

## 基于 PEGASIS 算法的一种改进方案

胡峻浩, 刘兴长, 谈昨非

(解放军后勤工程学院 后勤信息与军事物流工程系, 重庆 401311)

**摘 要:** 采用禁忌算法优化拓扑结构, 并提出在出现节点死亡时, 选择拓扑重构是全部节点参与还是直接将死亡节点在链上前两个存活节点直接连接. 同时, 在簇头选择时, 提出了一种综合考虑剩余能量最少节点在下一轮能耗、节点剩余能量和节点到基站距离的簇头选择算法, 在数据传输阶段, 比较节点到下一跳节点的距离和节点到簇头节点的距离, 选择距离较短的路径. 仿真结果表明, 改进算法较 PEGASIS 算法第一个节点死亡时间延长了 4.9 倍, 半数节点的死亡时间也得到了延长, 从而提高了整个网络的生存时间.

**关键词:** 无线传感器网络; PEGASIS 算法; 距离阈值; 生存时间

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2014)11-0036-05

## An Improved Scheme Based on PEGASIS Algorithm

HU Jun-hao, LIU Xing-chang, TAN Zuo-fei

(Logistic Engineering University of PLA, Chongqing 401311, China)

**Abstract:** This paper uses tabu algorithm to optimize the topology and puts forward the methods which is choosing topology reconstruction is involved in all nodes or directly to connect the surviving nodes before and after death node on the chain according to whether death nodes is on the long chain, in the event of node dies. At the same time, when cluster leader is choosing, the paper presents a algorithm which is comprehensive consider that the energy consumption of minimum residual energy nodes in the next round, residual energy of nodes and the distance to base station, in data transmission phase, compares the distance of the nodes to the next hop node with the distance of the node to the cluster head nodes, chooses the shorter path. Simulation results show that the lifetime of the first dead node is more than 4.9 times of such node in PEGASIS, the death time of half of the nodes is also longer, so as to improve the survival time of the whole network.

**Key words:** wireless sensor network; PEGASIS algorithms; Distance threshold value; survival time

### 1 引言

目前分簇路由算法<sup>[1-2]</sup>是无线传感器网络路由算法研究的热点, 其中最著名为 LEACH<sup>[3]</sup>算法, 基本思想是网络周期性地选举簇头节点, 将网络划分为若干个簇. 各个节点计算到各个簇头的距离, 选择加入距离自己最近的簇, 数据传输阶段, 节点将自己感知到的数据发送给簇头, 然后由簇头将本簇内的数据融合后转发给基站. PEGASIS<sup>[4]</sup>算法是在 LEACH 算法的基础上改进而来, 算法开始之前, 每个节点需要知道其他节点的地理位置, 通过贪婪算

法将网络中的所有节点形成一条链, 节点轮流充当簇头与基站通信. 数据传输阶段, 节点在链上, 只需要与距离自己最短的节点进行通信, 从而减少了节点之间通信能耗.

### 2 PEGASIS 算法的基本思想

PEGASIS 算法, 借鉴了 LEACH 算法的成簇思想, 该算法是将网络中的所有节点通过贪婪算法组织成一条链, 数据传输时, 由簇头发令令牌, 令牌沿着链传递, 当节点收到令牌时, 节点就将收到上一个节点传送来的数据与自身的数据融合, 再发送给

下一个节点,最后簇头将融合之后的数据发送给节点.虽然 PEGASIS 算法通过链式结构极大地减少了节点的能量消耗,极大地延长了网络生存时间.但是 PEGASIS 算法依然存在以下不足:(1)PEGASIS 建链方法是基于贪婪算法,在建链的后期,由于已经加入链的节点和死亡的节点都不被再次访问,所以相邻节点间的长链不可避免,这样就增加了位于长链上节点的能耗;(2)簇头节点采用轮流担任的方式将导致远离基站的节点因能量过度消耗而率先死亡;(3)每当出现节点死亡后,都需要重新建链,从而增加了网络的通信能量开销.

针对 PEGASIS 算法存在的问题,国内外学者展开了相应的研究,文献[5]中通过在建链阶段设置距离阈值,防止了长链的产生.文献[6]和文献[7]中分别采用模拟退火算法、遗传算法,以距离的平方和为目标函数,优化了网络拓扑结构,避免了长链的产生.文献[8]中在簇头选择时采用模糊控制,以节点的剩余能量与节点到基站的距离为模糊化对象,优化簇头选择过程.文献[9]中簇头的选择考虑了节点到基站的距离和所有节点到基站距离的平均值的比值,以及节点剩余能量与节点初始能量的比值,通过设置两个参数不同的权值,适应了不同的网络要求.文献[10]为了防止长链的产生,在建链阶段采用禁忌算法优化链结构,在簇头选择时,只考虑节点的剩余能量,仿真结果表明,网络的生存时间得到了很大的延长.

以上研究都没有考虑节点死亡时,网络重构所带来的节点的额外开销,针对这个问题本文提出了通过设置阈值,判断节点是否处在长链中,当出现节点死亡时,动态地选择网络重构方式,同时在簇头选择时,考虑节点剩余能量和节点到基站的距离以及剩余能量最少节点在下一轮的能耗.

### 3 改进算法的设计

#### 3.1 算法假设

考虑一个由  $N$  个随机部署的传感器节点形成网络,周期性地采集数据.并基于以下假设:

- (1)所有节点的初始能量相同,且为同构节点;
- (2)节点随机分布,且不能移动;
- (3)基站位置固定;
- (4)基站可以获知节点信息;
- (5)基站的能量由额外的电站供给.

#### 3.2 算法的能量模型

在 PEGASIS 算法中,能量的耗损主要在数据

接收、数据融合、数据发送三个方面<sup>[6]</sup>.

本文采用比较常用的一阶无线能量模型.节点收发数据的能耗模型公式<sup>[3]</sup>如下:

$$E_{TX}(k, d) = E_{TX-elec}(k) + E_{TX-amp}(k),$$

$$E_{TX}(k, d) = E_{elec} * k + \epsilon_{amp} * k * d^2,$$

$$E_{RX}(k, d) = E_{elec} * k.$$

式中,  $k$  表示发送或接收数据的长度,  $d$  表示发送距离,  $E_{elec}$  表示无线收电路消耗的能量,  $\epsilon_{amp}$  表示信号放大器消耗能量的比例系数,其中  $E_{elec}$  和  $\epsilon_{amp}$  由硬件的情况决定.

### 3.3 算法的描述

#### 3.3.1 建链阶段

PEGASIS 算法初始建链阶段采用贪婪算法建链,容易产生长链,拓扑结构如图 1 所示.

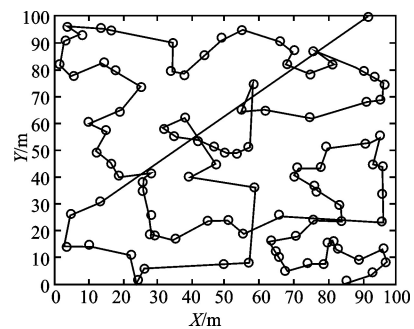


图 1 贪婪算法拓扑图

为了防止网络在建链阶段产生长链,本文采用禁忌搜索算法优化链结构<sup>[10]</sup>.以节点之间距离平方和为目标函数,贪婪算法所产生的链为初始解,任意交换链上两个节点为一个移动,设置迭代次数为 100.所产生的拓扑结构如图 2 所示.

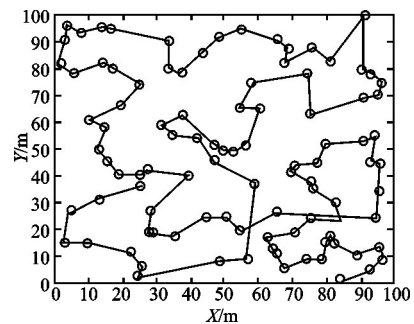


图 2 禁忌算法拓扑图

如图 2 所示,当网络中出现死亡节点时,为了节约网络拓扑重造成的能量消耗,考虑直接将死亡节点在链上的前后两个存活节点连接,可以有效地减少建链带来的额外能耗,但是这样可能会造成网络中长链如图 3 所示.但是处在长链中的节点

死亡时,由于节点本来就处于长链中,直接将节点前后两个存活的节点连接,出现长链的可能性更大,这样就造成这部分节点的过早死亡。

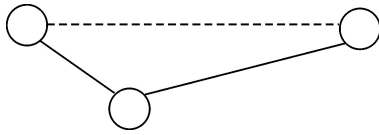


图3 长链示意图

为了解决这个问题,设置距离阈值,  $d_{\text{threshold}} = i * x_m$ , 其中  $i$  为比例系数  $0 < i < 1$ ,  $x_m$  为网络所在区域横坐标的长度。

每当网络中出现节点死亡时,判断节点是否位于长链,动态地选择拓扑改变方式。网络建链完成后,计算出每个节点在链中到前后两个节点的距离分别为  $d_f(i)$  和  $d_b(i)$  (链头节点和链尾节点除外),然后将它们与距离阈值  $d_{\text{threshold}}$  进行比较:

(1) 如果  $d_f(i) > d_{\text{threshold}}$  或者  $d_b(i) > d_{\text{threshold}}$ , 表示节点  $i$  处于长链中,节点属性值  $\text{Node.d}(i) = 1$ ;

(2) 如果  $d_f(i) < d_{\text{threshold}}$  并且  $d_b(i) < d_{\text{threshold}}$ , 表示节点  $i$  处于非长链中,节点属性值  $\text{Node.d}(i) = 0$ ;

然后下一轮开始时,检查所有上一轮结束时死亡节点的  $\text{Node.d}$  属性值,如果出现  $\text{Node.d}(i) = 1$ ,表示网络中处于长链的节点死亡,如果将死亡节点前后两个节点直接连接很有可能产生长链,所有网络重新根据贪婪算法建立拓扑;反之,表示网络中处于长链的节点没有死亡,网络只需要将死亡节点的前后两个节点连接来构造新的网络拓扑。这样就动态地节约了节点重新构建拓扑所消耗的能量。

### 3.3.2 簇头选择阶段

PEGASIS算法中节点轮流担当簇头节点,造成远离基站的节点较早死亡。在 PEGASIS 算法的链上每个节点的数据传输方向有两个,本文根据剩余能量最少节点在链上位置,考虑选择簇头使剩余能量最少节点在下一轮的能耗最少,选择剩余能量最少节点消耗最少方向的节点作为候选簇头节点。再在候选节点中根据节点剩余能量和节点到基站的距离选择簇头节点,如图4所示。

具体步骤如下:

(1) 找出所有节点中能量最少的节点,假设为3号节点。

(2) 当  $d_1 > d_2$  时,4、5、6号节点作为簇头的候选节点;当  $d_1 < d_2$  时,1、2号节点作为簇头的候选节点;

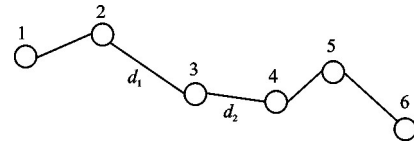


图4 应用程序框架

(3) 在候选节点中,计算出各个节点  $c(i)$  值,公式如下:

$$c(i) = \frac{E_{\text{res}}(i)}{d_{\text{toBS}}(i)} \quad (1)$$

式中,  $E_{\text{res}}(i)$  为  $i$  号节点的现在的剩余能量,  $d_{\text{toBS}}(i)$  为  $i$  号节点到基站的距离。

每一轮基站通过式(1)计算出候选节点的  $c(i)$  值,比较各个节点的  $c(i)$ ,选择  $c(i)$  值最大的节点,广播给整个网络。

采用上述簇头选择算法,虽然考虑了能量最少节点的能耗,但是可能导致所选簇头不是最优簇头,造成节点能耗的分布不均,所以本文考虑当网络中出现即将死亡的节点时,使用以上算法,即当网络中出现节点剩余能量小于初始能量的  $\frac{1}{10}$ 。

节点在每轮数据传输时,各个节点将自身的剩余能量信息加入数据中,由于能量信息数据较小,能量增加可以忽略不计。

### 3.3.3 数据传输阶段

PEGASIS算法中,全部节点通过建立一条固定的数据链来实现数据的接收和转发,当网络中簇头节点确定后,节点沿着数据链相应的方向传递数据,但是可能出现一种情况,如图5所示。传感器节点到下一个节点的距离要大于传感器节点直接将数据传输给簇头的距离,如果把节点传给链上的下一个节点,会造成节点耗费更多的能量,所以本文考虑通过判断每当节点传输数据之前比较节点到下一个节点的距离和节点到簇头节点的距离,采用更加节能的数据传输方法。

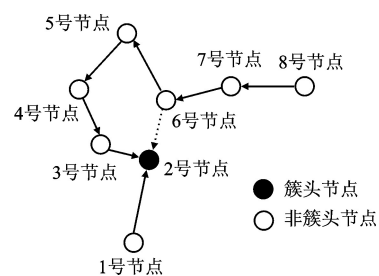


图5 不必要传输

在 PEGASIS 协议中,由于节点都知道其他节

点的地理位置,在网络传输数据阶段,链上的任意一个节点在接收到上一个节点传来的数据后,计算自身到下一个节点的距离以及自身到簇头节点的距离,比较两者,选择距离最小的作为下一跳节点,如图 6 所示.如果距离簇头节点较近,发送一个消息告诉链中的下一跳节点,在发生拓扑重构之前,都不需要等其传来的消息.

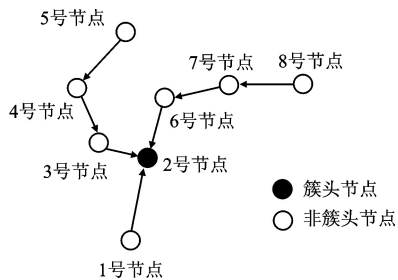


图 6 改进传输图

以上方法,形成了一种多分支的结构,链的长度有所减少,改进协议考虑根据分支数的数量,采用多个令牌的方法,来减小网络的延时.

采用以上方法,总能量是有所减少的,虽然簇头节点增加了负担,消耗了更多的能量,但是考虑到簇头节点作为网络中能量较多的节点,适当地增加簇头节点的负担可以有效地平衡网络节点的剩余能量.

#### 4 算法仿真与分析

本文采用 MATLAB 进行仿真,节点随机分布在  $100\text{ m} \times 100\text{ m}$  的局域内,初始能量相同,能量模型相同,参数具体设置如表 1 所示.

表 1 实验模拟参数

| 参数名                     | 参数值                                                       |
|-------------------------|-----------------------------------------------------------|
| 节点数量                    | 100 个                                                     |
| 基站位置                    | (50,175)                                                  |
| 节点初始能量                  | 0.25 J                                                    |
| 数据包长度                   | 2 000 bit                                                 |
| 控制包长度                   | 100 bit                                                   |
| $\epsilon_{\text{amp}}$ | $100\text{ pJ} \cdot \text{bit}^{-1} \cdot \text{m}^{-2}$ |
| 融合数据的能耗                 | 5 nJ/bit                                                  |

通过仿真比较不同  $i$  值条件下网络性能的表现,改进算法、PEGASIS 算法在网络生存时间和所有节点每一轮的平均剩余能量来比较两种算法在能耗高效性方面的表现.

图 7 所示为不同阈值情况下网络生存时间的

比较.

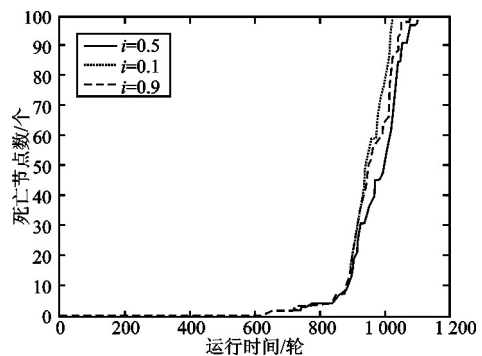


图 7 不同阈值下生存时间比较

从图 7 中可以看出,三种情况下第一个节点的死亡时间相当,这是由于采用上文提出的簇头选择算法,基于网络中能量最少的节点,且网络拓扑结构相同.但是当节点死亡数达到 10 个以后,网络生存时间出现了差异.这是由于网络中出现死亡节点后,三种情况根据不同的阈值选择拓扑重构方式,能量消耗有所不同.

$i$  为 0.1 情况下,50% 节点死亡时间和全部节点死亡时间要短于 0.5 和 0.9 两种情况,这是由于距离阈值过短,每当节点出现死亡重新建链时,都会判断死亡节点处于长链,选择上一节提出的方式重新建链,增加了节点额外能量消耗; $i$  为 0.9 情况下,50% 节点死亡时间要短于 50 m 情况和全部节点死亡时间要短于 0.5 情况下,但是长于 0.1 情况下,这是由于阈值过长,每当出现死亡节点,都会判断节点处于短链中,采用直接将前后两个节点连接的方法建链,减少了节点能耗,但是这样可能导致网络产生新的相对的长链,影响网络生存时间.综上所述, $i$  为 0.5 的情况下,算法表现出了较好的性能,所以下面的仿真参数  $i$  设置为 0.5.

图 8 所示为  $i=0.5$  情况下两种算法节点死亡时间的比较.表 2 所示为三种算法节点死亡时间.

表 2 三种算法节点死亡时间

| 死亡节点个数     | 1%    | 20%   | 50%   | 100%    |
|------------|-------|-------|-------|---------|
| 改进算法       | 632 轮 | 912 轮 | 998 轮 | 1 105 轮 |
| PEGASIS 算法 | 110 轮 | 784 轮 | 888 轮 | 981 轮   |

改进算法明显要优于 PEGASIS 算法,生存时间分别延长了 4.7 倍;20% 节点的生存时间,改进算法较 PEGASIS 算法分别的延长了 16%;50% 节点生存时间,改进算法较 PEGASIS 算法分别的延长了 12%. 而全部节点死亡时间,改进算法较 PEGASIS 算法分别的延长了 12%. 综上所述,改进算

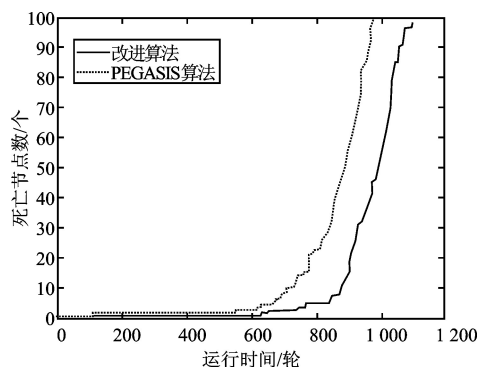


图8 节点死亡时间比较

法在网络生存时间方面的表现优于 PEGASIS 算法. 两种算法和能量高效利用方面的比较如图 9 所示.

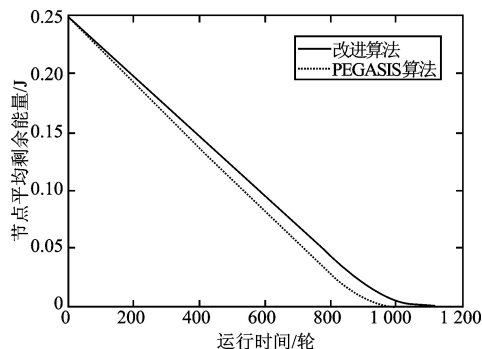


图9 节点平均剩余能量比较

改进算法节点的平均剩余能量明显高于 PEGASIS 算法, 这是由于改进算法当节点死亡时, 根据死亡节点是否处于长链, 动态地选择下一轮拓扑构造方法, 减少了拓扑重构带来的额外开销.

## 5 结束语

在 PEGASIS 算法中, 由于每当节点死亡时, 网络需要重新构造拓扑, 这样就造成了节点额外开销. 同时在簇头选择时, 没有综合考虑网络生存时间各方面影响因素. 针对以上问题, 本文结合距离阈值, 当网络中出现死亡节点时, 动态地选择拓扑重构方式, 并且在簇头选择上, 以使剩余能量最少节点耗能最少为前提, 综合考虑节点的剩余能量和节点到基站的距离, 并在数据传输阶段选择较近的距离的节点作为下一跳节点. 仿真结果表明: 改进后的算法减

少节点能耗, 延长了网络生存时间.

## 参考文献:

- [1] 尚凤军. 无线传感器网络通信协议技术[M]. 北京: 电子工业出版社, 2011.
- [2] 罗玉宏, 陈松乔, 王建新. 移动自组网中能量有效的路由算法[J]. 计算机工程及应用, 2004, 40(36): 15-21.
- [3] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy efficient communication protocol for wireless micro-sensor network[C] // IEEE Proceeding of the Hawaii International Conference on system Science. Washington DC: IEEE Computer Society, 2000: 3005-3014.
- [4] Lindsey S, Raghavendra C S. PEGASIS: Power-efficient gathering in sensor information systems[J]. IEEE Aerospace and Electronic Systems Society, 2002, 3(3): 1125-1130.
- [5] 余勇昌, 韦岗. 无线传感器网络中基于 PEGASIS 协议的改进算法[J]. 电子学报, 2008, 36(7): 1309-1313.
- [6] 吴联芳. 基于模拟退火算法的无线传感网 PEGASIS 算法[J]. 江南大学学报: 自然科学版, 2008, 7(4): 438-442.
- [7] 胡森来, 张昱, 金心宇, 等. 基于遗传算法的无线传感网 PEGASIS 算法的改进[J]. 江南大学学报: 自然科学版, 2008, 7(4): 420-424.
- [8] Vibha Nehra, Raju Pal, Ajay K Sharma. Fuzzy-based leader selection for topology controlled PEGASIS protocol for lifetime enhancement in wireless sensor network[J]. International Journal of Computers & Technology, 2013, 4(3): 755-764.
- [9] Feng Sen, Qi Bing, Tang Liangrui. An improved energy-efficient PEGASIS-based protocol in wireless sensor networks[C] // 2011 Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). China: Shanghai, 2011: 2282-2285.
- [10] 胡峻浩, 刘兴长, 谈昨非. 基于禁忌算法的无线传感器网络 PEGASIS 算法改进[J]. 后勤工程学院学报, 2013, 29(4): 91-96.

## 作者简介:

胡峻浩 男, (1988-), 硕士研究生. 研究方向为无线传感器网络.