

# 无线传感器网络中基于 PEGASIS 协议的改进算法

余勇昌<sup>1,2</sup>, 韦 岗<sup>1</sup>

(1. 华南理工大学电子与信息学院, 广东广州 510640; 2. 中国电信集团广州研究院, 广东广州 510630)

**摘要:** 作为能量有效、基于链状结构的路由协议的典型代表, PEGASIS 协议存在三个方面的不足, 一是链的生成算法会导致相邻节点间产生长链; 二是链头节点选取方法会导致节点间能量消耗不均衡; 三是链头节点重选频率增加了通信开销. 本文基于 PEGASIS 提出了一种新算法, 新算法通过引入距离门限避免相邻节点间产生长链; 通过综合考虑节点剩余能量以及节点到基站的距离来选取链头节点; 通过降低链头节点重选频率来节省通信开销. 分析和仿真结果表明, 在未增加算法复杂度的情况下, 新算法在平衡节点能耗和延长网络寿命方面比 PEGASIS 算法具有更优越的性能.

**关键词:** 无线传感器网络; 路由算法; 能量有效; 距离门限; 选取策略

**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2008) 07-1309-05

## An Improved PEGASIS Algorithm in Wireless Sensor Network

YU Yong-chang<sup>1,2</sup>, WEI Gang<sup>1</sup>

(1. School of Electronics and Information, South China University of Technology, Guangzhou, Guangdong 510640, China;

2. Guangzhou Research Institute, China Telecom Co., Ltd, Guangzhou, Guangdong 510630, China)

**Abstract:** As a representative of chain based protocol in wireless sensor network, PEGASIS is an elegant solution on energy efficiency. However, there are still three deficiencies, inevitability of long link (LL) between neighboring nodes, unevenness of energy consumption distribution among nodes and too much overhead on LEADER node reelection strategy, with PEGASIS protocol. In this paper, we propose a new energy efficient PEGASIS based protocol called EEPB that uses distance threshold to avoid formation of LL and adjusts the frequency of LEADER node reelection to lessen overhead. Also unlike LEADER node election method used in PEGASIS, our proposed algorithm adopts the remaining energy levels of nodes and the distances between nodes and the BS (base station) to decide which node is qualified to be the LEADER of the chain. Our analysis and simulation results show that our proposed algorithm, without increase in complexity, provides more uniform energy consumption among nodes and can prolong network lifetime compared to PEGASIS.

**Key words:** wireless sensor network; routing algorithm; energy efficient; distance threshold; election strategy

### 1 引言

低功耗无线通信技术、嵌入式计算技术和微型传感器等技术的飞速发展和日益成熟, 使得大量的、低成本的微型传感器通过无线链路自组织成无线传感器网络 (Wireless Sensor Network) 成为现实<sup>[1,2]</sup>. 在无线传感器网络中, 节点的能量非常有限, 一旦电池消耗完, 对节点进行充电或更换电池一般都不是现实的. 因此, 如何提高能量有效性、均衡节点能量消耗、延长网络寿命、避免网络分裂等问题便成为无线传感器网络路由协议研究的重要课题.

无线传感器网络中, 节点的能量大部分用于数据发送和接收<sup>[3]</sup>. 如果所有的节点通过直接通信的方式传送数据到基站 (Base Station), 那么离基站最远的节点将因为能量很快耗尽而提前死亡. 如果节点通过多跳的方式

发送数据到基站, 那么离基站最近的节点将会因为要转发大量其他节点的数据而率先死亡.

能否均衡节点能耗和延长网络寿命已经成为判断路由算法性能优劣的一个重要指标<sup>[4]</sup>, 因此好的路由算法必须要在尽量不增加路由算法复杂度的情况下均衡节点的能量消耗、延长网络的寿命.

本文基于 PEGASIS<sup>[5]</sup> (Power-Efficient Gathering in Sensor Information Systems) 提出了一种新算法 EEPB (Energy-Efficient PEGASIS-Based protocol), 新算法通过引入距离门限避免相邻节点间产生长链; 通过综合考虑节点剩余能量以及节点到基站的距离来选取链头节点; 根据网络中节点剩余率决定链头节点的重选频率. 分析和仿真结果表明, 相对 PEGASIS 算法, 改进后的算法在平衡节点能耗和延长网络寿命方面具有更优越的性能.

## 2 PEGASIS 算法

PEGASIS 协议是一种典型的基于链状结构的路由协议, 是 LEACH 协议<sup>[6]</sup>的增强算法。PEGASIS 算法的核心思想是利用贪婪算法生成一条由所有节点组成的单链, 链上的节点只与自己的邻居节点通信。除端节点外, 每个节点将自己收到的数据与自己产生的数据进行融合, 然后再将融合后的数据沿链头节点(LEADER)方向传递给链上的相邻节点, 这一过程一直持续直到数据到达 LEADER 节点, LEADER 节点负责将数据传送到远端的基站。

相比 LEACH 算法, PEGASIS 算法通过以下几种措施取得了节点能耗的均衡和网络寿命的延长<sup>[5]</sup>: (1) 基于链状的拓扑结构减少了节点间通信的平均距离; (2) 不需要动态成簇的开销; (3) 仅仅一个 LEADER 节点传送数据到基站, 减少了发往远端基站的数据量, 从而节省了能量; (4) 链上节点轮流担任 LEADER 节点的方式使得节点间能耗更加均衡。另外, PEGASIS 建链算法比较简单, 复杂度为  $O(n^2)$ 。

PEGASIS 算法的建链过程按轮进行, 首先从离基站最远的节点开始建链, 这个节点作为链的端节点(END)。在接下来的每一轮都让离 END 最近的节点加入链并作为新的 END 节点。这个过程一直持续直到所有的节点都加入了链, 这样一条基于贪婪算法由所有节点组成的链就形成了。

PEGASIS 建链方法如图 1 所示, END 表示链的端节点, CHAIN 表示需要建的链,  $N$  表示节点集,  $d(i, j)$  表示节点  $i$  和节点  $j$  之间的距离,  $Append(CHAIN, END)$  表示将 END 节点加入到链的末端。

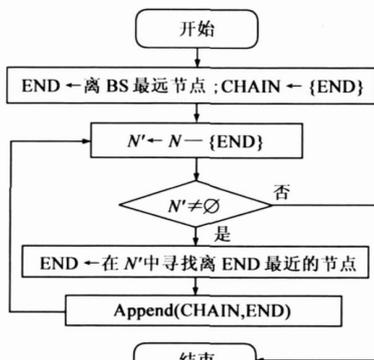


图 1 PEGASIS 建链算法

图 2 为一个 50 米 × 50 米的区间随机布放了 20 个节点, 利用 PEGASIS 建链方法得到的链如图 3 所示。从图中可以看出, 节点 11 和节点 12 之间的链路、节点 15 与 16 之间链路、节点 18 与 19 之间链路比其他的链路长得多, 我们称这些链路为长链 LL。假如节点 20 为 LEADER 节点, 很显然, 节点 11、15、18 在发送数据给节点 12、16、19 时要消耗大量的能量。所以, 为了节约能量, 在建链的过程中就必须采取措施以避免邻居节点间产生长链。

PEGASIS 链的 LEADER 节点由链上的节点轮流担

任。在第  $i$  轮数据传输时, 由第  $i \bmod N$  个节点担任 LEADER 节点并负责将数据传送给远端的基站。每一轮通信结束后, LEADER 节点将进行重新选取, 这样便增加了通信开销。为了减少通信开销, 我们需要避免每一轮通信结束后都进行链头节点重选。

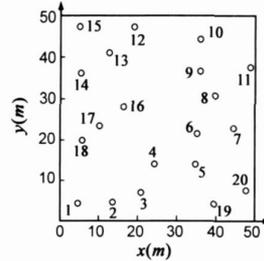


图 2 随机布放了 20 个节点的 50m × 50m 网络

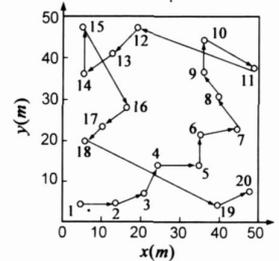


图 3 利用 PEGASIS 建链算法得到的链

由于在每一轮通信中, LEADER 节点消耗的能量要远远多于链上其他节点消耗的能量, 所以 PEGASIS 链头节点选取方法会导致离基站远的节点率先死亡。为了避免这种情况发生, 我们需要采取新的 LEADER 节点选取算法。

## 3 其他相关改进算法

虽然 PEGASIS 算法在节点能耗均衡和网络寿命延长方面取得较好的效果, 但 PEGASIS 算法仍然存在以下三个方面的不足, 一是: PEGASIS 建链方法是基于最近邻居成链算法, 由于已经加入链的节点和死亡的节点都不能被再次访问, 所以相邻节点间的长链不可避免; 二是: LEADER 节点采用轮流担任的方式将导致离基站远的节点率先死亡; 三是每一轮通信结束后都重新选取链头节点, 从而增加了通信开销。

针对 PEGASIS 算法存在的问题, 国内外学者也开展了相关研究, 其中文献[7]和[8]是取得较好性能的改进算法。在文献[7]中, 作者仍然采用 PEGASIS 的建链方法, 但在 LEADER 节点选取方法上作了改进。作者将所有链路长度进行排序, 然后限制其中最长的 5 条链路的端节点担任 LEADER 节点。仿真结果显示改进后的算法在能耗方面比原 PEGASIS 算法具有更好的性能。在节点死亡百分比为 1%、20%、50%、100% 的情况下, 新算法的通信轮数比 PEGASIS 算法的通信轮数分别延长了 14.1%、35.9%、40.3% 和 33.8%<sup>[7]</sup>。虽然文献[7]在能量节省方面取得不错的效果, 但其算法仍然会导致离基站远的节点在通信过程中率先死亡。

在文献[8]中, 为了克服 PEGASIS 建链方法中存在的缺陷, 作者提出了一种新的建链方法。新方法仅在建链第一步和第二步等同于 PEGASIS 方法, 其核心思想就是每一步将一个待加入链的节点插入到链上两个相邻的节点之间, 插入位置的选择标准就是使得数据沿插入

节点后的链传输时能量消耗的增加量最小, 所以这种建链方法也称为最小能量建链法。实验结果表明, 在中等节点密度条件下, 当 1% 的节点死亡时, 新算法比 PEGASIS 算法在能耗方面节省了 15% 到 30%。具体建链方法如图 4 所示。虽然文献[8]在节省能量方面取得一定的效果, 但其相对 PEGASIS 建链方法来说, 算法复杂度增加了, 其算法复杂度为  $O(n^3)$ 。

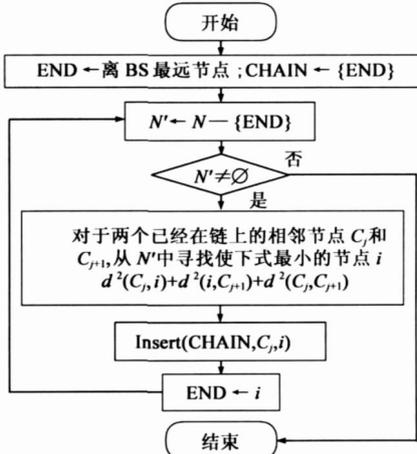


图 4 基于最小能量建链算法

#### 4 基于 PEGASIS 协议增强算法 EEPB

为了均衡节点的能量消耗和延长节点的寿命, 本文基于 PEGASIS 协议提出了一种新算法 EEPB, EEPB 算法由三个阶段组成, 分别是 (1) 建链阶段; (2) LEADER 节点选取阶段; (3) 数据传输阶段。

##### 4.1 建链阶段

与 PEGASIS 建链方法一样, EEPB 建链方法也从离基站最远的节点开始建链。为了判断相邻节点之间的链路是否为长链, 我们设定一个用于判断链路是否为长链的距离门限并用  $d_{threshold}$  表示。如图 5 所示, 假设已经有  $i$  个节点加入了链。节点  $v$  和节点  $v+1$  之间的链路长度用  $d_v$  表示。

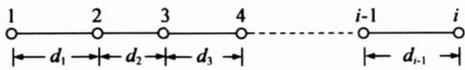


图 5 由  $i$  个节点组成的部分链

如果两个相邻节点之间的链路长度大于等于  $d_{threshold}$ , 我们便称这两个相邻节点之间的链路为长链。我们采用两种方法定义距离门限, 如方程(1)和(2)所示:

$$d_{threshold} = \alpha \left[ \sum_{v=1}^{i-1} d_v / (i-1) \right], \quad v = 1, 2 \dots i-1 \quad (1)$$

$$d_{threshold} = \beta \max(d_v), \quad v = 1, 2 \dots i-1 \quad (2)$$

其中  $\alpha$  和  $\beta$  都是可以由用户来定义的系数。上述两种距离门限定义方法没有实质上的区别, 为了简化讨论, 本文的距离门限采用方程(1)进行定义。

在所有待加入链的节点中, 假设节点  $i+1$  是最靠近节点  $i$  的节点。在节点  $i+1$  加入链前, 先将自身与节点  $i$  之间的距离  $d_i$  与  $d_{threshold}$  进行比较:

(1) 如果  $d_i \leq d_{threshold}$ , 表示节点  $i+1$  和节点  $i$  之间的链路不是长链, 那么节点  $i+1$  通过直接与节点  $i$  相连的方式加入链;

(2) 如果  $d_i > d_{threshold}$ , 表示节点  $i+1$  和节点  $i$  之间的链路是长链, 这时节点  $i+1$  不能直接与节点  $i$  相连, 节点  $i+1$  将从链上找出距离自身最近的节点。假设在所有已经加入链的节点中, 节点  $j$  离节点  $i+1$  最近。如果节点  $i+1$  与节点  $j$  之间的距离小于  $d_{threshold}$ , 那么节点  $i+1$  将通过直接与节点  $j$  相连的方式加入链; 如果节点  $i+1$  与节点  $j$  之间的距离大于等于  $d_{threshold}$ , 那么说明此时长链不可避免, 节点  $i+1$  仍然通过与节点  $i$  相连的方式加入链。

EEPB 建链方法具体实现如图 6 所示。

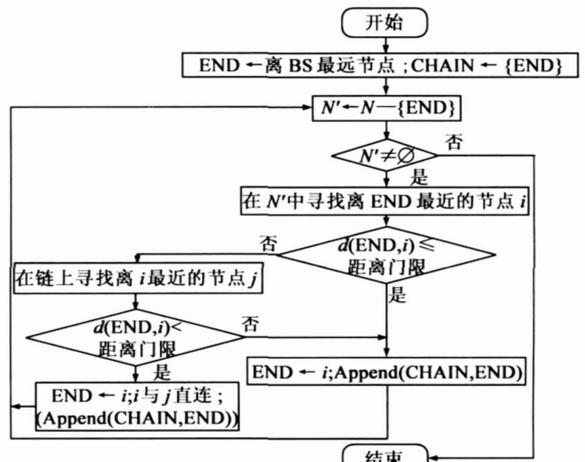


图 6 EEPB 建链算法

取  $\alpha = 1.1$ , 利用 EEPB 建链方法, 对图 2 中的 20 个节点重新成链, 计算后得到的链将如图 7 所示。从图 7 中可以看出, 20 个节点组成的不再是单链, 而是有分支的链, 图 3 中的长链已经不存在了。所以通过设立距离门限的建链方法可以有效地避免相邻节点间产生长链, 从而可以有效地节约能量。

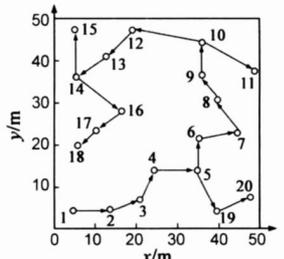


图 7 采用 EEPB 建链方法得到的链

值得提出的是, 当  $\alpha$  为无穷大(或足够大)时, 距离门限  $d_{threshold}$  也趋于无穷大(或足够大), 因此任何两个相邻的节点之间的距离都小于  $d_{threshold}$ , 也即任何两个相邻节点之间都不会产生长链, 在这种情况下, EEPB 算法得到的链与 PEGASIS 算法得到的链完全相同。

另外, 与 PEGASIS 算法一样, EEPB 算法中链上任何

一个节点死亡都将重新按上述建链方法建链, 并且距离门限也随之而变. EEPB 算法并没有增加计算的时间复杂度, 其时间复杂度和 PEGASIS 算法一样为  $O(n^2)$ .

#### 4.2 LEADER 节点选取阶段

EEPB 算法的 LEADER 节点选取方法与 PEGASIS 有很大的不同. EEPB 算法在选取 LEADER 节点时考虑了节点的剩余能量及节点到基站的距离. 很显然, 在发送同样数据量到基站的情况下, 离基站远的节点消耗的能量要多于离基站近的节点. 所以在节点剩余能量相等的情况下, EEPB 算法优先选取离基站近的节点担任 LEADER 节点, 以便在传输数据到基站时节约能量; 同样, 在节点离基站距离相同的条件下, EEPB 算法优先选取剩余能量多的节点担任 LEADER 节点, 以便避免剩余能量少的节点率先死亡. 总之, EEPB 算法优先选取剩余能量多、距离基站近的节点担任 LEADER 节点. EEPB 算法利用式(3)和(4)计算各个节点的  $Q$  值, 并从链上选取  $Q$  值最大的节点担任 LEADER 节点.

$$d_i = \sqrt{(x_i - x_{BS})^2 + (y_i - y_{BS})^2} \quad (3)$$

$$Q_i = E_{residual_i} / d_i \quad (4)$$

其中  $(x_i, y_i)$  和  $(x_{BS}, y_{BS})$  分别是节点  $i$  和基站的坐标,  $E_{residual_i}$  是节点  $i$  的剩余能量,  $Q$  是节点到基站的距离  $d$  及节点剩余能量  $E_{residual}$  的函数.

EEPB 算法与 PEGASIS 算法一样, 基站已知所有节点的位置并负责链头节点的选举并广播通知所有的节点. 每个节点都具有功率控制能力, 能以最小的能量到达目的接收节点. 如果节点没有足够的能量直接与基站通信, 那么将被排除参与链头节点的选取.

#### 4.3 数据传输阶段

EEPB 算法生成的链不同于 PEGASIS 算法生成的链, PEGASIS 链是有两个端节点的单链, 而 EEPB 链一般都是具有分支的单链, 并具有不止 2 个端节点. 图 7 中的链就是典型的 EEPB 链, 从图中可以看出, EEPB 链共有 5 个端节点.

每轮数据收集由基站发送 Beacon 信息触发, 由于节点知道自身在数据链上的位置, 因此可以计算得出自己应该获得的时隙. EEPB 算法用经链头节点初始化后的控制令牌包“Control Packets”从链的端节点启动数据传输过程. 由于控制令牌包“Control Packets”非常小, 因此其带来的开销可以忽略不计.

EEPB 算法的数据传输阶段等同于 PEGASIS. 端节点直接将自身产生的数据沿着 LEADER 节点方向传递给相邻节点; 中间节点将收到的数据与自身的数据进行融合, 然后将融合后的数据沿着 LEADER 节点的方向传递给相邻节点. 这一过程一直持续直到数据到达 LEADER 节点. LEADER 节点收到两边相邻节点的数据后, 将

自身的数据与两个相邻节点的数据进行融合, 然后再将融合后的数据传递给远端的基站. 至此, 一轮数据传输过程就结束了.

值得提出的是, EEPB 算法并不会在每一轮通信结束后重选链头节点, 而是要等到一定轮数的通信结束后才重选链头节点. EEPB 算法采取的重选策略是: 开始阶段每 30 轮才发起一次链头节点重选; 当节点剩余率为 80% 时, 每 20 轮发起一次链头节点重选; 当节点剩余率为 50% 时, 每 10 轮发起一次链头节点重选; 当节点剩余率为 30% 时, 每 5 轮发起一次链头节点重选; 当节点剩余率为 20% 时, 像 PEGASIS 算法一样, 每轮都发起链头节点重选; 一旦节点在担任链头节点期间因为能量减少而不能与基站通信, 那么 EEPB 算法立即重新启动链头节点选取过程.

#### 5 仿真结果及分析

为了评价 EEPB 算法的性能, 我们采用在  $100\text{m} \times 100\text{m}$  的区域随机布放 100 个节点, 基站位于坐标(50, 300)处, 数据包的长度为 2000 比特. 我们的射频能量模型采用文献[6]中的射频能量模型, 每个节点都具有功率控制能力, 能以最小的能量到达目的接收节点. 我们采用节点参与通信的轮数代表节点的寿命, 轮数越大表示节点的寿命越长. 当一个节点死亡后, 在其后的仿真中都不再考虑. 并且如同 PEGASIS 算法一样, 假设所有的节点初始能量相同(为 0.5J)并且能和基站直接通信.

当  $\alpha$  值取 1.5 时, EEPB 算法与 PEGASIS 算法的性能比较如图 8 所示. 从图 8 可以看出, 当  $\alpha$  值取 1.5 时, EEPB 算法在能量效率方面的性能要好于 PEGASIS 算法. 当 1%、20%、50%、100% 节点死亡时, EEPB 算法的通信轮数比 PEGASIS 算法分别延长了 20%、32%、35% 和 47%. 当  $\alpha$  分别取 1.2、1.5、3 时, EEPB 算法的性能如图 9 所示, 从图 9 可以看出,  $\alpha$  值越小, EEPB 算法的性能越好, 这是因为随着  $\alpha$  值减小, 距离门限变小, 这将缩短节点间通信的平均距离, 从而节省了通信能量.

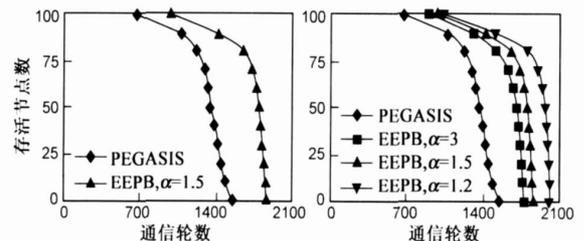


图 8 PEGASIS 与 EEPB 的性能比较 ( $\alpha=1.5$ )

图 9  $\alpha$  取不同值 (1.2, 1.5, 3) EEPB 的性能情况

为了对比 EEPB 算法和 PEGASIS 算法在节点能耗均衡方面的性能, 如图 10 所示在  $100\text{m} \times 100\text{m}$  的区域随机布放 100 个节点. 当其中 25 个节点死亡时, PEGASIS 算法和 EEPB 算法剩余节点的分布情况分别如图 11 和

图 12 所示. 从图 11 可以看出, 在 PEGASIS 算法中, 由于节点能耗不均衡导致死亡的节点基本都是离基站远的节点, 这与我们前面分析的结果一致. 而在 EEPB 算法中, 由于采取了改进措施, 节点的能耗更加均衡, 因此死亡的节点也随机地分布在整个网络, 这样就保证了网络不至于因为某些节点死亡而造成网络分裂甚至失效.

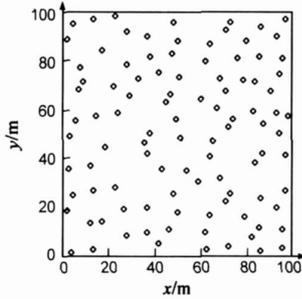


图 10 100m×100m 的区域随机布放的 100 个节点

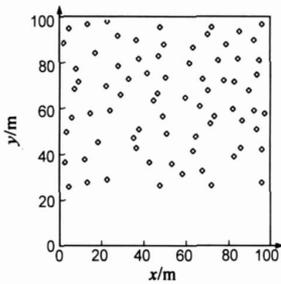


图 11 PEGASIS 算法存活节点的分布情况

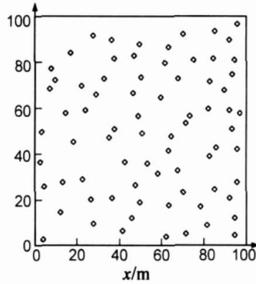


图 12 EEPB 算法存活节点的分布情况

## 6 结论

通过分析典型的基于链状结构的 PEGASIS 算法, 我们发现虽然 PEGASIS 算法在能耗方面比 LEACH 算法优越, 但 PEGASIS 算法同样存在着三大不足: 一是链的生成算法容易导致相邻节点间产生长链; 二是链头节点选取方法容易导致节点间能量消耗分布不均匀; 三是每一轮通信结束后都重新选取链头节点, 从而增加了通信开销. 针对 PEGASIS 的不足, 本文提出了一种称为 EEPB 的改进算法, EEPB 算法通过在建链过程中引入距离门限避免相邻节点间产生长链; 通过优先选取剩余能量多、距离基站近的节点担任链头节点来均衡节点间的能量消耗; 通过降低链头节点的重选频率来减少通信开销. 分析和仿真结果表明, 相对经典 PEGASIS 算法, 改进后的算法不但没有增加复杂性, 而且在平衡节点能耗和延长网络寿命方面具有更优越的性能.

然而, 在基于链状结构的网络中, 为了保证可靠性, 每个时刻只允许一对节点进行通信, 这无疑大大增加了数据传输时延, 资源利用率较低; 另外, 如果链上某个节点失效, 会引起整条数据链重建, 这样会影响网络的健壮性. 总体来说, 链状结构比较适合较小规模的无线传感器网络, 可以充分发挥数据聚合和短距离通信的优势, 达到节省能量的目的. 而对于规模较大的网络, 如果整个监测区域只形成一条数据链, 那么数据传递时延会

大大增加, 此时链状结构具有较大的局限性. 因此, 可以考虑在区域内按照某些规则形成多条链, 多链结构会在很大程度上改善上述问题.

## 参考文献:

- [1] 任丰源, 黄海宁, 林闯. 无线传感器网络[J]. 软件学报, 2003, 14(7): 1282-1291.  
Ren Feng yuan, Huang Hai ning, Lin Chuang. The wireless sensor networks[J]. Journal of Software, 2003, 14(7): 1282-1291. (in Chinese)
- [2] Edgar H C. Wireless Sensor Networks: Architecture and Protocol[M]. New York: Auerbach Publications, 2004. 21-110.
- [3] A Sinha, A Chandrakasan. Dynamic power management in wireless sensor networks[J]. IEEE Design and Test of Computers, 2001. 1567-1676.
- [4] 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, 36: 15-21.  
Luo Yu hong, Chen Song qiao, Wang Jiar xin. The energy efficient routing algorithm in mobile ad hoc networks[J]. Engineering and Application of Computer, 2004. 36: 15-21. (in Chinese)
- [5] S Lindsey, C Raghavendra. PEGASIS: Power efficient gathering in sensor information systems[J]. IEEE Aerospace Conference Proceedings, 2002. 1125-1130.
- [6] Heinzelman W, Chandrakasan A, Balakrishnan K. Energy efficient communication protocol for wireless micro sensor networks[A]. In Proceedings of the Hawaii Conference on System Sciences[C]. 2000. 3005-3014.
- [7] 王薇. 无线传感器网络低功耗分级路由协议研究[D]. 浙江, 浙江大学信息学院, 2006.  
Wang Wei. Research on Low Energy Hierarchy Routing Protocol in Wireless Sensor Networks[D]. Zhejiang, The School of Information of Zhejiang University, 2006. (in Chinese)
- [8] Kemei Du, Jie Wu, Dan Zhou. Chair based protocols for data broadcasting and gathering in the sensor networks[A]. Proc. of Workshop on Parallel and Distributed Scientific Engineering Computing with Applications[C]. 2003. 1926-1933.

## 作者简介:



余勇昌 男, 1976 年生于江西临川, 博士研究生. 主要研究方向为无线传感器网络路由算法、无线接入. E-mail: yuyongchang@21cn.com

韦 岗 男, 1963 年生于广西宾阳, 教授, 博士生导师. 主要研究方向为现代数字信号处理、无线通信网络. E-mail: yych@gsta.com