

传感器网络中基于多级选举的非均匀簇头选择算法

王雷春 马传香

(湖北大学数学与计算机科学学院 武汉 430062)

摘要 为了节省和平衡网络中簇头的能耗,提出了一种基于多级选举的非均匀簇头选择算法 MSEBNECHSA。该算法首先通过空间虚拟网格、簇头和 Sink 构成的三级选举机制构造网络候选簇头的能量分布图,然后采用基于候选簇头能量和距离的模拟退火算法,使网络中簇头的竞争范围随着到 Sink 的距离远近而相应从小到大非均匀分布。仿真实验表明,该算法能够以较低的能量成本将网络中的节点有效分簇,并且平衡网络中的能量消耗。

关键词 传感器网络,多级选举,非均匀,簇头选择

中图分类号 TP391 文献标识码 A

Multi-stage Election Based Non-even Cluster Heads Selection Algorithm for Wireless Sensor Networks

WANG Lei-chun MA Chuan-xiang

(College of Mathematics & Computer Science, Hubei University, Wuhan 430062, China)

Abstract In order to save and balance energy consumption of cluster heads (simply CHs), a multi-stage election based non-even cluster heads selection algorithm for wireless sensor networks, MSEBNECHSA, was proposed. Firstly, MSEBNECHSA built energy distribution graph of candidate CHs by three-stage election mechanism including spatial virtual grids, CHs and Sink. By adopting simulation annealing algorithm based on the parameters including energy of candidate CHs and distance among them, MSEBNECHSA then made CHs' competition range vary with the distance from CHs to Sink. Simulation results show that MSEBNECHSA can cluster CHs by less energy and balance energy consumption of CHs in WSN.

Keywords Wireless sensor networks (simply WSN), Multi-stage election, Non-even, Cluster heads selection

1 引言

节点分簇是传感器网络拓扑控制的重要内容之一,簇头选择则是节点分簇的基础和核心。通过选择良好的簇头,不但可以有效管理网络中的节点,还可以帮助完成路由选择、资源管理以及数据融合等功能,从而减少节点维护拓扑和广播数据的能耗,显著提高网络的使用寿命^[1]。

目前,已经有许多文献对传感器网络中的簇头选择算法进行了研究。Heinzelman W 等^[1,2]提出了 LEACH 及基于 LEACH 的多个分簇算法。HEED^[3]簇头选择算法通过把能量消耗分布在整个网络中来延长网络生命。与 LEACH 等分簇算法相比,HEED 分簇速度更快,能产生更加分布均匀的簇头、更合理的网络拓扑。EECS^[4]给出了一种计算通信成本的新方法。通过该方法,节点可以决定是作为一个成员节点还是作为一个类首。实验表明,EECS 的通信成本远远小于 HEED,且网络生命被大大地延长。考虑传统分簇算法的部署受限或可靠性缺乏等弊端,文献^[5]提出了一种启发式的分簇控制算法。该算法以较合理的簇规模进行分簇划分,所获拓扑结构具有全局能耗低、骨干网健壮性高的特点,能有效地延长 WSN 的生命期。针对无线传感器网络节点能量受限的

特点,文献^[6]提出了一种响应式分布分簇算法(RDCA)。该算法不需预先得知节点自身及其他节点的位置信息,而仅根据局部拓扑信息快速进行分布式的簇头选举,并根据代价函数进行簇的划分。针对分簇算法中的重新分簇所带来的高负载问题,李建波等^[7]提出了一种基于完全图的能量有效分簇算法(CGCA)。仿真实验表明:在节点密集部署的情况下,CGCA 产生的消息交换个数远小于 HEED 分簇算法;在簇头均匀分布方面,CGCA 也明显优于 LEACH 分簇算法。然而,上述簇头选择算法选择簇头时主要考虑的是作为簇头节点的剩余能量要高,网络中的能量消耗尽量均匀,以延长网络生命;并没有考虑到簇头分布对传感器网络中数据传输的重要意义。在 WSN 中, Sink 的数目非常有限(一个或几个),网络中的簇头与 Sink 之间的距离并不相同。与距离 Sink 较远的簇头相比,距离较近的簇头一方面要收集本簇中节点采样的数据,另一方面还需要将距离 Sink 较远的簇头收集的信息转发给 Sink,这样能量消耗将大大加快,从而导致 WSN 中后期数据收集任务无法完成。

本文提出了一种基于多级选举的非均匀簇头选择算法 MSEBNECHSA。该算法一方面通过选举机制减少收集网络中节点能量状况的通信量;另一方面利用基于候选簇头领域

到稿日期:2009-06-12 返修日期:2009-08-12 本文受国家自然科学基金(60603069)资助。

王雷春(1974—),男,博士,讲师,主要研究方向为传感器网络等,E-mail:wlc2345702@163.com;马传香(1971—),女,博士,副教授,主要研究方向为移动 Ad hoc 网络等。

竞争的模拟退火算法,使靠近 Sink 的簇头竞争范围较小,以节约簇头能量,供数据转发使用,达到节省和均衡簇头能量消耗的目的。

2 基于多级选举的非均匀簇头选择算法

2.1 簇头选择原理

本节设计的基于多级选举的非均匀簇头选择算法 MSEBNECHSA 由 Sink 完成,以候选簇头的剩余能量和它们之间的距离作为主要考虑依据。其方法是:由空间虚拟网格、簇头、Sink 构成三级结构的选举方式来收集网络中节点能量较高的节点,构成候选簇头能量分布图;然后从这些能量较高的节点中通过模拟退火算法选择合适的节点作为簇头。

2.1.1 构造候选簇头能量分布图

网络中候选簇头能量分布图的构造由空间虚拟网格、各个簇的簇头以及 Sink 共同完成。

1) 空间虚拟网格

传感器网络中节点之间的高度密集布置导致搜集到的数据高度相关和冗余^[8];另一方面,同一区域中节点的竞争信道也会带来过多的数据发送冲突。因此,本文将整个传感器网络划分成 $M \times N$ 个正方形小区域(M 和 N 的取值由簇的大小和密度等决定),每个小区域是基本的管理分区单元,称为空间虚拟网格。

空间虚拟网格的每个节点维护着一个包括该虚拟网格中所有存活节点能量状况的链表,称为能量状态表 $List\{E_i\}$ 。能量状态表将虚拟网格的节点按照其能量状态进行排序,能量高的在表头,能量低的在表尾(见图 1)。在每一轮,选择能量最高的节点作为工作节点(ActiveNode, AN),其它节点休眠(SleepingNode, SN)。在一轮的末尾,AN 将自己的能量状况写入能量状态表,并将能量状态表广播给空间虚拟网格的每个节点。收到能量状态表的节点根据自己在能量状态表中的位置决定是工作还是休眠。同时,工作节点从该表中选择能量状态最高的、在下一轮工作的节点即该空间虚拟网格选举的候选簇头 CH_{Candi} 的相应信息(包括该节点所在空间虚拟网格的编号 $ID_{CH_{Candi}}$,空间位置 $LOC_{CH_{Candi}}$,能量状况 $E_{CH_{Candi}}$ 和局部空间相关指标 $C_{CH_{Candi}}$)发送给簇头。

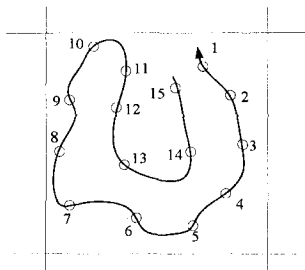


图 1 空间虚拟网格中的节点调度

2) 簇头

各个簇候选簇头能量状况的收集任务由每个簇的簇头在本轮工作的末尾完成。其方法是:簇头首先接收各个空间虚拟网格发送的候选簇头相关信息 $M_{CH_{Candi}}$ ($ID_{CH_{Candi}}$, $LOC_{CH_{Candi}}$, $E_{CH_{Candi}}$, $C_{CH_{Candi}}$);然后对收集的各个簇头按照其能量状况排序;最后,选择若干能量较高候选簇头的相关信息 $M_{CH_{Candi}}$ 发送给 Sink。

3) Sink

Sink 负责在一轮的末尾接收各个簇头发送的候选簇头相关信息 $M_{CH_{Candi}}$,并构造候选簇头的能量分布图。候选簇头的能量分布图中各个候选簇头信息中最主要的两个内容是各个候选簇头的空间位置 $LOC_{CH_{Candi}}$ 和各个候选簇头的能量状况 $E_{CH_{Candi}}$ 。前者是计算各个候选簇头之间距离的依据,后者则是选择簇头时考虑其能量的依据。

2.1.2 簇头选择

簇头选择由 Sink 通过模拟退火算法完成。

1) 簇头竞争范围

由于靠近 Sink 的簇头除了要将本簇节点的数据收集、处理后发送给 Sink,还要转发来自距离 Sink 较远簇头收集的数据,因此距离 Sink 较近的簇头竞选半径应该较小,成员数目较少,以预留部分能量供簇头间通信使用。

记候选簇头 CH_{Candi} 的最大竞争半径 $R_{Candi_max} = R_m$,最小竞争半径 $R_{Candi_min} = (1-\alpha)R_m$,其中 α 是控制候选簇头竞争范围的参数,在 0~1 之间取值。

定理 1 本文规定候选簇头 CH_{Candi} 的竞争范围与该候选簇头到 Sink 之间的距离呈线性递增关系,则候选簇头 CH_{Candi} 的竞争半径 R_{Candi} 可计算如下:

$$R_{Candi} = \left[1 - k\alpha \frac{d_{CH_{Sink_max}} - d(CH_{Candi}, Sink)}{d_{CH_{Sink_max}} - d_{CH_{Sink_min}}} \right] R_m \quad (1)$$

其中, $d_{CH_{Sink_max}}$ 和 $d_{CH_{Sink_min}}$ 分别是候选簇头 CH_{Candi} 到 Sink 距离的最大值和最小值; $d(CH_{Candi}, Sink)$ 为候选簇头 CH_{Candi} 到 Sink 的距离; k 是簇头竞争半径调节系数,其取值范围是 0~1 之间的一个常数,由用户确定。

2) 基于候选簇头能量和竞争范围的模拟退火算法

模拟退火算法是局部搜索算法的扩展,其关键是要选定合适的目标函数和初始参数。簇头选择一般要考虑两个因素:一是能量因素,即被选作簇头的节点能量应该尽量高;二是簇头之间的距离,要使簇头的竞争范围适当。由于采用的是非均匀分簇算法,因此这里选择的是两个簇头之间的综合相对参数指标作为目标函数的参数。

定义 1(邻簇头) 与候选簇头在一定距离范围内的簇头定义为该簇头的邻簇头。

在簇头选择算法中,候选簇头 CH_{candi} 的邻簇头集合 S_{candi} 为:

$$S_{candi} = \{CH_j | d(CH_{candi}, CH_j) \leq \text{Min}(d^0, 2R_m)\} \quad (2)$$

其中, d^0 是节点的最大通信距离, R_m 是候选簇头的最大竞争半径。

假定候选簇头 CH_i 和 CH_j 相互之间是邻簇头,能量分别为 e_{candi_i} 和 e_{candi_j} ,则它们之间的能量参数可通过式(3)计算:

$$e_{candi_ij} = \frac{\text{Min}(e_{candi_i}, e_{candi_j})}{\text{Max}(e_{candi_k})} \quad (3)$$

其中, $\text{Min}(e_{candi_i}, e_{candi_j})$ 是簇头 CH_i 和它的邻簇头 CH_j 之间的较小能量, $\text{Max}(e_{candi_k})$ 是所有簇头的最大能量。

假定邻簇头 CH_i 和 CH_j 的竞争半径分别为 R_{candi_i} 和 R_{candi_j} , d_{candi_ij} 是两个簇头之间的实际距离,则它们之间的距离参数可通过式(4)计算:

$$d_{candi_ij} = \frac{d_{candi_ij}}{R_{candi_i} + R_{candi_j}} \quad (4)$$

两个邻簇头 CH_i 和 CH_j 之间的综合参数 w_{candi} 可计算如

下:

$$w_{candi} = w_{candi_e} * e'_{candi_{ij}} + w_{candi_d} * d'_{candi_{ij}} \quad (5)$$

其中, w_{candi_e} 和 w_{candi_d} 分别是能量参数和距离参数所占的权重。

方差是衡量参数变异程度的指标, 这里采用邻簇头之间的综合参数方差作为使用模拟退火算法求最佳簇头的目标函数。其计算公式为:

$$D_{candi} = \sum (\omega_{candi_{ij}} - \overline{\omega_{candi}})^2 \frac{1}{N_{Neigh_CHs}} \quad (6)$$

其中, $\omega_{candi_{ij}}$ 是第 i 个候选簇头的第 j 个邻簇头的综合参数, $\overline{\omega_{candi}}$ 是所有簇头的所有邻簇头综合参数的平均值, N_{Neigh_CHs} 是所有簇头的所有邻簇头数目。

2.2 算法描述

簇头选择实现算法描述如下。

算法输入: 选举产生的候选簇头及其相关信息。

算法输出: 满足要求的簇头。

算法步骤:

- Step 1 //工作节点 AN 推选空间虚拟网格 SVG 的候选簇头 CH
1. gather E_{Node_i} of SVG;
 2. sort($Node_i$) according to E_{Node_i} ;
 3. write sort($Node_i$) to List(E_{Node_i});
 4. choose CH_{Candi} with Max(List(E_{Node_i}));
 5. send M_{CH_Candi} (ID_{CH_Candi} , Loc_{CH_Candi} , E_{CH_Candi} , C_{CH_Candi}) to CH;
- Step 2 //簇头 CH 推选各个簇的簇头
1. receive M_{CH_Candi} ;
 2. sort(CH_{Candi}) according to E_{CH_Candi} ;
 3. choose N_{CH_Candi} CHs_{Candi} with Max(E_{CH_Candi});
 4. send N_{CH_Candi} M_{CH_Candi} to Sink;
- Step 3 //Sink 选择簇头
1. receive M_{CH_Candi} ;
 - //使用模拟退火算法选择簇头。
 2. choose $N_{CH}CHs_{Candi}$ as CHs; //选择部分候选簇头作为初始值。
 3. calculate paras according to Equ (3), (4) and (5); //计算能量、距离及综合参数。
 4. calculate D_{candi} according to Equ (6); //计算方差。
 5. swap small proportion of CHs and CHs_{Candi} ; //交换部分候选簇头和簇头。
 6. calculate D'_{candi} according to Equ (6);
 7. calculate $\Delta D_{candi} = D'_{candi} - D_{candi}$;
 8. if ($\Delta D_{candi} < 0$) accept D'_{candi} ;
 - else accept D'_{candi} with p ; //以一定的概率接受 D'_{candi} 。
 9. repeat 2-8 until $D_{candi} \leq D_{candi_{th}}$. //当 D_{candi} 小于规定的阈值 $D_{candi_{th}}$, 算法结束。

3 实验结果及分析

通过 NS-2 实现了算法 MSEBNECHSA。实验统计传感器节点接收数据、融合数据和发送数据所消耗的能量, 计算了网络的存活时间(用轮数表示), 分析了算法的能量效率。实验中所用的参数如表 1 所列, 其中能量消耗模型相关的参数取自文献[2]。其它协议中的参数通过运行多次实验找出其最优取值。为了评价算法的性能, 比较了 MSEBNECHSA 与 LEACH-C, HEED 和 EECS 在簇头质量、簇头能耗方差、分簇能耗以及能量使用效率等方面的指标, 实验结果见图 2-图 5。

表 1 实验参数

参数	取值
网络覆盖区域	(0,0)~(300,300)m ²

Sink 位置	(400m,150m)
节点数目	900
簇头数 N_{CH}	50
数据包大小	525Byte
节点初始能量大小	1J
簇头最大竞争半径	45m
簇头竞争范围控制系数 α	1/3
簇头竞争半径调节系数 k	0.9
E_{elec}	50nJ/bit
ϵ_{fs}	10pJ/bit/m ²
ϵ_{amp}	0.0013pJ/bit/m ⁴
EDA	50nJ/bit/signal

从图 2 和图 3 可知, MSEBNECHSA 在选择簇头时消耗的能量最少, 但簇头的质量比较高。其原因是, 一方面, MSEBNECHSA 通过网络中的多级选举, 在较少通信开销的情况下获得了能量的基本信息; 另一方面, 网络在推选候选簇头时选择的是各个虚拟网格和簇能量最高的节点, 增加了簇头选择质量。LEACH-C 通过集中式方式选择的簇头能量很高, 但能耗也最大。HEED 和 EECS 的簇头选择基于局部竞争, 在竞争时需要消耗较大的能量, 因此簇头的能量波动较大。EECS 由于考虑了节点成簇的成本, 因此性能略好于 HEED。

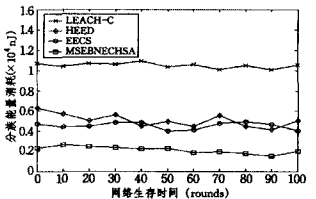
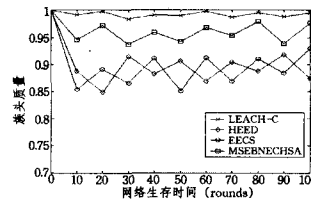


图 2 不同分簇算法的簇头质量

图 3 不同算法的分簇能耗

图 4 显示了随机选取的 10 轮簇头消耗的能量方差。从图中可以看出, MSEBNECHSA 的能耗方差最低而且稳定, 这说明算法最好地均衡了簇头的能量消耗。LEACH-C 和 HEED 的方差比较接近, 而且它们都有明显的波动, 这说明它们都没有采取有效的策略均衡簇头的能量消耗。EECS 由于考虑了负载均衡和距离 Sink 较远的簇头能耗过快等因素, 因此簇头能耗方差波动较小。

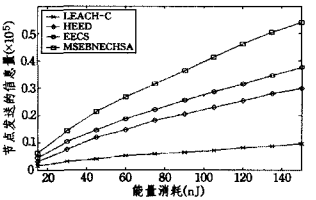
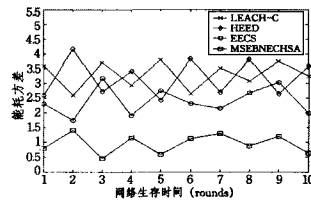


图 4 不同分簇算法的簇头能耗方差

图 5 不同分簇算法的能量利用效率

从图 5 中可以看出, 算法 MSEBNECHSA 的能量利用效率最高, 其原因是节点通过簇头多级选举节省了能量, 同时在选择簇头时考虑了节点与簇头间的距离以及节点和簇头之间的空间相关性, 减少了传输的数据量和数据传输时的能耗, 提高了能量的使用效率。LEACH-C 由于在每次选择簇头时需要收集整个网络中节点的能量, 因此能耗较大, 能量使用效率最低。HEED 通过考虑簇内通信代价和减小控制报文开销, 提高了能量效率。EECS 进一步考虑了节点成簇代价和控制报文的开销, 能量利用效率比 HEED 有所提高。

结束语 本文提出了一种基于多级选举的非均匀簇头选择算法 MSEBNECHSA。与 LEACH-C, HEED 和 EECS 等

分簇算法相比, MSEBNECHSA 能够以较低的能量成本将网络中的节点有效分簇, 并且平衡网络中的能量消耗。仿真结果证明了该算法的这些特点。

参考文献

[1] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless micro-sensor networks [C]//Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000; 3005-3014

[2] Heinzelman W, et al. An application-specific protocol architecture for wireless microsensor networks[J]. IEEE Trans on Wireless Communications, 2002, 1(4): 660-670

[3] Younis O, Fahmy S. Heed: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks[J]. IEEE

Trans. on Mobile Computing, 2004, 3(4): 660-669

[4] Ye M, et al. EECS: An energy efficient clustering scheme in wireless sensor networks[C]//Proc. of the IEEE Int'l Performance Computing and Communications Conf. New York; IEEE Press, 2005; 535-540

[5] 刘林峰, 刘业. 一种无线传感器网络拓扑的启发式分簇控制算法[J]. 计算机研究与发展, 2008, 45(7): 1099-1105

[6] 胡静, 沈连丰, 宋铁成, 等. 新的无线传感器网络分簇算法[J]. 通信学报, 2008, 29(7): 20-26

[7] 李建波, 黄刘生, 徐宏力, 等. 一种密集部署传感器网络的分簇算法[J]. 计算机研究与发展, 2008, 45(7): 1106-1114

[8] Jindal A, Psounis K. Modeling Spatially Correlated Data in Sensor Networks [J]. ACM Transactions on Sensor Networks, 2006, 2(4): 466-499

(上接第 81 页)

从图 3 中可以看出, 采用 HRRF 算法, 系统总的吞吐量提高了 10% 左右, 因为对于软实时任务采用高响应比优先调度策略, 单位时间内提高了系统的吞吐量。

图 4 显示, 随着时间推移, 系统总的能量消耗大约提高了 3% 左右, 虽然任务调度要消耗系统能源, 但是调度耗能只占系统一小部分, 传感器网络能耗主要消耗在通信模块中, HRRF 调度策略所带来的额外开销是很小的, 适合应用在能源受限的无线传感器网络中。

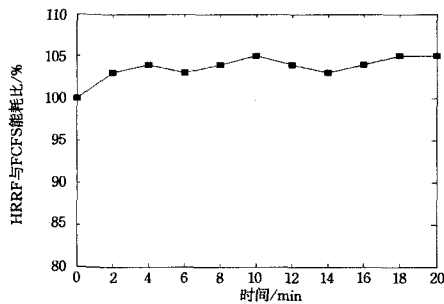


图 4 HRRF 和 FCFS 能耗百分比

图 5 显示了 10 个传感器节点在 1000s 内运行 Sense 时成功执行任务的比率。可以看出, HRRF 调度策略在面临强实时性任务时, 任务成功执行的比率远高于 TinyOS 原有的 FCFS 策略。

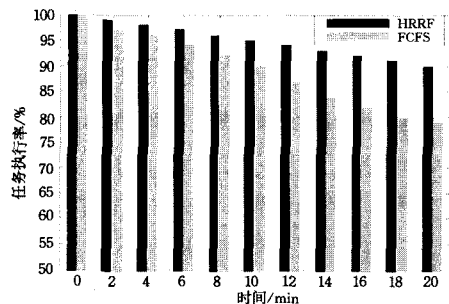


图 5 HRRF 和 FCFS 任务执行成功率和时间对比

结束语 TinyOS 作为一种典型的传感器网络操作系统被广泛使用, 但是其 FCFS 调度策略使得 TinyOS 在强实时性及系统吞吐量方面受到限制。本文提出的 HRRF 调度策略,

优先调度具有时限要求的任务, 提高了系统对紧急任务的及时响应, 同时对没有时限要求的任务综合考虑了其要运行时间和已经等待时间, 采用高响应比调度策略, 提高了系统吞吐量, 改善了通信状况, 保证了节点的高实时性和高吞吐量。仿真结果证明, 该算法适合应用在能源受限的无线传感器网络中。

参考文献

[1] Hong W, Madden S. TinySchema: creating attributes and commands in TinyOS [EB/OL]. [2005-03-11]. <http://webs.cs.berkeley.edu/tos>

[2] Rmer K, Mattern F. The design space of wireless sensor networks[J]. IEEE Wireless Communications, 2004, 11(6): 54-61

[3] Karlof C, Wagner D. Secure routing in wireless sensor networks: Attacks and countermeasures[J]. Ad Hoc Networks, 2003, 1(1): 293-315

[4] Vuran M C, Akan O B, Akyildiz I F. Spatio-temporal correlation: theory and applications for wireless sensor networks[J]. Computer Networks, 2004, 45 (3): 245-259

[5] Madden S, Franklin M J, Hellerstein J M, et al. TinyDB: an acquisitional query processing system for sensor networks [J]. ACM Transactions on Database Systems, 2005, 30(1): 122-173

[6] Intanagonwiwat C, Govindan R, Estrin D. Directed Diffusion: a scalable and robust communication paradigm for sensor networks[J]. Proceedings of the ACM MobiCom'00, 2000(1): 56-67

[7] Venkita S, Huang Huangming, Seema D, et al. Priority scheduling in TinyOS: a case study[R]. TR: WUCSE-2003. Washington University, 2003; 74

[8] Madden S, Franklin M J, Hellerstein J M, et al. TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks[M]. OSDI, Boston, USA, Dec. 2002

[9] Kuo S P, Kuo H J, Tseng Y C, et al. Detecting movement of beacon in location-tracking wireless sensor networks[A]// Proceedings of 2007 IEEE 66th Vehicular Technology Conference [C]. Baltimore, USA, 2007; 362-366

[10] Sivrikaya F, Yener B. Time synchronization in sensor networks: a survey[J]. IEEE Network, 2004, 18(4): 45-50