生,主要研究方向为无线传感器网络,E-mail:953284937@qq.com;罗永健(1971—),男,博士,教授,主要研究方向为阵列信号处理、雷达目标识别及多用户通信等。

出了一种基于成簇优化的非均匀分簇算法。

2 无线传感器网络模型

采用与多数非均匀分簇算法相似的网络模型。考虑一个有 N 个传感器节点随机分布在一个 $L \times L$ 的正方形区域 A 内,其应用场景为周期性的数据收集,用 s_j 表示第 j 个节点,相应的节点集合为 $S = \{s_1, s_2, \dots, s_N\}$ 。模型假设如下:

(1) 所有节点的功能相同,具备数据融合的功能,每个节点都有一个唯一的身份标识(ID)。

(2) 基站位于正方形观测区域 A 的外侧,传感器节点和基站在部署后不再发生移动。

(3) 对于不同距离的接收者,节点可以自由调整其发射功率以减小能量消耗。

(4) 链路是对称的,若已知对方发射功率,节点可以根据接收信号的强度(RSSI)计算出发送者到自己的近似距离[7]。

文中采用与文献[6]相同的传输能量消耗模型。数据发送能量消耗由发射电路损耗和功率放大损耗两个部分组成,即

$$E_{TX}(l, d) = \begin{cases} lE_{dec} + l\epsilon_{fs}d^2, & d < d_0 \\ lE_{dec} + l\epsilon_{mp}d^4, & d \geq d_0 \end{cases}$$

其中, l 表示发送的比特数, d 表示发送距离, ϵ_{fs} 和 ϵ_{mp} 表示不同距离范围内功率放大所需的能量, d_0 为判断采用自由空间模型或多路径衰减模型的临界点[8]。数据接收消耗的能量为

$$E_{RX}(l) = lE_{dec}。$$

3 算法设计

新算法首先根据节点到基站的距离对节点进行分层,接着进行簇头选举和簇的组建。在簇头选举阶段优化了簇头选举方法,每个节点根据自己与所在层中间线的距离和自身剩余能量计算出一个加权和,然后与接收到的其他节点的加权和进行比较。如果都小于自身的加权和,则宣布自己当选簇头。每层节点经过两个批次即可完成簇头选举,非簇头节点选择距离最近的簇头加入。簇组建完成之后,簇头及时调整通信距离,并对相邻层簇头发布公告消息,构建层间多跳路由。最后,在规定时间内簇头的数据通过多跳路由汇聚至基站。

3.1 对节点分布区域分层

网络分层模型如图1所示。基站与网络分布区域的距离已知,记为 d_{min} ,又知道网络分布区域为 $L \times L$ 的正方形,可以计算出最远处节点与基站的距离 d_{max} , $d_{max} = ((d_{min} + L)^2 + (L/2)^2)^{1/2}$ 。每层圆环的上界值和下界值分别用 UC_i 和 LC_i 表示,每层的间距为 d_i ,层的中间线距离基站的距离为 $L_{i, cen}$, $L_{i, cen} = (UC_i + LC_i)/2$ 。节点的最大通信半径记为 R_c^0 。分层以基站为圆心,以 d_{min} 为最小半径,以 d_{max} 为最大半径,分层数目 k 由基站决定。 d_i 的大小受以下条件制约:

$$d_{min} + d_1 \leq R_c^0 \quad (1)$$

$$d_k > d_{k-1} \cdots d_2 > d_1 \quad (2)$$

$$d_k + d_{k-1} \leq R_c^0 \quad (3)$$

$$d_{min} + d_1 + d_2 + \cdots + d_k = d_{max} \quad (4)$$

网络层次划分完毕后,基站把每层上下界值、层间距以及层数目用 HELLO 消息向全网广播,消息格式如下:

UC_1, LC_1, d_1	UC_2, LC_2, d_2	$\dots$	UC_k, LC_k, d_k	k
-------------------	-------------------	---------	-------------------	-----

通过接收广播消息,每个节点根据接收到的信号强度计算自身与基站的距离 $S_{j, dist-to-BS}$,进而确定自己属于哪一层。此外,节点还可以计算自己离所在层中心线的距离 $S_{j, dist-to-center}^{i, layer} = abs(S_{j, dist-to-BS} - L_{i, cen})$,其中, S_j^i 表示 ID 号为 j 的节点处在第 i 层。处在第 i 层的节点,以通信半径 d_i 来竞选簇头。

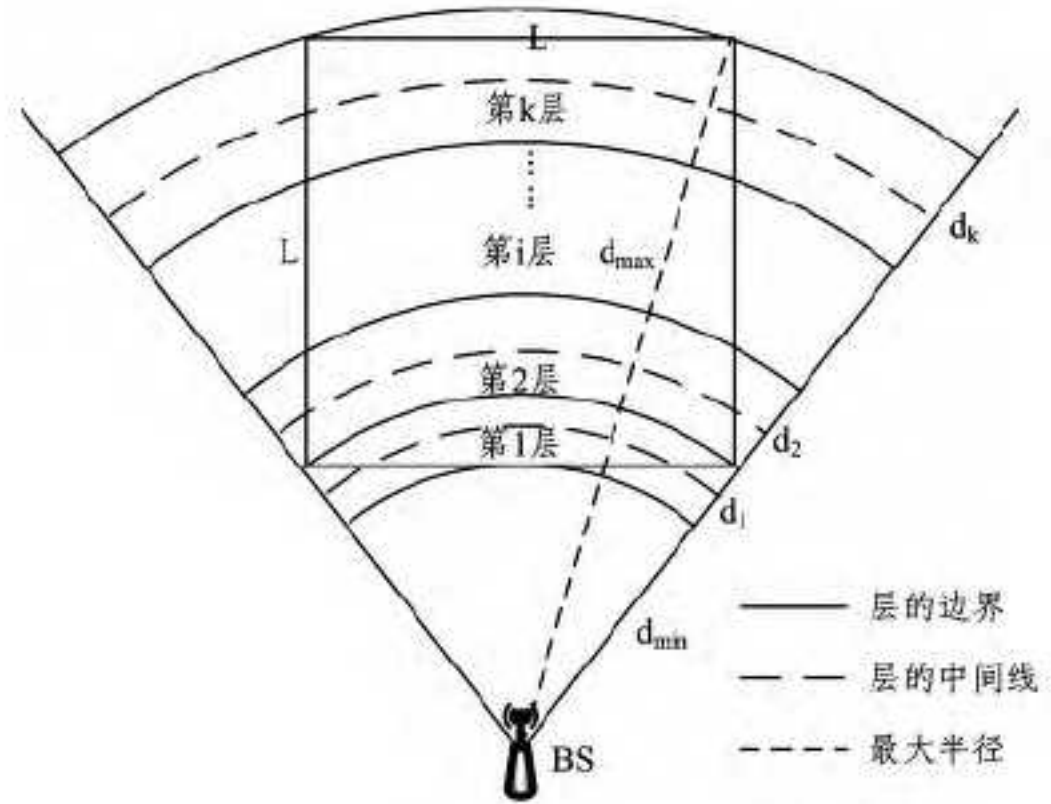


图1 网络分层模型

3.2 簇头的选择和簇的形成

各节点确定自己所在层之后,各层展开簇头的选举。在成簇阶段采用基于 CSMA/CA 的 MAC 层协议,分簇完成之后采用基于 TDMA 的混合 MAC 接入协议,并遵循簇头的时序调度[6]。成簇过程大致分为如下步骤:

(1) 节点计算自己竞选簇头的参数 p 。 p 是节点剩余能量 $E_{residual}$ 及节点离所在层中间线距离 $S_{j, dist-to-center}^i$ 的加权和,以节点 j 为例:

$$p_j = \lambda \frac{E_{residual}}{E_{max}} + (1 - \lambda) \left(1 - \frac{S_{j, dist-to-center}^i}{d_i/2}\right)$$

其中, λ 为平衡参数,作用是使能量成为主导因素,取值范围为 $(0.5, 1)$ 。然后,节点以通信距离 d_i 广播簇头竞选消息,内容包括 ID 号 j 、所在层 i 、剩余能量 $E_{residual}$ 、距离中间线距离 $S_{j, dist-to-center}^i$ 及加权和 p_j 。接收到其他节点的簇头竞选消息时,只处理本层节点的消息,非本层节点的消息不处理。

(2) 如果节点接收到的簇头竞选消息中的 p 值都小于自身的 p 值,则节点当选簇头,并广播自己成为簇头的声明消息;若有节点 p 值大于自身的 p 值,则暂时放弃竞选簇头,等待簇头发来的当选声明。当选为簇头的节点以通信距离 d_i 广播当选声明,收到簇头当选声明的节点本轮放弃竞选簇头。

(3) 一般情况下,第(2)步完成之后,会存在部分节点没能收到簇头的当选声明,在文献[4]中并没有考虑到这种情况。在图2中节点B的 p 值仅小于节点A的 p 值,箭头所指区域内的节点因收到节点B的簇头竞选消息,且 p 值都小于节点B的 p 值,所以都暂时放弃竞选簇头,而节点B由于收到节点A的簇头竞选消息,也不能成为簇头,最终致使图2中箭头所指区域节点没有收到簇头的当选声明。所以,如果节点在一定时间 T 之后,仍没有收到簇头的当选声明,则再次广播竞选簇头消息,重复第(2)步,完成分簇。每层的节点经过两个

批次即可完成簇头选举。

(4) 如果节点收到多个簇头的当选声明,选择距离自己较近的簇头加入。如图3所示,箭头所指区域内节点会收到两个簇头的当选声明,节点选择距离较近的簇头加入。

(5) 一轮结束之后,下一轮按照(2)–(4)步重新组簇。

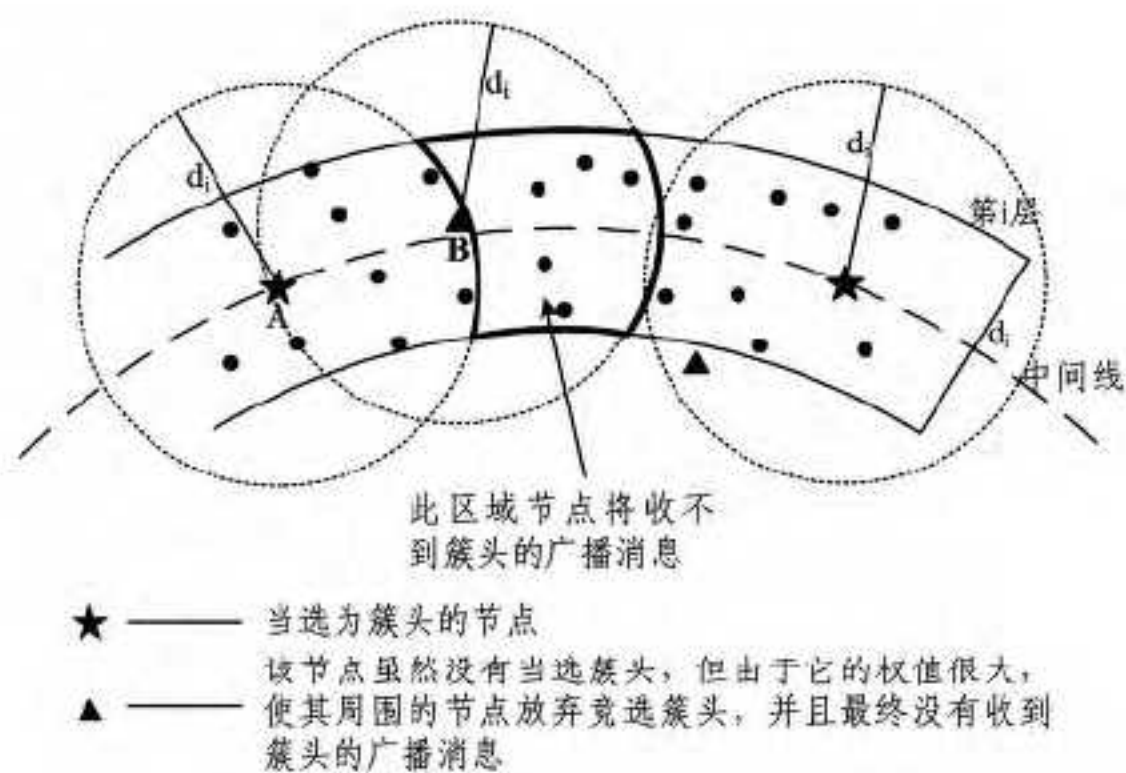


图2 部分没有收到簇头广播消息的节点

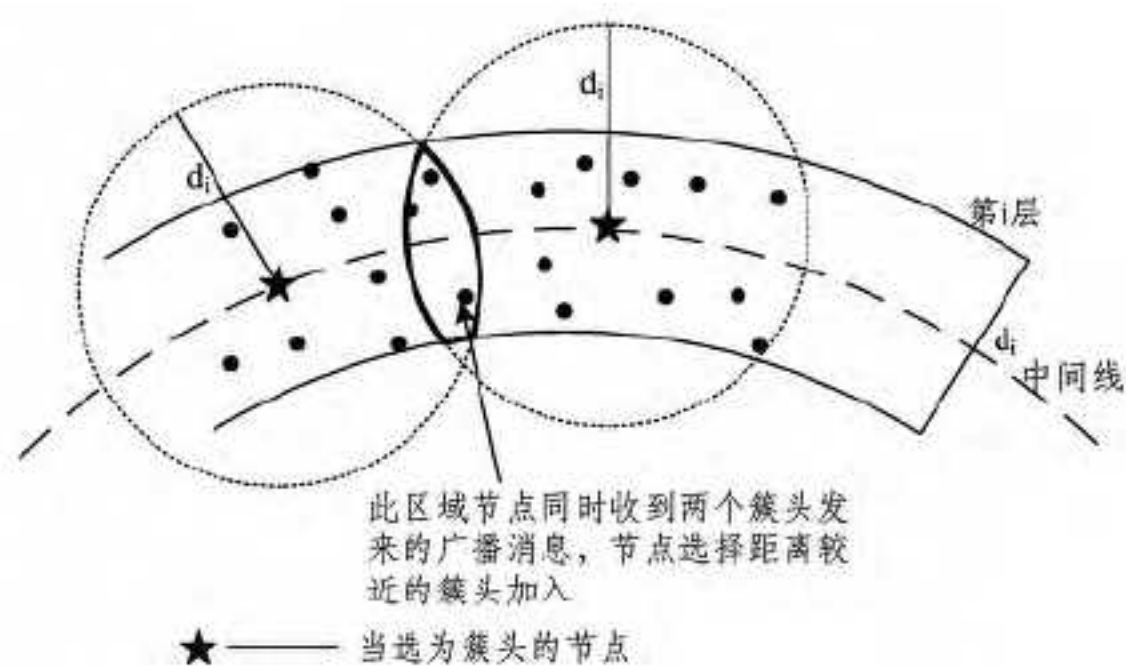


图3 部分收到多个簇头广播消息的节点

3.3 簇内路由和簇间路由

簇内因各节点间的距离较小,以单跳模式实现路由^[9],即簇成员节点直接发送数据给簇头节点。簇间以多跳进行数据的传输。内层的簇头只要接收到外层的数据,就会向基站方向传递,直到通过第一层的簇头传递给基站。多跳路由的构建方法如下:分簇完成之后,各簇头调整通信距离为 $d_i + d_{i+1}$,使相邻层的簇头都能收到组建路由的广播消息,同时将接收到的其他簇头的消息存储起来,这样每个簇头都存储了其邻居层簇头的信息,可以完成邻居簇头路由表的构建。簇间传递数据时,簇头再次调整通信距离为 $S_{j \text{ dist} \rightarrow \text{BS}} - LC_{i-1}$,只需保证第 $i-1$ 层的簇头能够接收到数据即可。第 i 层簇头选择在自己通信距离范围内且能量较大的第 $i-1$ 层簇头传递自己以及接收到的外层簇头的数据包。

4 算法仿真分析

算法性能仿真利用 Matlab 工具。为简单起见,整个仿真过程中假定采用理想的 MAC 协议,忽略了无线链路中可能发生的丢包错误。

4.1 仿真参数设置

模拟环境为: $L=200\text{m}$, $n=100$, 基站 BS 坐标为 $(100, -50)$, 网络共分为 3 层, $d_1=50\text{m}$, $d_2=75\text{m}$, $d_3=95\text{m}$, 节点的最大通信半径 R_c^0 满足仿真要求。 λ 取经实验推导出的最优值 0.8。仿真的参数与文献^[6]中基本相同,如表1所列。

表1 仿真参数

参数名称	取值
网络覆盖区域	$200\text{m} \times 200\text{m}$
基站位置	$(100\text{m}, -50\text{m})$
节点初始能量	0.5J
节点总数	100
d_0	88m
E_{elec} (发射电路损耗能量)	50nJ/bit
E_{DA} (融合单位比特数据所需能量)	5nJ/bit
ϵ_{fs} (自由空间模型功率放大所需能量)	10pJ/(bit · m ²)
ϵ_{mp} (多路径衰减模型功率放大所需能量)	0.0013pJ/(bit · m ⁴)
控制包大小	100bits
数据包大小	4000bits

4.2 仿真结果分析

为了验证本文算法的性能,从以下两个方面与 LEACH 算法和 EEUC 算法进行比较。

(1) 分别对 LEACH 算法、EEUC 算法和新算法随机抽取 100 轮,统计各轮的簇头数目,其中 LEACH 算法的 P 值为 0.05^[10]。图4—图6显示3种算法的统计结果, X 轴表示簇头数目, Y 轴表示轮数统计结果。LEACH 分簇算法采用随机数的方法产生簇头,生成簇头的数目变化较大,很不稳定。虽然 5% 是 LEACH 算法中簇头占全网节点总数的最佳百分比,但是产生 5 个簇头的轮数却不占总轮数的 20%。EEUC 算法生成簇头的数目要比 LEACH 算法稳定些,但是与文中算法相比簇头波动范围较大。图7为新算法随机一轮的拓扑结构图,用星号标记的节点表示当选为簇头的节点。由该图可以看出,本文算法生成的拓扑结构较为合理。

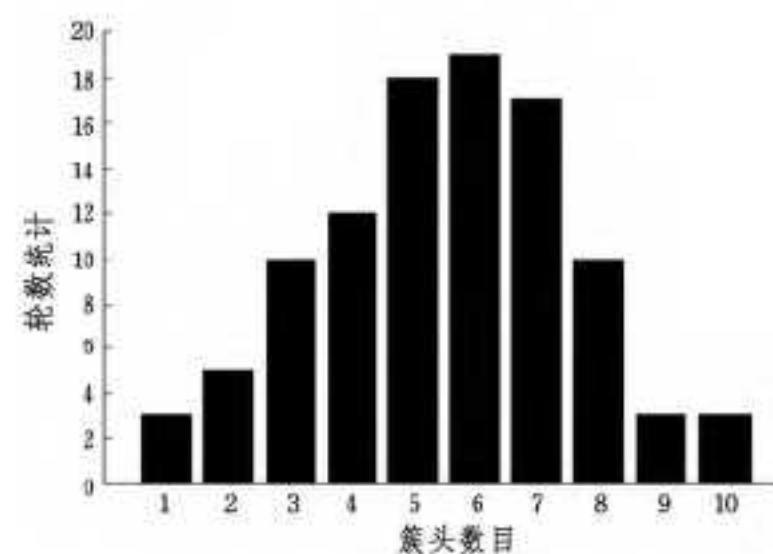


图4 LEACH 算法产生的簇头数目分布统计

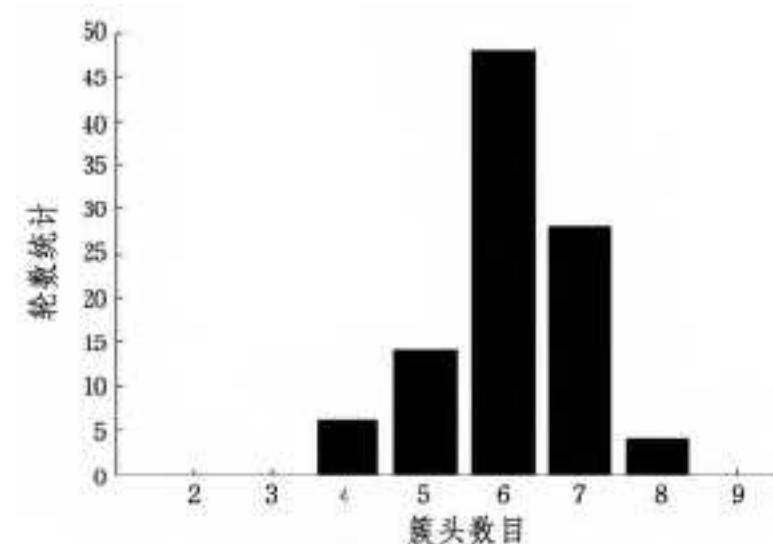


图5 EEUC 算法产生的簇头数目分布统计

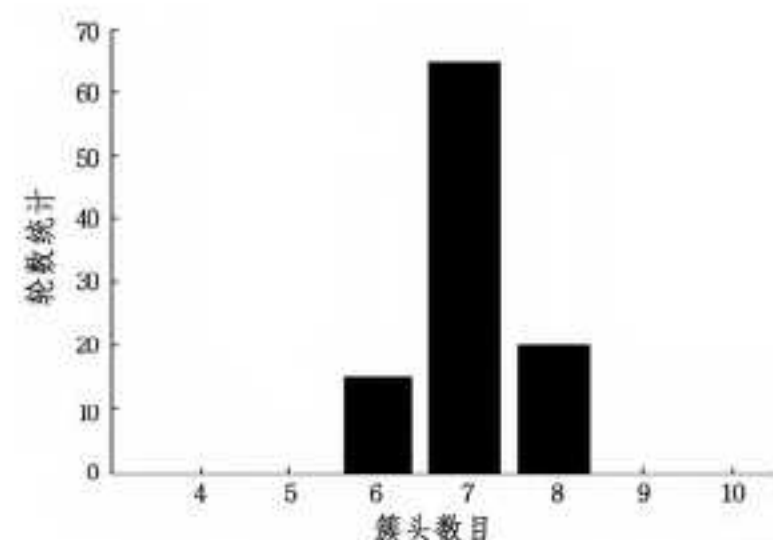


图6 本文算法产生的簇头数目分布统计

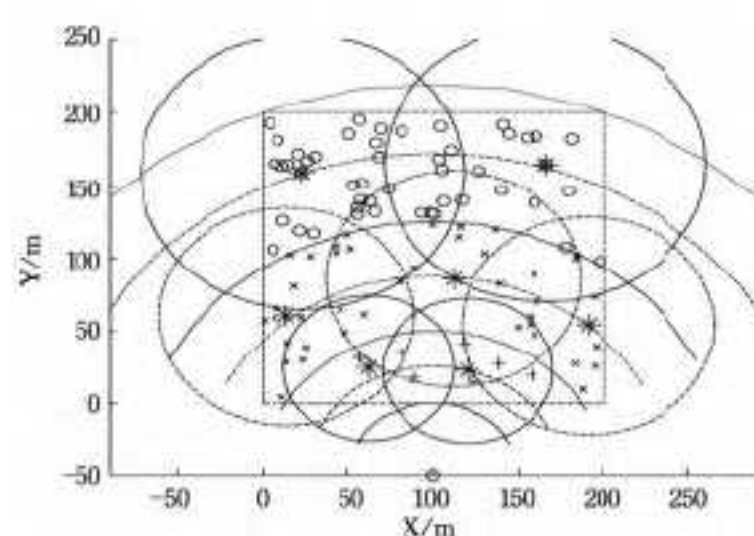


图7 本文算法随机一轮生成的拓扑结构图

(2)在随机生成的一种节点分布模型下,3种分簇算法的网络生存时间如图8所示。从图可以看出,新算法延长了网络的生存时间。在簇头选举时,新算法先对节点分层,然后在每一层各自展开簇头的选举,选举方法比EEUC算法简单高效,减小了能耗。

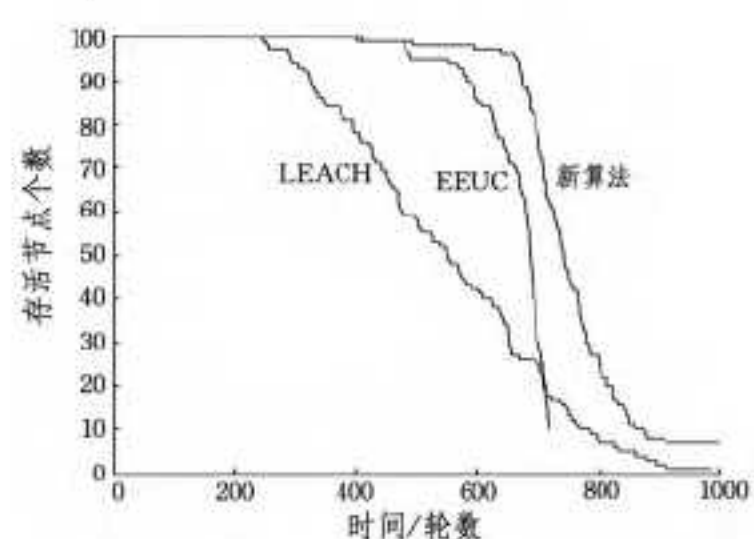


图8 3种分簇算法的网络存活时间比较

结束语 本文研究了无线传感器网络接近实际的一种网络模型,通过对网络非均匀分层并且将各层节点组织成不同规模的簇,使靠近基站的节点簇规模较小,以此来达到均衡簇头能耗的效果。文中重点优化了簇头选举方法,使得新算法运行简单,产生的拓扑结构更为合理。仿真实验证明新算法能降低成簇开销,均衡网络节点能耗和延长网络寿命,是解决

(上接第288页)

算法误差率在700时达到40%。本文算法的效率和精度比传统的DV-Hop算法要高。当节点的数量达到400时,锚节点所占比例增加,结果提高了算法的平均定位误差率,如图5所示。通过图5可知,相比DV-Hop定位算法,本文算法的误差率减小了。

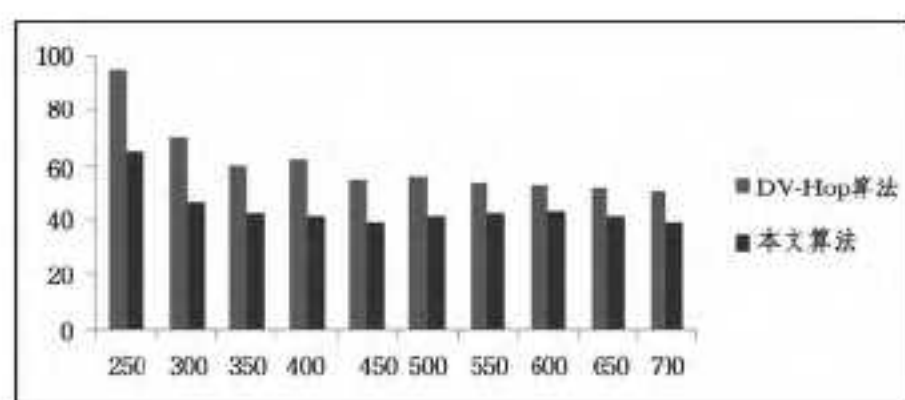


图5 锚节点平均分布的平均定位误差

结束语 基于一个随机行走的非监督变化检测方法是一个典型的变化检测方法。该方法的特点是提取遥感图像的关键点作为一个随机行走的初始种子点,使用一个策略来减小像素提取的不透明度。它和特征点的图像配准变化检测相结合,提高了变化检测的效率。本文算法适用于现实网络,具有定位误差小、范围广等优点,比传统的DV-Hop定位算法具有更好的精度,是一种高效的定位算法,具有较强的应用价值。

参考文献

[1] 江山. 有关计算机网络路由的分析与探讨[J]. 科技资讯, 2012

无线传感器网络多跳通信中“热区”问题的一种有效方法。

参考文献

- [1] Soro S, Heinzelman W. Prolonging the lifetime of wireless sensor networks via unequal clustering [C] // Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, San Francisco: IEEE Computer Society Press, 2005: 236-240
- [2] 李成法, 陈贵海, 叶懋等. 一种基于非均匀分簇的无线传感器网络路由协议[J]. 计算机学报, 2007, 30(1): 27-36
- [3] 刘志. 无线传感器网络中的能量高效覆盖与路由算法研究[D]. 北京: 北京交通大学, 2011
- [4] 董杨, 郭拯危, 王青正. 一种基于非均匀分簇的WSN路由协议[J]. 郑州轻工业学院学报, 2009, 24(6): 70-74
- [5] 李超良, 胡春华. 无线传感器网络中面向动态多跳的非均匀分簇路由[J]. 中南大学学报, 2011, 42(7): 2048-2053
- [6] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-efficient Communication Protocol for Wireless Microsensor Networks[C] // Proceedings of the 33rd Hawaii International Conference on System Sciences, Maui, Hawaii, USA: IEEE Computer Society, 2000
- [7] 马慧颖. 无线传感器网络路由协议研究[D]. 哈尔滨: 哈尔滨工程大学, 2008
- [8] 黄琛, 房鼎益, 陈晓江. 传感器网络中基于非均匀分簇负载均衡路由算法[J]. 计算机应用研究, 2009, 26(9): 3475-3477
- [9] 林观康, 程良伦. 基于地理信息静态分簇的无线传感器网络路由算法[J]. 计算机应用与软件, 2011, 28(2): 37-39
- [10] Heinzelman W, Chandrakasan A, Balakrishnan H. An Application-Specific Protocol Architecture for Wireless Microsensor Networks[J]. IEEE Transactions on Wireless Communications, 2002, 1(4): 660-670
- [28] 曹付元. 面向分类数据的聚类算法研究[D]. 太原: 山西大学, 2010
- [3] Ghosh A K, Michael C, Schatz M. A real-time intrusion detection system based on learning program behavior [C] // Algorithms for Finding Paths with Multiple Constraints Networks. 1984: 95-116
- [4] 于兴敏, 彭学平, 林培芳. 现代预分解窑设计产量初探 [J]. 中国建材装备, 1998(4)
- [5] Celik T. Unsupervised change detection in satellite image using principal component analysis and k-means clustering [J]. IEEE Geoscience and Remote Sensing Letters, 2009, 6(1): 33-37
- [6] Rouhollah D, Shohreh K. Change detection in optical remote sensing images using difference-based methods and spatial information [J]. IEEE Geoscience and Remote Sensing Letters, 2010, 7(1): 215-219
- [7] Kulpers R, Van Mlegheem P, Korkmaz T, et al. An overview of constraint based path selection algorithms for Qos routing [C] // IEEE Communications Magazine, December 2002
- [8] Korkmaz T, Krunz M. A randomized algorithm for finding a path subject to multiple Qos requirements [J]. Computer Networks, 2001, 36: 251-268