

# 无线传感器网络的簇头间距自适应 HDA-LEACH 算法

张 昱

ZHANG Yu

浙江大学 信息学院 信电系, 杭州 310027

Dept. of Information &amp; Electronic Engineering, College of Information, Zhejiang University, Hangzhou 310027, China

E-mail: zangwill@263.net

ZHANG Yu. Cluster head distance adaptive HDA-LEACH algorithm in WSN. Computer Engineering and Applications, 2007, 43(30): 124-127.

**Abstract:** In this paper, to solve the problem of sensing information redundancy and more power waste on cluster heads when some cluster heads are too near in LEACH, an improved HDA-LEACH algorithm is presented, in which the remaining node power and the cluster transmission phase are considered, and two cluster head distance adaptive topology control improvement schemes are presented, one is to reassign the cluster heads, and the other is to disqualify the less power cluster head in the two too near cluster heads. Both HDA-LEACH and LEACH are simulated in MATLAB. The simulation results prove that HDA-LEACH algorithm is much better than LEACH algorithm in network lifetime and LBF.

**Key words:** cluster head distance adaptive; clustering routing algorithm; network lifetime; LBF

**摘 要:** 针对 LEACH 算法中当某些簇头间距过近时会导致传感信息冗余和簇头多余能耗的问题进行了改进, 提出了 HDA-LEACH 算法, 增加了对节点剩余能量以及簇传输阶段的考虑, 并且从两方面提出了簇头间距自适应的拓扑控制改进方案: 一是重新分配簇头以消除簇头间距过近情况, 二是取消两太近簇头中能量较低簇头的簇头资格。用 MATLAB 对 LEACH 算法和 HDA-LEACH 算法进行仿真, 证实 HDA-LEACH 算法在网络生存时间和负载平衡程度上比 LEACH 算法有了明显提高。

**关键词:** 簇头间距自适应; 分簇路由算法; 网络生存时间; 负载平衡程度

**文章编号:** 1002-8331(2007)30-0124-04 **文献标识码:** A **中图分类号:** TP393

关于无线传感器网络的研究近年来变得日益重要。无线传感器网络是集信息采集、信息处理、信息传输于一体的综合智能信息系统, 具有广阔的应用前景<sup>[1-4]</sup>。在无线传感器网络中的传感器节点分布在一个较大的地理区域内, 它们建立起一个无线自组织网络, 必须要采用合适的路由协议才能保证整个网络的正常通信。传感器节点通常由电池供电, 且计算能力和存储空间都有限, 传感器网络要尽可能在十分有限的电池能源容量条件下用最低的能量消耗, 进行较大的数据处理及传输, 因此必须要有一个好的路由协议以尽量提高其负载平衡程度, 同时尽量延长其网络生存时间。

无线传感器网络的路由协议可以分成平面路由协议和分簇路由协议两种<sup>[5]</sup>。由于平面路由协议需要维持较大的路由表, 占据较多的存储空间, 因而并不适合在大规模网络中采用。分簇路由算法可以在一定程度上解决这个问题。LEACH 算法是一种比较成熟且常用的分簇路由算法, 但 LEACH 算法有节点传输能耗大、网络生存时间短、负载平衡程度低等缺陷。针对 LEACH 算法中当某些簇头间距过近时会导致传感信息冗余和簇头多余能耗的问题, 本文提出了一种考虑节点剩余能量并且着重考虑簇头间距自适应拓扑控制的 HDA-LEACH 算法, 更接近无线传感器网络的实际情况, 并且经 MATLAB 仿真表明, HDA-LEACH 算法能明显提高网络生存时间和负载平衡程度。

## 1 分簇路由算法原理

在分簇路由算法中, 网络节点按照不同的分簇算法分成相应的簇。簇中每个节点完成的功能是不同的, 有些节点被赋予一些特别功能, 如簇头节点管理和维护本簇范围内节点, 负责簇内节点通信, 同时为簇间节点通信提供合适的路由信息<sup>[5]</sup>。

### 1.1 LEACH 算法

低能量自适应分簇分级机制(LEACH 算法)是分簇路由算法中一种专门用于无线传感器网络的分簇协议, 其目标是尽可能将能耗均匀分配到各个节点上, 以延长整个网络的寿命<sup>[6]</sup>。在 LEACH 算法中, 节点自组织成不同的簇, 每个簇只有一个簇头。所有非簇头节点将自己的数据发给所属簇的簇头节点, 为减少冗余数据的传输, 簇头节点在数据融合后将数据发送给远方的基站接收器。这样, 每个非簇头节点都只需要知道自己所属簇的簇头信息即可; 簇头也只需要维持很小的路由表。在 LEACH 算法中, 为了避免簇头能量消耗过快, 每个节点须轮流担任簇头。因此 LEACH 算法的实现分成若干轮, 每一轮又可分成簇形成阶段和簇传输阶段, 而簇传输阶段远远长于簇形成阶段。在簇形成阶段, 先为之前还没有担任过簇头的每个节点分别生成 0-1 之间的随机数, 如果生成的随机数小于阈值  $T$ , 那么这个节点就被选为簇头, 这里阈值  $T$  的大小由式(1)确定:

$$T = \begin{cases} p/(1-p \cdot (r \bmod (1/p))) & n \in G \\ 0 & \text{其它} \end{cases} \quad (1)$$

其中,  $p$  是网络中簇头占有所有节点的比例,  $r$  是目前正在进行的第  $r$  轮,  $n$  是表示某节点,  $G$  是到第  $r$  轮之前尚未当过簇头的节点集合。节点被选为簇头后, 就向外发送簇头广播信息。而其它节点根据收到的簇头广播信息的信号大小决定要加入哪个簇, 并向决定加入的簇的簇头发送加入簇的请求。簇头在收到请求后将该节点加入自己的路由表并为每个节点设定一个 TDMA 时间表, 再将该表发送给所有簇内节点。此后的簇传输阶段, 这些分布式的节点即按照该 TDMA 时间表、以一跳式或多跳式的路由拓扑结构进行数据传输。每隔一定时间, 整个网络重新进入簇形成阶段开始新一轮的簇头选举过程。

和平面路由算法相比, LEACH 算法可以延长将近 30% 的网络生存时间。但是, 由于 LEACH 算法中簇头的产生具有极大的随机性, 可能会出现部分簇头相距过近的情况, 这将大大增加节点的传输能耗, 故不能有效地延长网络生存时间; 而且由于簇头选举的随机性使得网络的簇头需要负担的节点数不同, 加重了个别簇头节点的负担, 使得网络的负载均衡程度下降<sup>[7]</sup>。

## 1.2 分簇路由算法中的相关参数

在传感器网络中, 评价一个分簇算法好坏的参数有网络生存时间、负载均衡程度、覆盖率、节点担任簇头时间的公平度等等。其中, 网络生存时间和负载均衡程度是两个比较重要的参数。

当一个传感器节点的能量小于零时, 即可认为该节点死亡; 而当网络中出现至少一个节点能量小于零时, 即可认为该网络死亡。从网络初始建立起, 到网络中第一个节点死亡的时间, 定义为网络生存时间。采用分簇算法进行分簇的能耗越少, 越有利于网络生存时间的延长。

在分簇路由算法中, 除了分簇时的能量消耗外, 每轮中簇覆盖节点的平均程度也会影响网络的生存时间。而且由于簇中包含节点数目的不平衡会导致不同簇之间数据的容量不同, 因此簇的负载均衡程度也是分簇路由算法中衡量簇性能的重要标准之一。负载均衡程度可以由负载平衡因子  $LBF$  参数来衡量。负载平衡因子  $LBF$  的计算公式如下:

$$LBF = \frac{head\_num}{\sum_{i=1}^{head\_num} (x_i - \mu)^2} \quad (2)$$

其中,  $head\_num$  为本轮簇头总数 (即本轮簇总数),  $x_i$  为第  $i$  个簇包含的节点个数,  $\mu$  为本轮各簇中平均包含的节点个数。  $LBF$  的数值越大说明簇的负载越平衡<sup>[9]</sup>。

## 1.3 分簇路由算法的能耗模型

在分簇路由协议中, 在保证通信质量的前提下, 节点消耗的能量越少越好。在无线传感器网络中, 传感器节点需要进行的运算和存储都较少, 其所需能量相对于传输数据所需能量也要小得多。因此, 无线传感器网络的网络生存时间主要取决于数据传输, 即簇传输阶段。下面基于文献[8], 并结合本文仿真的无线传感器网络实际情况对该文献结果进行修正, 分别给出簇形成阶段和簇传输阶段的节点能耗模型。

在簇形成阶段, 簇头在本簇中的耗能为:

$$k(E_{tran} + E_{amp} \cdot d_{cover\Sigma}^2) + kE_{tran}(node\_num - 1) + k(E_{tran} + E_{amp} \cdot d_{cover}^2) \quad (3)$$

其中,  $k$  为所发送数据包的比特数,  $E_{tran}$  为节点发送 1 bit 数据

消耗的能量,  $E_{rec}$  为节点接收 1 bit 数据消耗的能量, 一般有  $E_{tran} = E_{rec}$ ,  $E_{amp}$  是放大器消耗能量,  $node\_num$  为本簇内节点总数 (包括本簇簇头),  $d_{cover\Sigma}$  为整个网络的覆盖范围,  $d_{cover}$  为本簇覆盖的最远距离。分析如下: 该簇头发到网络中其它所有节点的簇头信息广播包所耗能量为  $k(E_{tran} + E_{amp} \cdot d_{cover\Sigma}^2)$ ; 该簇头收到本簇内节点发来的加入簇消息数据包所耗能量为  $kE_{tran}(node\_num - 1) = kE_{tran}(node\_num - 1)$ ; 该簇头发到本簇内节点的 TDMA 时间表所耗能量为  $k(E_{tran} + E_{amp} \cdot d_{cover}^2)$ 。而簇头作为普通节点在其它某一簇中的耗能为:

$$kE_{rec} = kE_{tran} \quad (4)$$

这是因为该簇头只接收一次来自于该其它某簇簇头的簇头信息包; 但由于不加入该簇, 所以不回应, 也就没有再发与再收了。

在簇形成阶段, 普通节点在其所属簇中的耗能为:

$$kE_{rec} + k(E_{tran} + E_{amp} \cdot d^2) + kE_{rec} = kE_{tran} + k(E_{tran} + E_{amp} \cdot d^2) + kE_{tran} \quad (5)$$

其中,  $d$  为该节点到本簇簇头的距离, 其它符号意义同上。分析如下: 该普通节点收到本簇簇头发来簇头信息广播包所耗能量为  $kE_{rec} = kE_{tran}$ ; 该普通节点发给本簇簇头加入簇的请求数据包所耗能量为  $k(E_{tran} + E_{amp} \cdot d^2)$ ; 该普通节点收到本簇簇头发来的 TDMA 时间表所耗能量为  $kE_{rec} = kE_{tran}$ 。而普通节点在非本簇的其它某一簇中耗能为:

$$kE_{rec} = kE_{tran} \quad (6)$$

这是因为该节点只接收一次来自于该其它某簇簇头的簇头信息包。

在簇传输阶段, 当采用一跳式路由算法, 即普通节点要发送数据给基站, 需通过该节点所属簇的簇头来进行中转, 即节点先发送数据到簇头, 然后簇头再将数据转发到基站去, 这时, 某簇内一普通节点要发送数据包到基站去所需耗能为:

$$k(E_{tran} + E_{amp} \cdot d^2) \quad (7)$$

而这时某簇内的簇头所需耗能则为本簇内要发送数据包给基站的所有节点其发送的数据包经过簇头中转时所消耗的簇头能量之和, 下式为某簇内一普通节点通过簇头的中转发送数据包到基站去时簇头所需耗能:

$$kE_{rec} + k(E_{tran} + E_{amp} \cdot d'^2) = kE_{tran} + k(E_{tran} + E_{amp} \cdot d'^2) \quad (8)$$

其中,  $d'$  为该簇头到基站的距离。分析如下: 该簇头接收普通节点发来数据包所需耗能为  $kE_{rec} = kE_{tran}$ , 该簇头转发数据包到基站去所需耗能为  $k(E_{tran} + E_{amp} \cdot d'^2)$ 。

## 2 簇头间距自适应 HDA-LEACH 分簇算法

在本文基于 LEACH 算法提出的 HDA-LEACH 算法中, 首先考虑了节点剩余能量和节点成为簇头容易程度之间的关系。和 LEACH 算法一样, HDA-LEACH 算法依然分成多轮, 每轮分成簇形成阶段和簇传输阶段, 而且簇传输阶段在时间上远远长于簇形成阶段。在簇形成阶段, 首先生成一个阈值  $T'$ , 其定义如式(9)所示。

$$T' = \begin{cases} 1-p/(1-p \cdot r) & n \in G \\ 0 & \text{其它} \end{cases} \quad (9)$$

其中各参数的意义同式(1)。然后  $G$  中的每个节点根据自己的剩余能量产生一个 0-1 之间的随机数, 剩余能量越大的节点越容易产生较大的随机数。生成的随机数大于阈值  $T'$  的节点将成为簇头并向外发送簇头信息; 生成的随机数不大于阈值  $T'$  的节点将成为普通节点, 根据自己的节点属性发送加入簇请



求。簇头在收到节点加入簇请求后,为自己选定下来的每个簇内节点分配 TDMA 时间表,并将该时间表发送给所有簇内节点。在簇传输阶段,本文假设簇内节点按照上述 TDMA 表,以一跳式路由拓扑结构进行分时的数据传输:普通节点要发送数据给基站,需通过该节点所属簇的簇头来进行中转,即节点先发送数据到簇头,然后簇头再将数据进行数据融合后转发到基站去。每隔一段时间,整个网络即重新进入簇形成阶段开始新一轮的簇头选举过程。本文假设上一轮簇传输阶段结束后,若网络还没有死亡,则立即进入下一轮簇形成阶段。这里值得说明的是,在式(9)中直接使用了  $r$  而没有像式(1)中那样使用  $r \bmod (1/p)$ ,这是因为:在本文算法中以回合为单位来考虑分簇过程,其中一个回合包含所有节点都当过一次簇头的若干轮,在每一个回合中使用式(9)来确定每一轮时相应的阈值,经过一回合仿真后记录该回合所经过的轮数;然后将  $r$  清零,重新开始下一回合仿真并计下一回合的轮数;依此类推循环下去,直到网络死亡,最后统计所经各个回合的总轮数即得网络生存时间参数的值。

在簇形成阶段中,如果各个簇头之间的间距太近,一方面由于两太近簇头所能感到的信息相差不大,会造成其在传感信息上的冗余;另一方面在后面的簇传输阶段中将分别以这些簇头作为节点到基站发送数据的中转站,如簇头相距太近则会浪费多余的耗能,因为毕竟它们相隔很近,只以其中一个簇头的能量就多余了,并且这样还容易造成簇分布的不均匀,如造成两相邻簇靠得太近。本文针对上述传感信息冗余和簇头多余能耗问题提出了两种簇头间距自适应拓扑控制改进方案:一是重新分配簇头直至所有簇头均不会间距太近(以下简称“方案一”),二是取消两太近簇头中能量较小簇头的簇头资格使其成为普通节点参与分簇形成(以下简称“方案二”),从而可以有效避免上述问题,并且有利于提高簇分布的均匀度。本文中采用的簇头间距太近的指标判据为:

$$D < \varepsilon L \quad (10)$$

其中,  $D$  为两簇头间的间距,  $\varepsilon$  为由网络实际情况决定的比例因子,  $L$  为整个无线传感器网络所覆盖范围的最大可能距离(这里假设为整个网络矩形范围的对角线长度)。这里需要说明的是,事实上由于簇头比例  $p$  变化就意味着簇密度及簇数目的变化,因此当  $p$  较小时由于簇比较疏所以簇头间距满足式(10)而出现簇头过近的概率较小,而当  $p$  较大时由于簇比较密所以簇头间距满足式(10)而出现簇头过近的概率较大。可见,簇头间距过近的判别准则式(10)中的比例因子  $\varepsilon$  其实最好是一个随簇头比例  $p$  而自适应变化的参量,当  $p$  较小时  $\varepsilon$  应较大,而当  $p$  较大时  $\varepsilon$  应较小,这样能够保证簇头间距过近出现的概率对于所有的簇头比例  $p$  值都是基本一致的。本文经过对比性仿真试验选择  $\varepsilon$  为 0.1,比较符合网络实际情况,而且最后的仿真结果也表明这时网络生存时间和负载平衡程度等参数相对于原来的 LEACH 算法提高的效果均为最优。

在簇传输阶段中,如前所述,采用一跳式路由算法:普通节点要发送数据给基站,需通过该节点所属簇的簇头来进行中转,即节点先发送数据到簇头,簇头再将数据转发到基站去。在本文中,簇传输阶段的工作场景设置为:所有普通节点都发送一定长度的数据包给基站。

本文提出的簇头间距自适应的 HDA-LEACH 分簇算法的流程图如图 1 所示。

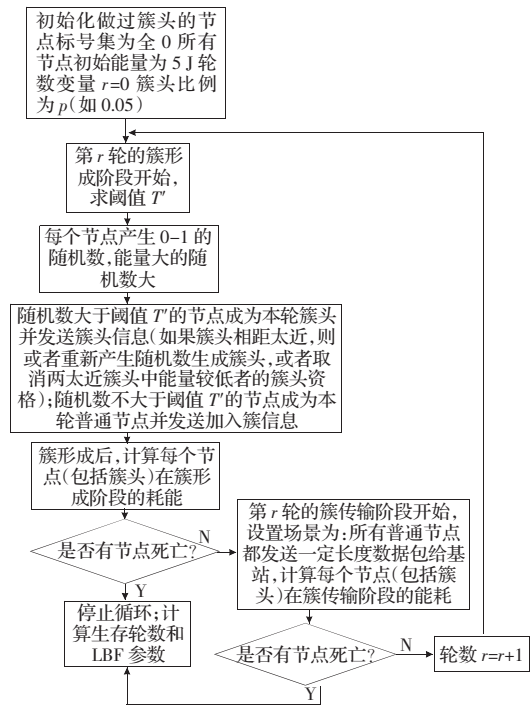


图1 HDA-LEACH 算法流程图

### 3 仿真及讨论

用 MATLAB 生成一个随机分布的无线传感器网络节点图,在  $100 \times 100$  的范围内随机安放 200 个节点,如图 2 所示。这里,假定每个节点最初都具有 5 J 的初始能量,每个节点接收或发送数据需要消耗能量为  $E_{min}=5$  nJ/bit,放大器消耗能量为  $E_{amp}=10$  pJ/bit/m<sup>2</sup>,且每个数据包的大小固定为  $k=50$  bit。

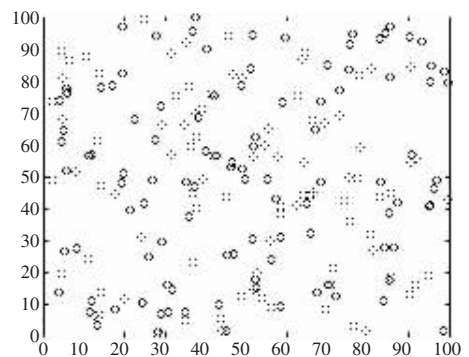


图2 随机产生的无线传感器网络节点图

根据以上参数定义,利用 MATLAB 仿真出在不同簇头比例( $p$  从 0.03 到 0.15)下既考虑簇形成阶段又考虑簇传输阶段时的 LEACH 算法和 HDA-LEACH 算法全过程,例如 HDA-LEACH 算法的某一轮分簇后的节点分布图如图 3 所示,其中五角星形标记位置是簇头,虚线圆圈形状是以簇头为中心的各簇覆盖范围,而其它如方形、三角形等标记位置为普通节点(注:在簇头位置处既画了五角星又画了本簇节点的标记,因为簇头本身也是节点);同时计算该无线传感器网络可以达到的网络生存时间和负载平衡因子 LBF,其结果分别如图 4、图 5 所示。由图 4、图 5 可见,方案一重新分配簇头方案对于提高生存轮数的效果比较明显,平均能够提高 30%~40%,但对于 LBF

参数改善的效果则不明显;而方案二取消一簇头资格方案对于提高生存轮数和 LBF 参数的效果都比较明显,生存轮数最多能够提高 60%,LBF 参数最多能够提高 90%。考虑到方案一显然会增加簇形成阶段所花费的时间及能耗,而方案二简单、计算量小且不会导致时间和能耗的过多浪费,因此方案二具有更好的可行性和实用性。而且图 4 的结果与文献[9,10]中关于每节点能耗与簇数目关系的理论分析结果基本相符。图 6 为方案二的生存轮数和 LBF 两个参数的仿真结果拟合曲线图,采用四次多项式曲线拟合算法。从图 6 中可以更加清楚地看到方案二对于提高生存轮数和 LBF 参数的明显作用。

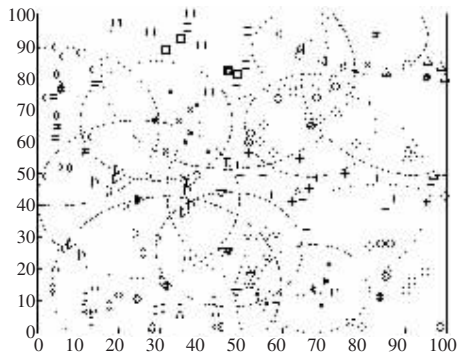
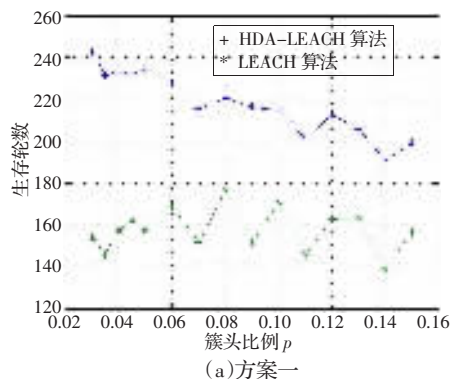
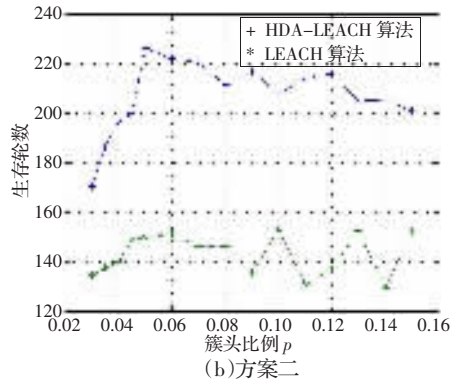


图3 HDA-LEACH 算法某轮分簇后节点分布图



(a)方案一

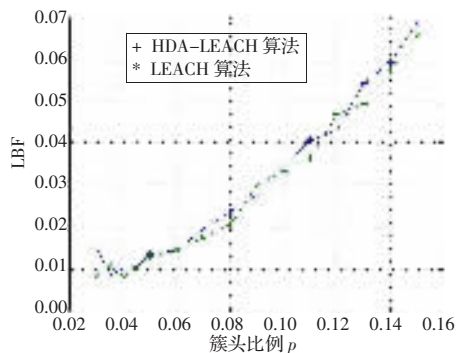


(b)方案二

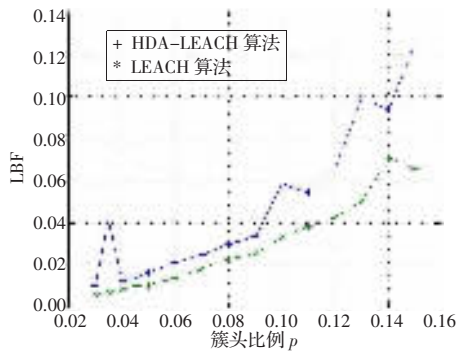
图4 HDA-LEACH 算法和 LEACH 算法中网络生存时间与簇头比例的关系

#### 4 结论

本文针对传统的 LEACH 算法中当某些簇头间距过近时会导致传感信息冗余和簇头多余能耗的问题进行了改进,基于节点剩余能量提出了 HDA-LEACH 算法,并进行了簇形成阶段和簇传输阶段的完整过程的 MATLAB 仿真。仿真结果表明,当两簇头间距过近时取消两太近簇头中能量较低簇头的簇头资格的 HDA-LEACH 算法改进方案能够增大簇的负载均衡程

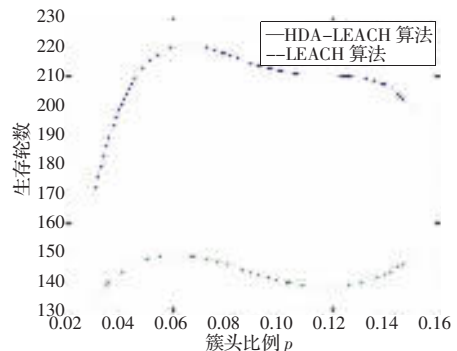


(a)方案一

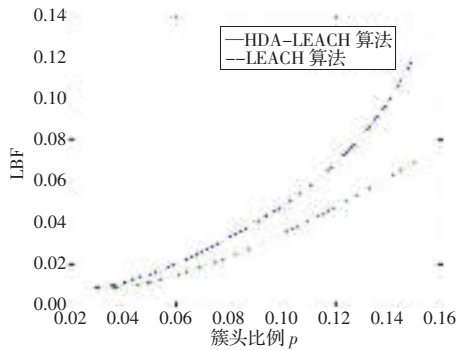


(b)方案二

图5 HDA-LEACH 算法和 LEACH 算法中 LBF 参数与簇头比例的关系



(a) HDA-LEACH 算法方案二和 LEACH 算法中网络生存时间与簇头比例关系拟合曲线图



(b)HDA-LEACH 算法方案二和 LEACH 算法中 LBF 参数与簇头比例关系拟合曲线图

图6 HDA-LEACH 算法方案二和 LEACH 算法中网络参数与簇头比例关系拟合曲线图

度,减少簇形成及簇传输整个过程中的能耗,明显延长网络生存时间。(收稿日期:2007 年 3 月)