

# 基于同心环分簇网络模型的 WSN 能量空洞避免方法研究

刘 震 郭 航

(国防科技大学信息系统与管理学院信息系统工程实验室 长沙 410073)

**摘 要** 如何避免能量空洞现象已经成为无线传感网络应用中迫切需要解决的关键问题。在分析国内外能量空洞的避免机制和方法的基础上,考虑在同心环模型的基础上对逐层分环构思进行扩展,提出了一种基于同心环分簇网络模型的无线传感器网络能量空洞避免方法,对该方法的网络结构模型和能量消耗模型进行了描述,对簇头节点位置初始化、簇头轮转选择算法和簇间多跳路由算法进行了设计,并通过仿真实验对算法进行了对比分析。

**关键词** 无线传感器网络,能量空洞,分簇,均匀分环

中图分类号 TP311 文献标识码 A

## Study on Concentric-Ring and Cluster-based Energy Hole Avoiding Method in Wireless Sensor Networks

LIU Zhen GUO Hang

(Science and Technology on Information System Engineering Laboratory, College of Information System and Management, National University of Defense Technology, Changsha 410073, China)

**Abstract** How to avoid “Energy Hole” phenomenon has become a critical problem in wireless sensor networks’ applications. This paper analyzed the existing domestic and foreign solution of energy hole and put forward a concentric ring and cluster based energy hole avoiding method in wireless sensor networks. It described the network model and energy consuming model of the method. Furthermore, it designed the location initial algorithm of cluster heads, rotary selection algorithm of cluster head and multiple-hop routing algorithm between clusters. Finally, simulations were conducted to test and analyze the algorithms.

**Keywords** Wireless sensor network, Energy is empty, Clumping, Uniform points ring

## 1 引言

无线传感器网络(Wireless Sensor Networks, WSN)被称为 21 世纪最为重要的应用技术之一,其应用前景是不言而喻的。WSN 中存在的一个关键问题是能量利用率问题。由于能量利用率问题的存在,使得能量空洞成为 WSN 一种特定的现象,它的存在使 WSN 的寿命缩短,网络有效工作时间减少。如何降低能量空洞现象已经成为目前 WSN 应用中迫切需要解决的关键问题。

WSN 中的节点一般分为传感器节点和 sink 节点,传感器节点采用多跳的方式为 sink 节点传输数据。一般传感器节点随机分布到工作区域,它们可自组织成为网络。但是节点能量是有限的,而且不可能人为地为节点更换电池或者电源设备,所以每个节点的寿命都是有限的。一般的,由于能源消耗完毕或者其他原因而导致节点不能正常工作,则称为节点死亡,节点死亡就有可能导致能量空洞现象的产生。

能量空洞避免机制的研究时间还不长,但是研究人员已经深刻认识到能量空洞现象的存在给 WSN 带来的危害,提出了不同的能量空洞避免策略。有些采用节点密度控制策略来改善能量利用率,如文献[4]给出多对一的 WSN 模式的研究模型,较早地提出采用合理的数据模型分析 WSN 的能量

空洞现象,文中主要是对节点均匀分布网络中的能量空洞进行有效分析,但是没有涉及到网络中的能量空洞是否可以避免。文献[5]弥补了文献[4]的不足,采用同心环模型来分析能量空洞现象,文中假设环半径是可调节的,证明圆环宽度相等时路由上的能量消耗才能达到最小,在此基础上给出能量空洞是否可以避免的结果。文献[6]提出基于能量均衡的节点部署算法。其以分层网络结构为基础,要求每个环上的节点数必须相等,这样,靠近 sink 节点的环的节点密度比较大,但是在实际环境中节点密度不需要这么高,这必然造成了能量浪费和成本的提高。文献[8]提出了相邻节点感知的节点密度控制策略,即在一条路由路径上并不是所有节点都需要监听数据,可通过减少不必要的监听来降低能量消耗。网络拓扑关系调整需要相邻节点互相监听来完成。不同的区域采用不同节点密度,在一定程度上提高了网络能量利用的效率。还有一些研究人员采用分簇策略来提高网络能量的利用率。文献[1]提出了 LEACH 分簇协议,文中假设簇内的普通节点可以通过单跳方式把数据传输给簇头节点,簇头节点是以轮换方式产生的,这种模式在一定程度上平衡了节点间的负载。但簇头的产生和簇头之间距离导致的能量消耗也比较大,影响网络能量的整体利用率。文献[2]给出了 HEED 分簇协议算法。该算法根据普通节点的剩余能量选出一些候选的簇头

节点,这些候选簇头节点通过通信代价的大小来决定最终簇头节点。但是这种算法必须把 WSN 分为大小相等的簇。文献[5,9-11]提出基于不等簇半径轮换的能量空洞现象避免策略,网络中能量消耗最大的传感器节点决定了整个网络寿命,采用不等簇半径轮换的方式,就可以让能量消耗不仅仅只限于一个传感器节点,这样能量消耗就会分布到各个节点,使得整个 WSN 的能量消耗接近于均衡状态,但相对算法复杂,计算量加大。

根据研究显示,在 WSN 的同心环模型中,合理设置圆环宽度  $\Delta$  和各个圆环实际采用的传输距离等级  $x_i$  就可以使得整个 WSN 寿命达到最大。而且研究人员经过对网络结构的深入研究得知分簇网络结构模型比平面网络结构模型要消耗更少的能量。本文考虑结合同心环模型和分簇网络结构模型,提出一种基于同心环分簇网络模型的 WSN 能量空洞避免方法。第 2 节介绍基于同心环分簇网络能量空洞避免方法,对网络结构模型、能量消耗模型进行了设计,并在此基础上提出了簇头节点初始化算法、簇头选择算法和簇头间路由算法;第 3 节仿真实验对算法的性能进行了对比分析;最后是总结。

## 2 基于同心环分簇网络能量空洞避免方法

### 2.1 网络结构模型

本文提出的同心环分簇网络模型中的传感器节点间歇而周期性地对被监测对象进行监测和数据采集,并把数据传送到位于网络中心的 sink 节点。WSN 是一个半径为  $R$  的圆形网络区域;圆心位置就是 sink 节点所在的位置;圆形网络区域内均匀分布着密度为  $\rho$  的节点,节点不再移动且每个节点的初始能量都是  $\epsilon$  ( $\epsilon$  大于零),发射的数据包长度均为  $L$ bits,并且发射半径可调,最大传输距离为  $t_x$ ,节点数据不会冗余。同时分析中只考虑节点接收和转发数据的能量消耗,其他的能量消耗不考虑。把圆形网络区域从圆心向外分为圆环宽度相等的  $K$  个圆环,每个圆环半径分别是  $r_k$ ,而且  $0 < r_1 < r_2 < \dots < r_K = R$ ;圆环  $C_i$  的宽度是  $\Delta_i = r_i - r_{i-1}$ ,每个圆环的最大宽度小于节点的最大通信距离,即  $\Delta_{i_{\max}} < t_x$ 。网络采用分层结构性网络,采用分簇拓扑控制策略,每个簇的簇头节点通过等效距离原则来确定。如图 1 所示,在圆形网络区域中,圆心处为 sink 节点,半径为  $R$ ,传感器节点均匀分布在圆形区域中,密度为  $\rho$ ,将圆形网络区域从圆心向外划分为宽度相等的  $K$  个圆环,分别记为  $C_1, C_2, \dots, C_K$ ;基于这种圆形网络区域,传感器节点具有分布均匀性和区域对称性特点,位于  $C_1$  内的节点直接与 sink 节点进行通讯,也可通过簇头节点传递,这就加强了 sink 节点附近的节点能量利用率;位于  $C_i$  内的节点被归结为  $i$  型簇; $C_i$  环内的簇头节点只能和  $C_{i-1}$  内最近的簇头节点之间以单跳的方式进行通信,通信不存在数据冗余现象;监测数据通过簇头节点以多跳的方式传送到位于圆心处的 sink 节点, $C_2, \dots, C_{K-1}$  内的簇头节点不仅要负责接收本簇的成员节点传输的数据,还要承担起转发外环的簇头节点发送的数据, $C_K$  内的簇头节点只负责发送簇内成员节点传输的监测数据,不负责其他簇的数据转发,即  $C_K$  内的节点不负责承担数据中继的责任。

如图 1 所示,半径为  $R$  的圆形网络区域被分为宽度相等的 3 个圆环,即宽度为  $\Delta$ ,圆形区域的圆心位置为 sink 节点所

在。圆环内节点自发形成半径相等的圆形簇。簇内节点间采用单跳方式进行数据传输,对外通过簇头节点进行交互,簇头通过与相邻环内最近的簇头节点交互,通过多跳方式将数据传输到 sink 节点。如图 1,箭头表示了簇头节点之间、簇头节点和 sink 节点之间的数据传输关系。 $C_3$  内的簇头节点将本环内的节点接收的数据转发给  $C_2$  内的簇头节点,而  $C_2$  内的簇头节点不但接收本环内节点发送的数据,还要转发  $C_3$  内的数据, $C_1$  最后把自身所在环内的数据和  $C_2, C_3$  的数据发送给 sink 节点。

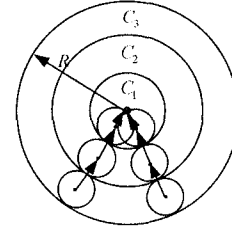


图 1 均匀分簇网络模型

### 2.2 能量消耗模型

#### 2.2.1 簇头节点周期轮转

在本网络模型中,各个簇的簇头节点周期性地地进行轮换,每一个传感器节点都有一次作为簇头节点的机会,这样保证簇内节点能量消耗相对均衡,从而提高网络能量利用率。本模型中,簇头节点周期性轮转一般包含两种方式:一种是外移方式(见图 2);一种是圆周轮转方式(见图 3)。

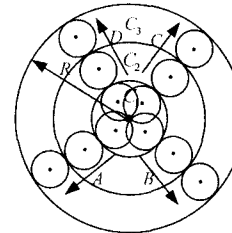


图 2 外移方式图示

如图 2 所示,簇头节点的轮转方式采用外移方式,ABCD 4 个箭头代表了网络簇头节点的周期轮转方向。

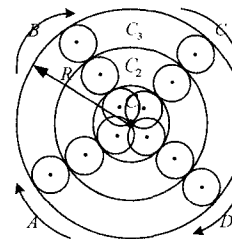


图 3 圆周轮转方式图示

如图 3 所示,采用了圆周方向轮转簇头节点,ABCD 4 个箭头代表了网络簇头节点周期圆周轮转方向,以 sink 节点为中心顺时针进行簇头节点轮转工作。

#### 2.2.2 簇头节点位置初始化

簇头节点的周期性轮转使得网络中每一个传感器节点在一个周期内尽可能地有一次承担簇头节点的机会,这就使得整个网络的能量消耗平均到每个传感器节点,防止靠近 sink 节点附近区域节点的过早死亡,避免能量空洞现象发生,使得网络寿命最大化。然而簇头节点初始位置的确定也是影响能量消耗大小的关键因素,簇头节点的初始位置对于簇头节点的周期性轮转起到前端铺垫作用。我们根据等效距离原则来决

定簇头节点的初始位置,由于传感器节点均匀分布在对称的圆形网络区域中,因此簇头节点均匀地分布在每个圆  $O_i$  上,  $C_i$  内一节点到 sink 节点的距离是  $d_i$ ,其中,假设环  $C_i$  内的簇头节点发送 1bit 数据到环  $C_{i-1}$  内的簇头节点所消耗的能量等价于圆  $O_i$  上的簇头节点发送 1bit 数据到  $C_{i-1}$  内的簇头节点所消耗的能量,  $x_i$  表示  $C_i$  内的簇头节点发送 1bit 数据到环  $C_{i-1}$  内的簇头节点所需要的有效距离,则:

$$x_i = d_i - d_{i-1} = \frac{2(r_i^2 + r_i r_{i-1} + r_{i-1}^2)}{3(r_i + r_{i-1})} - d_{i-1} \quad (1)$$

$$d_i = \frac{2(r_i^2 + r_i r_{i-1} + r_{i-1}^2)}{3(r_i + r_{i-1})} \quad (2)$$

### 2.2.3 簇头轮转选择算法

下面给出本网络模型中簇头分布式轮转选择算法,簇头节点选择依据是剩余能力的大小。首先在网络中选择剩余能量比较大的一部分节点作为簇头节点竞选的候选节点,成员节点成为簇头节点的概率为  $\tau$ 。在簇头选择过程中,其他节点处于休眠状态。

本算法中每个传感器节点都有唯一标识  $ID$ ,  $H_i$  代表候选簇头节点,候选簇头节点发射信号的半径为  $R_h$ ,即候选簇头的成员控制在半径为  $R_h$  的圆形区域中,以减少簇头间信号干扰和数据冗余。当一个候选簇头  $H_i$  正式上任后,该节点半径  $R_h$  区域内其他的候选簇头节点自动停止竞选。具体实现算法如算法 1。

#### 算法 1 均匀分簇的簇头节点轮转选择方案算法

```

1.  $\epsilon = \text{InitialE}$ ;
2.  $\phi \in \text{RNDE}(0, 1)$ ;
3.  $H_i, E = \phi\epsilon$ ;
4. While  $\phi < 1$  do
5.   if  $\phi * (H_i, E / E_{\text{average}}(c-1)) < \tau(1 - p * \text{mod}(c/\text{round}(1/p)))$ 
6.     be Clusterhead;
7.   End
8.   if be Clusterhead = true
9.     Send a Clusterhead_msg( $ID, H_i, E$ );
10.  End
11.  Wait;
12.  receiving a Clusterhead_msg from  $H_j$ ;
13.  if distance( $H_j, H_i$ )  $< R_c$ 
14.     $H_j$  be  $H_i, \text{self\_Clusterhead}$ ;
15.  End
iteration
16.  While be Clusterhead = true do
17.    for  $H_j \in H_i, \text{self\_Clusterhead}$ 
18.      if  $H_i, E > H_j, E$ 
19.        Send a Be_Clusterhead_msg( $ID$ );
20.      else
21.        Wait;
22.      End
23.    End
24.    Wait, receiving a Be_Clusterhead_msg from  $H_j$ ;
25.    if  $H_j \in H_i, \text{self\_Clusterhead}$ 
26.      Send a Quit_msg( $ID$ );
27.    End
28.    Wait, receiving a Quit_msg from  $H_j$ ;
29.     $H_i$  remove  $H_j$  from  $H_i, \text{self\_Clusterhead}$ ;
30.  End

```

31. End While

32. End While

算法符号说明:

$\epsilon$ : 节点能量, 赋初值;

$\phi$ : 在 0 和 1 之间的概率数;

$H_i, E$ : 节点剩余能量;

$\tau$ : 成为簇头节点的概率;

$c$ : 第几周期;

$p$ : 成为簇头的预先概率;

$ID$ : 节点的唯一标识;

$R_h$ : 节点所根据的发射半径;

$E_{\text{average}}(c-1)$ : 上一周期的节点平均能量;

$\text{Clusterhead\_msg}$ : 候选簇头节点信息;

$H_i, \text{self\_Clusterhead}$ : 候选簇头的邻近候选簇头集合;

$\text{Be\_Clusterhead\_msg}$ : 竞选为簇头节点;

$\text{Quit\_msg}$ : 去除候选簇头节点信息。

在算法 1 的伪代码中,  $E_{\text{average}}(c-1)$  在上一周期结束的时候将节点的平均能量信息传递给簇头节点, 簇头节点再把本身的能量剩余信息及其成员节点的能量信息传递给 sink 节点, sink 节点计算整个 WSN 节点的能量和, 从而决定下一周期的簇头节点布局。

当 sink 节点把整个 WSN 的簇头节点分布信息传递给上一周期的簇头节点后, 各个簇头节点都会维护一个候选簇头节点集合  $H_i, \text{self\_Clusterhead}$ 。算法 1 的第 1-7 步说明了成员节点成为候选簇头节点的依据原则, 当一个节点的  $H_i, E / E_{\text{average}}(c-1)$  小于  $\tau(1 - p * \text{mod}(c/\text{round}(1/p)))$  时, 其就会成为候选簇头节点, 这些候选簇头节点集合在一起就成为  $H_i, \text{self\_Clusterhead}$ 。算法中的第 8-10 步说明了候选簇头节点以半径为  $R_h$  的大小发射自己的信息, 这些信息包含本身的唯一标识  $ID$  和剩余能量  $H_i, E$ 。算法中的第 11-15 步讲述了候选簇头节点  $H_i$  发射完信息后会等待着其他候选簇头节点的信息广播, 当候选簇头节点  $H_i$  到其他候选簇头节点的距离比发射半径  $R_h$  小时, 这个候选簇头节点就会成为  $H_i$  的邻居候选簇头节点, 加入到  $H_i$  的邻居候选簇头节点集合, 即  $H_i, \text{self\_Clusterhead}$ , 这个过程一直循环到整个 WSN 完毕。算法 17-22 步说明了候选簇头节点  $H_i$  的剩余能量会和其邻居候选簇头节点的剩余能量循环相比较, 直至选出剩余能量最大的那个候选簇头节点, 例如候选簇头节点  $H_i$  的剩余能量最大, 则候选簇头节点  $H_i$  宣布成为簇头节点。簇头节点  $H_i$  将广播包含自身唯一标识  $ID$  的簇头节点的信息  $\text{Send a Be\_Clusterhead\_msg}(ID)$ 。算法 23-32 步描述了在以半径为  $R_h$  的圆形区域内的候选簇头节点收到簇头节点  $H_i$  的广播信息时, 这些候选簇头节点就会向簇头节点  $H_i$  发送去除候选簇头节点职位的请求, 簇头节点  $H_i$  收到它们的信息后会删除这些候选簇头节点的竞选信息, 这些候选簇头节点就会变为簇头节点  $H_i$  的普通成员节点, 而其他在休眠状态中的节点开始苏醒, 至此一个新的簇就形成了。

算法 1 中, 伪代码会一直循环找出剩余能量最大的候选簇头节点(以半径为  $R_h$  的圆形区域内), 然后这个区域内的剩余能量最大的节点宣布成为簇头节点。如图 4 所示, 一个区域内包含 3 个候选簇头节点  $H_1, E, H_2, E, H_3, E$ , 假设它们的剩余能量大小为  $H_3, E < H_2, E < H_1, E$ , 则  $H_1$  宣布成为

簇头节点,簇头节点发布  $Be\_Clusterhead\_msg(ID)$  后,其他两个候选簇头节点向簇头节点  $H_1$  发射  $Quit\_msg$  信号,簇头节点  $H_1$  收到信号后就会去除它们的邻居簇头信息,形成以自身为簇头节点的簇。

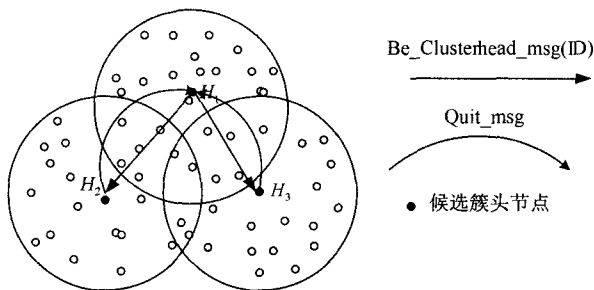


图4 簇头节点轮转选择算法图示

当 WSN 中的簇头节点确定后,簇头节点会给成员分配 TDMA 时隙表,根据 TDMA 时隙表,成员向簇头发送消息。簇头综合成员发射的信息,并以多跳方式传输给 sink 节点。

### 2.2.4 簇间多跳路由算法

当簇头竞争结束后,簇头节点会发布当选信息,成员节点和其邻近簇头节点都会收到这些信息,这些信息包括簇头节点  $ID$ 、当前剩余能量和它到 sink 节点的距离。然后邻近簇头节点会根据收到的信号强度计算与此簇头节点的距离  $d$ ,并将这个信息存储起来。其中  $d_0$  表示某一阈值,当簇头节点到 sink 节点的传输距离  $d_{tosink}$  小于临界值  $d_0$  时,两个簇头节点可以直接进行数据传输,不再需要中间节点的多跳方式,就可以消耗比较少的能量,即簇头节点  $H_i$  可以直接单跳到 sink 节点;当传输距离  $d_{tosink}$  大于临界值  $d_0$  时,簇头节点  $H_i$  需要中继簇头节点  $H_j$  再传输到 sink 节点,这样簇头节点  $H_i$  消耗的能量要小于直接发射到 sink 节点消耗的能量。

$d_0$  这一阈值比较抽象,为了考虑到簇头节点和中继簇头节点的剩余能量和链路消耗代价,我们把  $d_0$  转化为一个有关能量的权值。

定义 1(权值)

$$W_j = \frac{S_j \cdot E}{E\_average(c-1)} + \frac{d_{tosink}^2(j) + d_{toij}^2(i, j)}{d_{tosink}^2(i)}$$

式中,  $d_{tosink}^2(j) + d_{toij}^2(i, j)$  表示簇头节点  $H_i$  需要中继簇头节点  $H_j$  再传输到 sink 节点,  $d_{toij}^2(i, j)$  代表簇头节点到簇头节点的距离,  $d_{tosink}^2(j)$  代表簇头节点到 sink 节点的距离;  $d_{tosink}^2(i)$  表示簇头节点  $H_i$  可以直接单跳到 sink 节点;  $d_{tosink}^2(i)$  代表簇头节点到 sink 节点的距离。

簇头节点  $H_i$  会把簇头节点  $H_j$  加入到其邻近簇头节点集合中,计算其权值。把邻近簇头节点的权值进行排序并存储在  $W_i.table$ , 权值最大的簇头节点就会被选为下一个路由,如果权值相等则会根据  $ID$  来选择下一跳路由。以上的簇间动态多跳路由方案可由算法 2 实现。

算法 2 簇间动态多跳路由算法

1.  $W_j = Initial$ ;
2. While be Clusterhead = true do
3. for  $H_j \in H_i$ . self\_Clusterhead
4.  $W_j = \frac{S_j \cdot E}{E\_average(c-1)} + \frac{d_{tosink}^2(j) + d_{toij}^2(i, j)}{d_{tosink}^2(i)}$ ;
5.  $W_i.table[W_j]$ ;
6.  $sort(W_i.table[W_j])$ ;
7. End

8. if have  $Max\_W_j\_RoutingNode$
10.  $Max\_W_j\_RoutingNode$  is next oneRouting;
11. else if  $W_j = W_{j+1}$
12. nextOneRouting( $ID$ );
13. End
14. End While

算法符号说明:

- $W_j$ : 簇头节点权值;
- $W_i.table[W_j]$ : 权值集合;
- $sort(W_i.table[W_j])$ : 权值按大小排序;
- $Max\_W_j\_RoutingNode$ : 最大权值簇头节点;
- $nextOneRouting(ID)$ : 根据  $ID$  选择下一跳路由。

在算法 2 中,第 1—7 步讲述了簇头节点  $H_i$  的邻近簇头节点权值的计算,并且把全部邻近簇头节点的权值存储到集合  $W_i.table[W_j]$  中,对集合中的权值按大小进行排序,得到最大权值的邻近簇头节点  $Max\_W_j\_RoutingNode$ 。第 8—14 步中,若存在最大权值的邻近簇头节点则为下一跳路由,如果权值相等则会根据  $ID$  来选择下一跳路由。

## 3 仿真分析比较

为了对本网络模型的簇头节点分布、簇头节点能耗和总体能耗均衡性进行分析,采用 Eclipse 编写模拟程序。为简单起见,不计较链路中可能发生的丢包错误,只统计实验中节点收发数据的耗能、广播数据耗能和路由耗能。而且实验中将本模型算法和经典的 LEACH 协议算法进行比较,具体的实验参数见表 1。

表 1 模拟参数

参数	取值
R (米)	100
初始能量 (焦耳)	0.5
节点数	200
$d_0$ (米)	87

下面对簇头节点数目稳定性、簇头节点能耗均衡性、网络能量利用效率和网络整体能量消耗均衡性进行比较分析。

### (1) 簇头节点数目稳定性

在网络结构拓扑稳定的情况下,一个 WSN 应该生成数目比较稳定的簇头节点。在这个实验中,每种协议进行 100 轮的簇头节点轮转周期,簇头节点的数目如图 5 所示。图 5 显示了 LEACH 仿真时的簇头数目设定为 8,轮转数的期望值为 17;而图 6 显示了本网络模型下的仿真,其簇头节点期望值为 8 时,轮转数期望值为 30。而且 LEACH 簇头节点数目波动幅度非常大,而本模型中簇头节点数目波动比较平坦,这是因为 LEACH 机械性地采用阈值机制产生簇头节点,而本模型算法中采用竞选机制有效控制了簇头节点波动幅度,所以本模型算法中簇头节点数目稳定性较好,形成的簇可靠稳定。

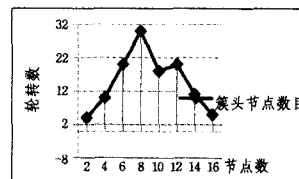
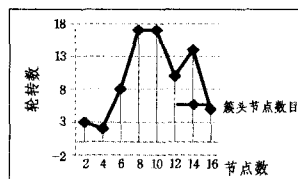


图 5 LEACH 簇头数目分布图示 图 6 本模型中簇头数目分布图示

## (2) 簇头节点能耗均衡性

簇头节点能耗均衡性的比较就是对每轮节点的能量消耗进行比较。在本网络模型中簇头节点的能量消耗占整个 WSN 的主要部分,因此簇头节点的能耗比较首先要比较簇头节点在每轮中所消耗能量的总和。在仿真实验中,每个协议算法随机选出 40 轮,LEACH 仿真时的簇头数目设定为 8,本模型网络中簇头节点预设为 8,即在相同的簇头节点下进行两个协议算法的比较。在实验中收集每轮簇头节点的能量消耗总和。如图 7 所示,本模型簇头节点能耗明显比 LEACH 要少,而且曲线也较平坦。LEACH 的簇头控制不稳定,而且簇头分布不均匀,从而导致 LEACH 簇头节点消耗较大能量。

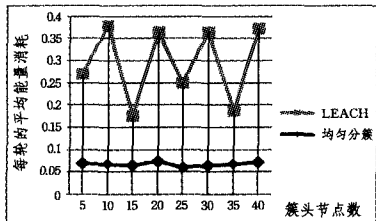


图 7 簇头节点能量消耗比较图示

## (3) 网络能量利用效率

在仿真实验中,从两个方面比较两种协议的网络能量利用效率,一是传感器节点利用率;二是网络总体能量剩余。图 8 和图 9 显示了两种利用率的比较图示。

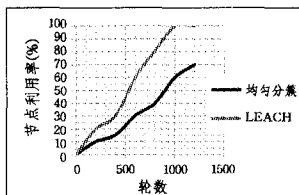


图 8 网络节点利用率图示

这里的传感器节点利用率是指在一定轮数内节点死亡的数量占总节点的利率。从图 8 中可以看出 LEACH 的节点死亡率是比较高的,基本上是本模型算法的 2 倍,从而显示基于同心环分簇网络的寿命比 LEACH 协议的要长。网络总体剩余能量是指在一定轮数后网络剩余的能量。图 9 显示了两者的能量剩余情况,本模型网络的剩余能量远远大于 LEACH 网络剩余能量。

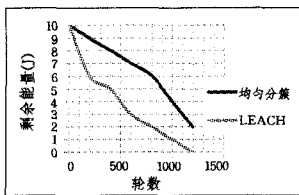


图 9 网络剩余能量图示

## (4) 网络整体能量消耗均衡性

本次实验共进行了 6 次,根据第一个节点死亡时间来计算网络寿命。根据图 10 可以看出 LEACH 网络寿命十分不稳定,而本模型网络的寿命长短是比较稳定的,这是因为 LEACH 网络簇头节点分布不均匀,簇头位置不稳定可靠,且簇头多跳路由耗能比较大,从而寿命长短波动幅度大。如果没有特别指出,在模拟实验中的网络寿命以 1000 轮次为单位(例如图 10 中的纵坐标的网络寿命 4 是指 4 \* 1000 轮次)。

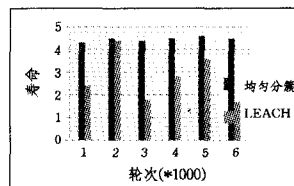


图 10 网络寿命比较示意图

**结束语** 本文致力于研究避免 WSN 中的能量空洞现象。首先对国内外能量空洞的避免机制和方法进行了分析,接着在同心环模型的基础上对逐层分环构思进行了扩展,提出了一种基于同心环分簇网络模型的 WSN 能量空洞避免方法,对网络结构模型和能量消耗模型进行了描述,并在此基础上,提出了簇头节点位置初始化、簇头轮转选择算法和簇间多跳路由算法。通过仿真实验与经典的 LEACH 协议进行比较,得出本文的算法在簇头节点数目稳定性、簇头节点能耗均衡性、网络能量利用效率和网络整体能量消耗均衡性上有着很大的优势。

## 参考文献

- [1] Heinzelman W R, Chandrakasan A, Balakrishnan H. Energy efficient communication protocol for wireless microsensor networks [C]//The 33rd Hawaii International Conference on System Sciences. Hawaii, 2000
- [2] Younis O, Fahmy S. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks[J]. IEEE Transactions on Mobile Computing, 2004, 3(4): 366-379
- [3] Chen G, Li C F, Ye M, et al. An Unequal Cluster-Based Routing Strategy in Wireless Sensor Networks[J]. Wireless Networks (JS), 2009, 15(2): 193-207
- [4] Li J, Mohapatra P. Analytical modeling and mitigation techniques for the energy hole problems in sensor networks[J]. Pervasive and Mobile Computing, 2007, 3(3): 233-254
- [5] Qlariu S, Stojmenovic I. Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks uniform distribution and unifolTS reporting[C]//Proceedings of the IEEE IV FOCOM. New York: IEEE Communications Society, 2006: 1-12
- [6] 任丽婕, 郭忠文, 唐瑞春. 无线传感器网络中基于能量平衡的部署算法[J]. 中国海洋大学学报, 2008, 38(5): 841-844
- [7] 王达山, 黄刘生, 徐宏力, 等. 基于矢量的无线传感器网络能量有效配置算法[J]. 计算机研究与发展, 2008, 45(4): 626-635
- [8] Chiang Mu-Huan, Byrd G T. Neighborhood-Aware Density Control in Wireless Sensor Networks[C]//The 2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing. 2008
- [9] 刘安丰, 任炬, 徐娟, 等. 异构传感器网络能量空洞分析与避免研究[J]. 软件学报, 2012, 23(9): 2438-2448
- [10] Liu Tao. Avoiding Energy Holes to Maximize Network Lifetime in Gradient Sinking Sensor Networks[M]. Wireless Personal Communications, 2012: 581-600
- [11] Matrouk K, Landfeldt B. RETT-gen: A globally efficient routing protocol for wireless sensor networks by equalising sensor energy and avoiding energy holes[J]. Ad-hoc Networks, 2009, 7(3): 514-536