

传感器网络簇头选举与调度策略研究

柳 絮, 李金宝, 纪守领, 郭龙江, 惠 丽

(1. 黑龙江大学计算机科学技术学院, 黑龙江哈尔滨 150080;

2. 黑龙江省数据库与并行计算重点实验室, 黑龙江哈尔滨 150080).

摘 要: 为了最大化分簇网络的生存周期, 提出基于线性规划的簇头选举策略(LPCHS)和基于簇头时间比的簇头调度算法(CHSA). LPCHS根据数据流守恒约束和能量约束等条件, 建立线性规划方程, 得到簇生存周期、簇中各节点轮为簇头的时间及相应的簇头时间比. CHSA采用多路径路由技术完成簇间数据转发, 得到基于簇头时间比的簇头调度方案.

关键词: 传感器网络; 簇头选举; 簇头调度

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2010) 08-1770-06

A Cluster Head Selecting and Scheduling Strategy in Sensor Networks

LIU Xu, LI Jin-bao, JI Shou-ling, GUO Long-jiang, HUI Li

(1. School of Computer Science and Technology, Heilongjiang University, Harbin, Heilongjiang 150080, China;

2. Key Laboratory of Database and Parallel Computing of Heilongjiang Province, Harbin, Heilongjiang 150080, China)

Abstract: A cluster head selecting algorithm based on linear programming called LPCHS and a cluster head scheduling algorithm based on cluster head time ratio called CHSA are proposed to maximize the networks lifecycle. LPCHS constructs a LP equation based on the data flow conservation, energy constraint etc, and obtains the cluster lifecycle, the time of a node being a cluster head and the time ratio correspondingly. CHSA can fulfill the data forwarding among clusters by multi-path routing technology, and obtains a cluster head scheduling scheme based on the cluster head time ratio.

Key words: sensor networks; cluster-head selecting; cluster-head scheduling

1 引言

随着通信技术与嵌入式计算技术的发展, 无线传感器网络已经应用到很多领域中, 如国防军事、环境监测和目标跟踪等. 对于典型的传感器网络, sensor节点以多跳转发的方式将感知数据传至 sink节点. 由于节点转发感知数据过程中会消耗大量能量, 如果将监测到的原始数据直接转发至 sink节点, 这些到达 sink节点的原始数据用户未必感兴趣.

为了高效利用网络中的能量资源, 最大限度地减少无效(冗余)数据在网络内的传输, 研究人员提出了分簇的方法^[1-12]. 从簇的工作机制可以看出, 簇头节点由于承担较多的数据预处理及转发工作, 其能量消耗速率会远远高于簇内其他节点. 因此, 如何合理地动态选择簇

头节点, 将会对网络的能量消耗、吞吐量以及生存周期产生很大影响.

为了解决簇头节点产生的热点问题, 并最大化网络生存周期, 本文提出一种半分布式的基于线性规划的簇头选举(Linear Programming Cluster Head Selecting)算法和基于簇头时间比的簇头调度算法(Cluster Head Scheduling Algorithm), 即各簇运行 LPCHS 和 CHSA 算法以完成簇头选举与调度. LPCHS 算法根据网络中的流量约束、能量约束等条件, 建立线性规划方程, 进而得到最大化簇生存周期的簇头选举方案、簇内各节点轮为簇头的时间及其对应的簇头时间比. CHSA 算法采用多路径路由技术完成簇间数据转发任务, 更新 LPCHS 得到的簇生存周期、簇内各节点轮为簇头的时间及其时间比. 实验结果表明: 本文提出的簇头选举与调度方法能够提高系

收稿日期: 2009-07-10; 修回日期: 2010-01-10

基金项目: 国家自然科学基金(No. 60803015); 中国博士后基金(No. 20080430902); 黑龙江省科技攻关(No. GL09A109); 黑龙江省科技计划(No. QC04C40); 黑龙江省自然科学基金(No. F200814); 黑龙江省研究生创新科研项目(No. YJSCX2009-019HLJ); 黑龙江省教育厅重点项目(No. 1154Z1001); 黑龙江省教育厅科学技术研究项目(No. 11531276); 哈尔滨市青年科技创新人才研究专项资金(No. 2009RFQXG080, No. 2008RFQXG107); 黑龙江省博士后基金(No. LRB08-021)

统吞吐量,并且延长网络生存周期.

2 相关工作

在对传感器网络的研究中,分簇的思想被广泛应用.林海^[1]等人提出一种利用能量预测的分簇算法,每个节点根据能量预测模型预测下一周期自己的剩余能量,并得出阈值,根据该阈值决定是否当选为簇头节点.为了使簇负载均衡且能量消耗小,Chiasi 等人^[2]提出了一种能量有效的优化成簇方法,该方法使得每个簇中最多只有一个簇头节点.Yu 等人^[3]提出了一种能量有效的动态分簇技术以最大化网络生存周期.每个节点根据邻居节点信息计算自身成为簇头节点的概率.Qin 等人^[4]研究了簇头选举策略,提出一种基于投票的分簇算法.但是该算法选择簇头时只是简单地考虑邻居节点的能量.

Fernandess 等人^[5]提出一种 K -簇的网络划分方案.虽然该方案能保证簇中任意两节点间的通信跳数小于 K ,但是没有考虑网络的维护方案.Manjeshwar 等人^[6]提出基于簇头节点层次的路由协议(TEEN).该协议按簇头节点到 sink 节点的物理距离进行层次划分,每个簇头节点将达到阈值的感知数据转发给上层簇头节点.Manjeshwar 等人^[7]又在 TEEN 协议的基础上,提出了一种改进的路由协议(APTEEN),该协议在监测超出阈值数据的同时定期地向 sink 发送监测数据.Chen 等^[8]提出一种 K -跳簇均匀分布的算法,最小化簇的数量并使得簇头节点在感知区域中均匀分布.但是并没有考虑网络的动态性,也没有考虑簇头的选举是否合理.

Heinzelman 等人^[9]提出一种基于簇的路由选择协议(LEACH)来进行数据聚集.在选择簇头时,LEACH 只考虑了节点的剩余能量.Younis 等人^[10]提出一种能量有效的分簇方法(HEED),HEED 在 LEACH 的基础上作了一些改善,考虑了簇内能量消耗和通信范围的限制.Kang 等人^[11]改进 LEACH 算法,用阈值方法选出一部分候选节点,再利用 CSMA MAC 协议在候选节点中选出簇头节点,该算法使得网络中簇头节点的分布更均匀.Furuta^[12]从灵活定位理论的角度出发,将 LEACH-C 算法中的分簇问题形式化定义为 p -中值问题,将分簇问题公式转化为无能灵活定位问题(uncapacitated facility location problem).但该方法为集中式算法,可扩展性不好.

上述基于簇的相关研究,在进行簇头选举时,主要采用一些启发式算法或集中式算法,因此很难得到较好的簇头选举与调度方案.

3 簇头选举策略

假设本文研究的传感器网络均匀分布在一个二维平面区域.对于网络中的任一节点 v_i ,通信半径为 R ,初

始能量为 e_i ,数据产生率为 r_i .由于数据的收发将消耗节点绝大部分能量,因此,本文假设节点由于监测数据及计算所消耗的能量忽略不计.同时,假设 v_i 和 v_j 通信时满足以下能量消耗模型^[13]:

$$\text{cost}_{ij} = \alpha + \beta d_{ij}^n \quad (1)$$

其中, α 和 β 是常数, d_{ij} 是 v_i 和 v_j 之间的物理距离, n 为路径衰减系数($2 \leq n \leq 4$), cost_{ij} 为 v_i 到 v_j 传输单位数据消耗的能量,其中 v_j 在 v_i 的通信半径内.

由于本文研究的主要内容是簇头选举与调度策略,因此本文在 3.1 节给出一种简单的成簇策略后,重点研究如何进行簇头的选举与调度,以最大化网络的生存周期.本文提出的簇头选举与调度策略并不局限于 3.1 节的成簇策略得到的簇,通过直接应用或简单修改同样适用于其他成簇策略得到的簇.

3.1 簇的建立

本文考虑的成簇方案是在网络部署完成后,以 sink 节点为参考原点,通过一系列等距的水平和竖直栅格线,将网络划分成若干个栅格区域,每个栅格区域内的节点形成一个簇.现在以图 1 为例说明具体如何成簇.在图 1 中,首先以 sink 节点为原点,通过相交于 sink 节点的水平线 S_0 和竖直线 H_0 ,将网络划分成四个区域;其次分别以 S_0 和 H_0 为基准,以每隔间距 $\sqrt{2}R$,在 S_0 和 H_0 两侧继续通过水平线和竖直线划分网络,直到网络覆盖区域划分完毕为止,划分结果如图 1 所示.

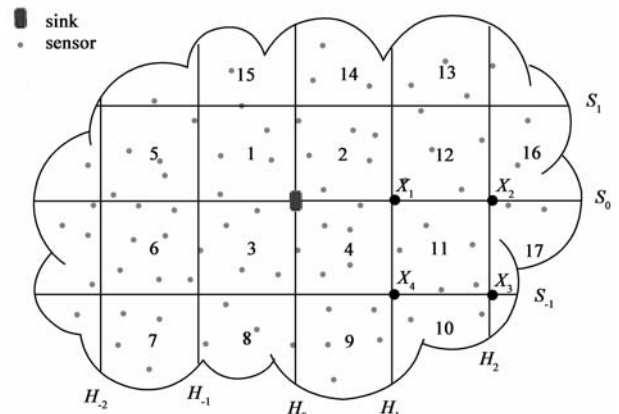


图1 簇的建立

当网络划分完毕后,每个栅格区域内的节点组成一个簇,例如在图 1 中, $X_1 X_2 X_3 X_4$ 组成的四边形内的节点形成一个簇.对于不能构成完整栅格的网络边缘区域,如果该区域的面积大于 R^2 ,那么该区域内的节点构成一个簇;反之,则将该区域内的节点并入相邻的簇中.例如在图 1 中,数字序号(1-17)标出了网络经划分后得到的簇,由于簇 15 对应的区域面积大于 R^2 ,所以构成一个簇;而簇 15 左侧的区域面积小于 R^2 ,则该区域内的节点自发地并入簇 15 或簇 5.当簇建立完毕后,

以 C 记网络中所有簇组成的集合, 即 $C = \{C_i | C_i \text{ 构成一个簇}\}$.

3.2 簇头选举策略 LPCHS

对于任一节点 $v_i \in C_i$, 定义 v_i 的簇内邻居节点集合 $N(i) = \{v_j | v_j \in C_i \text{ 且 } d_{ij} \leq R\}$; v_i 轮为簇头的时间为 T_i . 下面研究节点和链路应该满足的约束条件.

(1) 数据流守恒约束

对于簇内的任一非簇头节点 v_i , 经过 v_i 的数据应满足约束:

$$\sum_{j \in N(i)} T_m f_{ij}^m - \sum_{k \in N(i)} T_m f_{ki}^m = r_i T_m. \quad (2)$$

其中, f_{ij}^m 是 v_m 轮为簇头时, v_i 到 v_j 的数据传输率; T_m 为 v_m 轮为簇头的时间. 方程(2)意味着在 v_m 轮为簇头的时间里, v_i 发送的总数据量等于进入 v_i 的数据量与 v_i 产生的数据量之和.

(2) 能量约束

对于簇内任一非簇头节点 v_i , 其能量约束满足:

$$E_{out}^i + E_{in}^i \leq e_i \quad (3)$$

$$E_{in}^i = e_{ir} \left[\sum_{1 \leq m \leq M} \sum_{k \in N(i)} T_m f_{ki}^m \right] \quad (4)$$

$$E_{out}^i = \sum_{1 \leq m \leq M} \sum_{j \in N(i)} T_m f_{ij}^m \text{cost}_{ij} \quad (5)$$

其中, E_{in}^i 和 E_{out}^i 分别表示在整个簇生存周期内, v_i 接收数据和发送数据所消耗的能量; e_{ir} 表示节点接收单位数据的能耗. 方程(3)~(5)表明: 在整个簇的生存周期内, 任何节点收发数据消耗的总能量, 不超过其初始携带的能量.

(3) 链路容量约束与非负约束

当 v_i 向其簇内邻居节点 v_j 传输数据时, v_i 到 v_j 之间的数据传输率不能超过链路的最大容量. 即

$$f_{ij}^m \leq B \quad (6)$$

同时, 簇内各节点轮为簇头的时间, 以及 v_i 到 v_j 的数据传输率还应该满足非负约束, 即

$$T_i \geq 0, f_{ij}^m \geq 0, 1 \leq i \leq M, 1 \leq m \leq M, j \in N(i)$$

综合以上各约束条件, 可以得到基于线性规划方程的簇头选举策略 LPCHS, 具体描述如下:

MAX T

S.T. $T = \sum_{1 \leq i \leq M} T_i.$

$$\sum_{j \in N(i)} T_m f_{ij}^m - \sum_{k \in N(i)} T_m f_{ki}^m = r_i T_m, 1 \leq i \leq M, 1 \leq m \leq M, m \neq i$$

$$E_{out}^i + E_{in}^i \leq e_i, 1 \leq i \leq M$$

$$E_{in}^i = e_{ir} \left[\sum_{1 \leq m \leq M} \sum_{k \in N(i)} T_m f_{ki}^m \right], 1 \leq i \leq M$$

$$E_{out}^i = \sum_{1 \leq m \leq M} \sum_{j \in N(i)} T_m f_{ij}^m \text{cost}_{ij}, 1 \leq i \leq M$$

$$f_{ij}^m \leq B, 1 \leq i \leq M, 1 \leq m \leq M, m \neq i, j \in N(i)$$

$$\text{cost}_{ij} = \alpha + \beta \cdot d_{ij}^n, 1 \leq i \leq M, j \in N(i)$$

$T_i \geq 0, f_{ij}^m \geq 0, 1 \leq i \leq M, 1 \leq m \leq M, m \neq i, j \in N(i)$ 其中, T 表示簇生存周期, M 表示簇内节点个数. 求解上述线性规划方程, 可以得到簇内各节点轮为簇头的时间以及簇的生存周期.

根据簇内各节点轮为簇头的时间, 引入簇头时间比来描述其簇头时间占簇的生存周期的比例.

定义 1 簇头时间比. 节点 v_i 的簇头时间比等于 v_i 轮为簇头的时间除以该簇的生存周期. 即

$$\text{ratio}_i = \frac{T_i}{\sum_{k=1}^M T_k}$$

4 簇头调度算法

LPCHS 假设所有的能量均用于簇内数据收集, 是因为簇间数据转发过程随着时间的推移而连续不断地进行, 因此得到的线性规划方程是一个非多项式线性规划方程, 由于任何一个非多项式线性规划问题都属于 NP-难问题^[14], 若考虑簇间能量消耗, 问题将变得相当复杂, 很难求解.

本文在 4.1 节首先研究簇间数据的转发模式, 提出一种簇间多路径路由技术. 然后 4.2 节根据 LPCHS 得到的结果, 考虑簇间数据转发所消耗的能量, 提出基于簇头时间比的簇头调度算法 CHSA.

4.1 多路径路由技术

一个簇在转发数据时, 数据在簇内的传输过程可以看作簇内的路由过程. 因此, 一个簇转发数据消耗的能量, 等于簇内路由上各节点所消耗的能量之和. 为了更均衡地利用簇的能量, 各簇采用簇内多路径技术, 即一个簇在转发来自上游簇的数据时, 并非采用固定路径, 而是可以利用多个路径完成转发任务. 例如在图 2 中, C_u, C_i, C_d 是三个簇, C_u 是 C_i 的上游簇, C_d 是其下游簇. C_i 转发来自 C_u 的数据时, 采用多路径技术, 即来自 C_u 的数据, 可经过 m 条路径 $S_{11} - S_{12} - \dots - S_{1k}, \dots, S_{m1} - S_{m2} - \dots - S_{mk}$ 中的任何一条转发至 C_d , 而非采用一条固定的传输路径.

4.2 簇头调度策略

对于一个簇 C_i 来说, 其向下游簇发送 D_i 单位的数据时簇 C_i 消耗的能量为

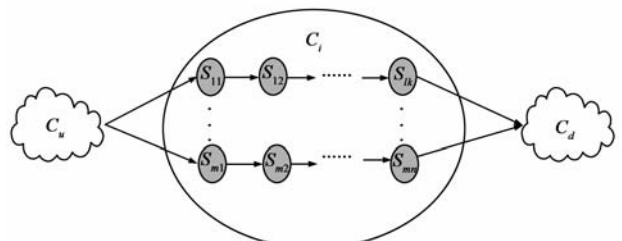


图 2 簇内多路径技术

$$E_{send} = \text{cost}_{send} \cdot D_s \quad (7)$$

其中 cost_{send} 表示簇 C_i 向其下游簇发送单位数据消耗的能量. 同理簇 C_i 从其上游簇接收 D_r 单位的数据消耗的能量为

$$E_{receive} = \text{cost}_{receive} \cdot D_r \quad (8)$$

其中 $\text{cost}_{receive}$ 表示簇 C_i 接收单位数据消耗的能量. 由此可见, 簇 C_i 的总能量消耗可以分为簇内数据收集消耗的能量 E_{intra} 和簇间数据转发消耗的能量, 即

$$E_{total} = E_{intra} + E_{send} + E_{receive} \quad (9)$$

LPCHS 描述的是簇内数据收集的过程, 而簇生存周期的长短和簇内节点携带能量的多少有着重要的关系. 换言之, 若减少簇内每个节点的能量, 则所得到的簇生存周期也会相应地减少. 由于每个节点都需要消耗部分能量完成簇间数据转发, 那么由 LPCHS 得到的每个节点轮为簇头的时间, 应根据完成簇间数据转发消耗能量的多少而按比例地减少.

对于簇 C_i , 假设其消耗 E_{total} 的能量可以得到该簇的生存周期为 T_{total} . 考虑簇间数据转发所消耗的能量 $E_{send} + E_{receive}$, 那么, 由 $E_{send} + E_{receive}$ 造成的簇的生存周期减少了:

$$T_{inter} = \frac{T_{total} \cdot (E_{send} + E_{receive})}{E_{total}} \quad (10)$$

更新簇的生存周期为:

$$T_{total} = T_{total} - T_{inter} \quad (11)$$

更新节点 v_i 轮为簇头的时间及其簇头时间比:

$$T'_i = T_i - T_{inter} \cdot \text{ratio}_i \quad (12)$$

$$\text{ratio}'_i = \frac{T'_i}{\sum_{k=1}^M T'_k} = \frac{T'_i}{T_{intra}} \quad (13)$$

其中, T'_i 为更新后 v_i 轮为簇头的时间. 方程(12)表示将簇由于完成簇间数据转发所引起的簇生存周期的减少, 按比例地平摊于簇内各节点.

基于以上思想, 本文提出一种基于簇头时间比的簇头调度算法 (CHSA). 算法的基本思想如下: 各簇根据 LPCHS 得到的结果, 确定簇头节点集 $CH = \{v_i | T_i > 0\}$. 在每个周期 (假设周期长度为 Δt) 开始时, 随机地从 CH 中选择一个节点作为簇头节点, 簇头节点在簇内广播成为簇头的消息. 当该周期结束时, 簇头节点统计簇间的能量消耗, 并根据方程 (10) ~ (13) 更新簇生存周期、簇内各节点轮为簇头的时间以及该簇的簇头节点集 CH , 广播更新后的簇内各节点的簇头时间, 并进入下一个周期.

CHSA 算法的描述如算法 1 所示.

- (1) For 每个簇 $C_i \in C$
- (2) 根据 LPCHS 得到簇生存周期, 簇内各节点簇头时间以及簇头节点集 $CH = \{v_i | T_i > 0\}$;
- (3) 设置周期 Δt ;
- (4) 随机从 CH 内选择一节点 v_i 为簇头;
- (5) while ($\Delta t \neq 0$ 且 $T_i \neq 0$)
- (6) 进行簇内数据收集及预处理工作;
- (7) 进行簇间数据转发工作;
- (8) $\Delta t - -, T_i - -$;
- (9) end while;
- (10) 计算簇间转发数据消耗的能量

$$E_{receive} = \text{cost}_{receive} \cdot D_r;$$

$$E_{send} = \text{cost}_{send} \cdot D_s;$$
- (11) 根据方程(10) ~ (13) 计算考虑簇间数据转发消耗的能量, 更新簇的生存周期、簇内各节点的簇头时间及簇头时间比;
- (12) 更新 C_i 的簇头节点集 $CH = \{v_i | T_i > 0\}$;
- (13) 若网络生存周期尚未结束, 转到第 4 步;
- (14) End For

下面以图 3 为例说明 CHSA 算法的工作流程.

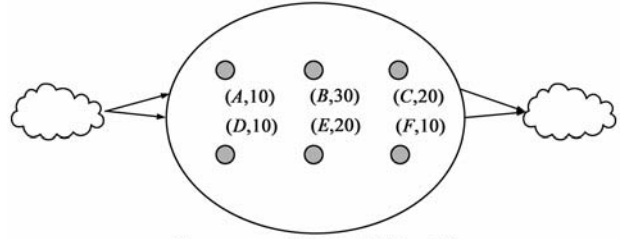


图3 CHSA簇头调度算法示例

在图 3 中, 簇 C_i 内共有 A-F 六个节点, 其中 (A, 10) 代表节点 A 的簇头时间为 10, 由上一周期更新得到 (初始状态时, 通过 LPCHS 求得). 显然, 节点 A 的簇头时间比等于 0.1. 假设 $E_{intra} = 9700$, $\Delta t = 2$, 在当前周期内, 簇头节点为 E. 当前周期结束时, 节点 E 统计其簇间能量消耗 $E_{receive} = 100$, $E_{send} = 200$. 然后计算由簇间数据转发耗能造成的簇生存周期减少

$$T_{inter} = \frac{T_{total} \cdot (E_{send} + E_{receive})}{E_{total}} = \frac{100 \cdot (100 + 200)}{100 + 200 + 9700} = 3$$

因此, 当前簇的实际生存周期应为:

$$T_{total} = T_{total} - T_{inter} = 100 - 3 = 97.$$

更新每个节点的簇头时间为: (A, 9.7), (B, 29.1), (C, 19.4), (D, 9.7), (E, 19.4), (F, 9.7), 进入下一周期.

上述例子可以看出, 节点 A 在经过一个周期后, 其簇头时间比仍为 0.1, 即在网络工作过程中, 每个节点的簇头时间比保持不变, 始终与 LPCHS 所得的簇头时间比相同. 定理 1 进一步说明上述结论.

定理 1 考虑簇间转发所消耗的能量后, 得到的更新后的簇头时间比与原来相同. 即对任一节点 v_i 有 $\text{ratio}_i = \text{ratio}'_i$.

$$\begin{aligned}
 \text{证明: } ratio'_i &= \frac{T'_i}{\sum_{k=1}^M T'_k} = \frac{T'_i}{T_{total} - T_{inter}} = \frac{T_i - T_{inter} \cdot ratio_i}{T_{total} - T_{inter}} \\
 &= T_i - \frac{T_{total} \cdot (E_{send} + E_{receive})}{E_{total}} \cdot ratio_i \\
 &= \frac{T_{total} - \frac{T_{total} \cdot (E_{send} + E_{receive})}{E_{total}}}{T_{total} - \frac{T_{total} \cdot (E_{send} + E_{receive})}{E_{total}}} \\
 &= \frac{T_i - \frac{T_i \cdot (E_{send} + E_{receive})}{E_{total}}}{T_{total} - \frac{T_{total} \cdot (E_{send} + E_{receive})}{E_{total}}} = \frac{T_i}{T_{total}} = ratio_i
 \end{aligned}$$

定理 1 表明:在长程的意义下,本文提出的簇头调度算法 CHSA 得到的调度策略,与第 3 节中的 LPCHS 得到的簇头选举时间,具有相同的效果。

5 实验与结果分析

假设网络中节点均匀分布,簇内节点个数均为 M . 簇头节点处数据融合率为 g (即簇头节点收到的数据量除以发往 sink 的数据量). 实验中涉及的参数有 $M = 10, \alpha = \beta = 1$. 为了实验的方便性,假设所有节点的数据产生率、初始能量及通信半径均相同, $r_i = 0.1 \text{ bit/s}$, 初始能量 $e_i = 2000 \text{ J}$, 通信半径 $R = \sqrt{2}$.

为了测试本文提出算法(简称 LPCHS)的性能,将 LPCHS 与以下算法进行比较。

(1) 基于能量的簇头选举策略(ECHS). 簇内数据收集时,以能量作为选举簇头的标准,在每个周期开始时,选择剩余能量最多的节点作为簇头节点. 该策略反映一类基于能量的簇头选举算法^[2~4],虽然定义的公尺不同,但都是基于能量的簇头选举。

(2) 基于平均距离的簇头选举策略(DCHS). 簇内数据收集时,以平均距离作为选举簇头的标准,选择和其它节点平均距离最小的节点作为簇头节点,在网络运行期间簇头节点不发生改变. 该策略可以代表一类基于距离的簇头选举算法^[5~8],在这些算法中,将距离作为簇头选举的衡量标准。

(3) LEACH^[9].

实验 1~4 测试在 LPCHS、ECHS、DCHS 和 LEACHS 四种方案下网络性能的变化情况。

实验 1:随着网络规模的增大,上述四种方案得到的网络生存周期如图 4(a) 所示呈递减趋势. 图中 LPCHS 得到的网络生存周期始终比 ECHS 得到的网络生存周期大. 这是因为 ECHS 所选的簇头节点距离其他节点可能较远,这样网络中能量消耗较快. 又因为 LEACHS 除了初始阶段,其他阶段也基于能量选择簇头,所以其结果和 ECHS 大体相同。

实验 2:随着数据融合率的增大,以上四种方案得到的网络生存周期变化现象如图 4(b) 所示呈递减趋

势. LPCHS、LEACHS 和 ECHS 所得到的网络生存周期几乎比 DCHS 长了一倍. 这是因为 DCHS 每个周期选择的簇头都是同一个,所以簇头节点能量消耗较快,进而网络生存周期也就比较短。

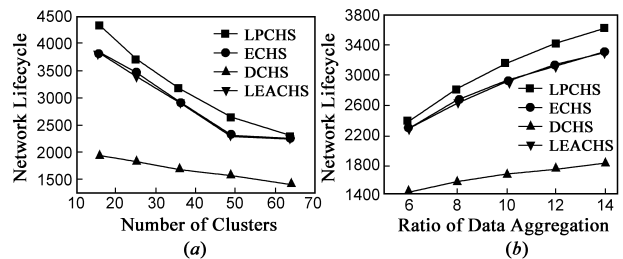


图 4 网络生存周期的变化

实验 3:随着网络规模的增大,四种方案得到的吞吐量是递增的,如图 5(a) 所示. 吞吐量的递增是因为随着簇个数的增加,网络中产生的数据包数随之增加,吞吐量也会随之增长。

实验 4:随着数据融合率的增大,网络按四种方案工作所得到的网络吞吐量如图 5(b) 所示. 从图中可以看出四种方案所得到的吞吐量都是递减的. 这是因为簇头节点融合的数据包数越多,发往 sink 节点的数据包数越少,进而吞吐量越少。

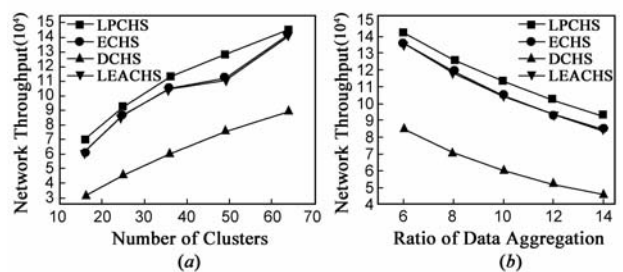


图 5 网络吞吐量的变化

6 结论

为了最大化分簇网络中簇的生存周期,本文提出了一种基于线性规划的簇头选举策略(LPCHS)和基于簇头时间比的簇头调度算法(CHSA). LPCHS 和 CHSA 较好的解决了网络中簇头节点耗能过快的问题,且得到较好的网络性能。

参考文献:

- [1] 林恺,赵海,尹震宇,罗玘玘. 一种基于能量预测的无线传感器网络分簇算法[J]. 电子学报, 2008, 36(4): 824-828.
Lin Kai, Zhao Hai, Yin Zhen-Yu, Luo Ding-ding. A clustering hierarchy arithmetic based on energy prediction for wireless sensor networks[J]. Acta Electronica Sinica, 2008, 36(4): 824-828. (in Chinese)
- [2] Ghiasi S, Srivastava A, Yang X, Sarrafzadeh M. Optimal energy aware clustering in sensor networks[A]. Sensors[C]. Florida:

- Academic Press, 2002. 258 – 269.
- [3] Ming Yu, Kin K Leung, Aniket Malvankar. A dynamic clustering and energy efficient routing technique for sensor networks [J]. IEEE Trans on Wireless Communications, 2007, 6(8): 3069 – 3079.
- [4] Min Qin, R Zimmermann. An energy-efficient voting-based clustering algorithm for sensor networks [A]. SNPD/SAWN [C]. Maryland: Academic Press, 2005. 444 – 451.
- [5] Yaacov Fernandess, Dahlia Malkhi. K-clustering in wireless ad hoc networks [A]. The Second ACM International Workshop on Principles of Mobile Computing [C]. Toulouse: ACM, 2002. 31 – 37.
- [6] A Manjeshwar, D P Agrawal. TEEN: A protocol for enhanced efficiency in wireless sensor networks [A]. The 1st International Workshop on IPDPS [C]. CA: IEEE Computer Society, 2001. 2009 – 2015.
- [7] Arati Manjeshwar, Dharma P Agrawal. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks [A]. IPDPS [C]. FL: IEEE Computer Society Press, 2002. 195 – 202.
- [8] Quanbin Chen, Jian Ma, Yanmin Zhu, et al. An energy-efficient K-Hop clustering framework for wireless sensor networks [A]. EWSN [C]. Delft: SPRINGER-VERLAG, 2007. 17 – 33.
- [9] W B Heinzelman, A P Chandrakasan, H Balakrishnan. An application specific protocol architecture for wireless microsensor networks [J]. IEEE Trans on Wireless Communication, 2002, 1(4): 660 – 670.
- [10] Ossama Younis, Sonia Fahmy. HEED: A hybrid, energy-efficient, distributed clustering approach for Ad Hoc sensor networks [J]. IEEE Trans Mobile Computing, 2004, 3(4): 366 – 379.
- [11] T Kang, et al. A clustering method for energy efficient routing in wireless sensor networks [A], WSEAS [C]. Greece: WSEAS, 2007. 133 – 138.
- [12] Takehiro Furuta, Mihiro Sasaki. Fumio Ishizaki, Atsuo Suzuki. A New Clustering Algorithm Using Facility Location Theory for Wireless Sensor Networks [R]. Technical Report NAN-ZAN-TR-2006-04, Nanzan Academic Society, 2007.
- [13] Yi Shi, Y Thomas Hou. Theoretical results on base station movement problem for sensor network [A]. INFOCOM [C]. Phoenix, AZ: IEEE, 2008. 376 – 384.
- [14] M R Garey, D S Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness [M]. W H Freeman and Company, New York, NY, 1979.

作者简介:



柳 絮 女, 1987 年生于黑龙江林口. 现为黑龙江大学计算机科学技术学院硕士生, 主要研究方向为无线传感器网络和数据库.

E-mail: lxliuxu@sina.com



李金宝 男, 1969 年生于黑龙江庆安. 博士, 黑龙江大学计算机科学技术学院教授, 主要研究方向为无线传感器网络、数据库、移动计算和并行计算.

E-mail: jbli@hlju.edu.cn