

WSN 中基于能量反馈的分簇路由算法 EFCA^①

史菲菲^{1,2}, 韩卫光², 李 昌^{1,2}

¹(中国科学院 研究生院, 北京 100049)

²(中国科学院 沈阳计算技术研究所, 沈阳 110171)

摘要: 分簇技术在无线传感器网络中得到广泛的应用. 提出了一种基于能量反馈的分簇路由算法(EFCA). 该算法以簇为单位构造数据聚合树, 解决了单跳传输外沿节点能耗过快和多跳传输时延过大的问题; 同时在簇内选择簇头对, 令主次节点分别负责簇间信息传递和簇内信息收集, 进一步减小了数据传输延迟; 提出的能量反馈机制很好的解决了网络“热区”问题. 经仿真验证, 可以有效的延长网络的生存时间.

关键词: 无线传感器网络; 生存时间; 数据聚合; 簇头对; 能量反馈

Low-Power Clustering Routing Algorithm EFCA Based on Energy-Feedback in WSN

SHI Fei-Fei^{1,2}, HAN Wei-Guang², LI Chang^{1,2}

¹(Graduate University, the Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, the Chinese Academy of Sciences, Shenyang 110171, China)

Abstract: Clustering technology has been widely used in wireless sensor networks. This paper introduces a clustering routing algorithm based on energy feedback (EFCA). The algorithm constructs a data aggregation tree in cluster units to reduce energy consumption of the outer edge node in multi-hop transmission and reduce the delay in single-hop transmission; selects a cluster head-couple in the cluster, the primary and secondary nodes are responsible for transmission of information between the clusters and the information gathering in the cluster, further reducing the data transmission delay; the energy feedback mechanism proposed eliminate the network “hot zone”. Simulation can effectively extend the network lifetime.

Key words: wireless sensor networks; networks lifetime; data aggregation; luster head-couple; energy feedback

1 引言

无线传感器网络的路由协议主要包括平面路由协议和层次路由协议, 而层次路由协议中的分簇路由协议由于其在网络扩展性、通信效率和能耗性能等方面具有明显的优势, 成为路由协议的主要方向^[1,2].

Heinzelman 等^[3]提出的 LEACH 协议以轮机制周期性地执行簇建立和稳定数据通信过程, 等概率随机选取簇头实施数据融合策略, 有效减少能量消耗, 延长了网络的生存时间. 但是 LEACH 协议未将节点的剩余能量纳考虑范畴, 导致部分节点很快耗尽能量而缩短网生存时间, 同时由于 LEACH 采取簇头与汇聚节点单跳通信的方式, 容易出现能耗不均的现象, 从

而进一步影响了网络的整体覆盖和生存时间.

为此, 许多学者在研究 LEACH 的基础上提出许多有针对性的改进. 刘庆^[4], 陆丽芳^[5]等提出在簇头与汇聚节点之间采取多跳的通信方式, 有利于节约簇头能量. 但是, 崔莉等^[6]认为距离汇聚节点较近的簇头需转发大量其他簇头发送的数据而消耗过多能量, 形成“热区”. Lindsey 等^[7]提出的多跳 PEGASIS 协议将网络内所有的传感器节点构成“链”, 全网只选择一个簇首(leader), 链中的节点与相邻节点通信, 把收集到的数据沿着链逐跳汇聚到 leader, 再由 leader 把数据发送给 sink. PEGASIS 避免了 LEACH 协议因频繁选取簇头、重构簇来的通信开销, 提高了网络的生存时间,

① 收稿时间:2012-02-20;收到修改稿时间:2012-04-30

但传输时延也明显增大. 文献[8,9]也根据具体的应用场合提出了各自有针对性的解决方法.

本文在研究各类路由协议的基础上, 针对层次型路由存在的问题, 提出一种基于能量反馈的分簇路由算法. 该算法以簇为单位构建最小生成树进行数据转发, 解决了多跳传输延迟过大, 单跳传输外层簇头节点能耗过快的问题. 同时对分簇算法进行改进, 引入了簇头对的概念, 令簇头对中的两个节点分别负责数据的收集处理与转发, 均衡了节点能耗, 加快数据处理转发速度, 进一步减小延迟. 并且为解决网络热区问题, 提出能量反馈机制, 平均剩余能量小于一定阈值的簇群自动发送反馈消息给下级簇群, 不再负责下级簇群的数据转发, 从而解决了网络“热区”问题, 均衡了能耗. 同时为了减少划分簇群, 重建路由的次数, 在稳定期进行簇内簇头选举, 大大节省了因大规模重建路由而消耗的能量, 延长了网络的生命期.

2 无线能量模型

无线能量模型根据发送节点和接收节点的距离 d 分为自由空间(Free Space, Fs)模型和多路径衰减(Multipath,MP)模型, 假设两节点之间的距离 d 小于 d_0 时采用自由空间模型, 大于等于 d_0 时采用多径衰减模型^[3].

d_0 由式(1)确定:

$$d_0 = \sqrt{\frac{16 \times \pi^2 \times L \times h_r^2 \times h_t^2}{\lambda^2}} \quad (1)$$

其中, L 是系统功耗因子, h_r 和 h_t 分别是接收和发送天线的高度, λ 是载波波长.

因此, 当两个距离为 d 的节点之间发送 1 比特数据时, 发送和接收所消耗的能量分别为:

$$E_{TX}(l, d) = E_{TX-elec}(l) + E_{TX-amp}(l, d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2, & d < d_0. \\ lE_{elec} + l\epsilon_{mp}d^4, & d \geq d_0 \end{cases} \quad (2)$$

$$E_{RX}(l) = E_{RX-elec}(l) = lE_{elec} \quad (3)$$

其中, E_{elec} 是信号处理所需能量, 由数字编码、调制、滤波和扩展信号等因素决定. ϵ_{fs} 、 ϵ_{mp} 分别为两种信道模型下功率放大所需能量.

3 算法

首先对网络模型作如下假设: 节点部署后将不再移动; 所有的节点都具有相同的处理和通信能力, 节点在网络中的地位平等; 节点部署后不需要人为维护; 每个节点的初始能量相同, 但由于探测收集到的数据量的大小不同, 距离簇头节点的远近不同, 每个节点的能量消耗是不一样的; 节点的无线发射功率可控, 节点可以根据算法来调整发射功率的大小; 节点的地理位置信息已知; 每个节点都能与基站通信.

3.1 节点发送一次数据的平均时间

吕涛等人指出^[10], 网络中由 n 个节点发送一个数据包的平均发送时间为

$$T_{avg}(n) = 2\tau \frac{1 - P_s}{P_s} + T + \tau \quad (4)$$

其中 τ 为端到端的延迟; T 为节点数据成功发送的时间; P_s 为 n 个节点中某一个节点发送成功的概率. 其中 $P_s = nq(1-q)^{n-1}$, q 为每个节点发送数据的概率; $T = l/s$, l 为数据包长度, s 为信道速率.

3.2 簇群的建立阶段

在每轮的开始首先进行簇头的选举. 然而群的数量对于算法性能有很大的影响, 过多或过少都会影响到算法的性能. Heinzelman 等人^[11]在一定收、发能耗参数的情况下推导出最优簇群数目 n_0 为:

$$n_0 = \sqrt{\frac{\frac{p}{1 - p(r \bmod \frac{1}{p})} \cdot M \sqrt{\epsilon_{fs}}}{2\pi \cdot \frac{E_{elec}(\frac{p}{1 - p(r \bmod \frac{1}{p})} + 1) + \epsilon_{mp}d^4}}}} \quad (5)$$

其中 p 为簇首节点占网络中节点总数的百分比, r 为当前循环进行的轮数, M 为传感器网络区域的边长.

簇头的选举以分布式的方式完成, 每个节点以一定概率决定在本轮中是否成为群首. 方法如下: 每个节点产生一个 $0 \sim 1$ 之间的随机数, 如果该数小于门限值 $T(n)$, 则该节点成为簇头. 计算 $T(n)$ 时, 将节点 S_i 在第 r 轮循环时的剩余能量 $E_i(r)$ 作为参数, 从而可以保证具有更多剩余能量的节点更多的担任簇首工作, 以延长网络的生存周期. $T(n)$ 的计算如下:

$$T(n) = \frac{n_0}{N - n_0 \times (r \bmod \frac{N}{n_0})} \times \frac{E_i(r)}{E(r)} \quad (6)$$

其中, N 为节点总数, $\overline{E_{(r)}}$ 为第 r 轮循环时网络节点的平均剩余能量.

竞选为簇头的节点向所有节点广播自己成为簇头的消息, 其余每个节点接受簇头节点发来的广播消息并根据收到的信号的强弱来选择加入信号最强的簇, 并报告给该簇头. 假设每个簇的成员数为 k , 共选举产生簇头数为 $N_{ch} = \lfloor \frac{N}{k} \rfloor$, 那么, 簇头发布消息时单簇能量损耗为

$$E_{s1} = E_{TX}(l_{msg}, d) + kN_{ch} E_{RX}(l_{msg}) + (k + 1)T_{avg}(N_{ch})E_{idle} \quad (7)$$

其中簇头为

$$E_{ch1} = E_{TX}(l_{msg}) + T_{avg}(N_{ch})E_{idle} \quad (8)$$

延时为

$$T_{s1} = T_{avg}(N_{ch}) \quad (9)$$

节点加入时单簇能量损耗为

$$E_{s1}' = kE_{TX}(l_{msg}, d) + kE_{RX}(l_{msg}) + (k + 1)T_{avg}(k)E_{idle} \quad (10)$$

其中簇头为

$$E_{ch1}' = kE_{RX}(l_{msg}) + T_{avg}(k)E_{idle} \quad (11)$$

延时为

$$T_{s1}' = T_{avg}(k) \quad (12)$$

3.3 簇头对的选择阶段

现在已经提出的分簇算法中, 簇头节点负责簇内数据的收集、处理和簇间数据传递. 但这样会造成簇头节点负担过重, 能量消耗过快. 所以引入簇头对的概念以均衡能量消耗.

每个簇头节点选择一个节点跟自己组成簇头对 G_{ij} , i 代表簇头节点编号, j 代表次节点编号. 对于次节点的选择, 簇内成员能量若大于剩余能量阈值 $E_{remains}$ (初始能量的 40%), 则向簇头发送竞争次节点消息, 消息最先到达簇头的节点当选为次节点, 然后簇头忽略剩余消息. 簇头向簇内节点广播次节点位置信息.

簇头对选择完成以后, 在稳定的数据传输阶段, 次节点将负责簇内节点的信息的收集和处理, 并将处理完之后的信息依照某个特定周期 T 转发给簇头节点.

簇头节点负责将信息传递给上级节点或者 sink 节点. 该阶段单簇能量损耗为

$$E_{s2} = E_{RX}(l_{msg}) + (m + 1)E_{TX}(l_{msg}, d) + (m + 1)T_{avg}(k)E_{idle} \quad (13)$$

簇头能耗为

$$E_{ch2} = E_{RX}(l_{msg}) + E_{TX}(l_{msg}, d) + T_{avg}(m)E_{idle} \quad (14)$$

延时为

$$T_{s2} = T_{avg}(m) \quad (15)$$

m 为簇内竞争次节点的成员数目.

3.4 路由的建立阶段

LEACH 算法实行单跳传输, 会导致远离簇头的簇群因大功率发射能量消耗过快导致网络生命期缩短. 为解决这个问题, 现在已经提出的方法是进行多跳传输, 但是这样容易造成远离 sink 的节点数据传递时延过大的问题, 所以本算法将以簇为单位构造数据聚合树.

由于每个节点的地理位置信息已知, 因而基站可以很容易计算出所有节点间的距离. 并且节点间的传输是对称的, 因此, 所有的簇头节点可构成一个带权的无向完全图. 因为在传输相同大小数据时, 节点消耗的能量和节点间的距离呈正相关, 因此把两个簇头间的距离作为边的权值, 可以利用 prim 算法计算出以簇为单位的最小生成树.

构造完成的数据传递路径如图 3:

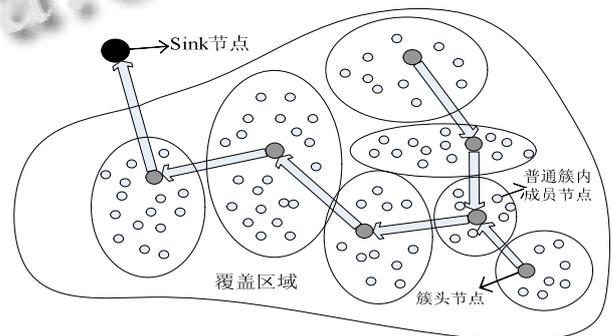


图 3 数据传递路径

在这一阶段基站负责簇头之间距离的计算, 比较及生成最小生成树. 选择距离基站最近的簇头所在的簇作为根节点. 基站将消息以 $msg(\text{父节点位置}, \text{子节点位置})$ 的形式发送, 各个簇头节点接收消息后, 将自

身位置信息与子节点信息比对后, 选择匹配成功的消息读取父节点位置. 匹配成功后不再接收信息.

单簇能量损耗为

$$N_3 = N_{ch3} = N_{ch} E_{RX}(l_{msg}) + \bar{n} E_{eda}(l_{msg}) \quad (16)$$

\bar{n} 为匹配成功时比较的次数.

该阶段次节点能耗可忽略.

3.5 稳定的簇通信阶段

簇内所有节点按照 TDMA(时分复用)时隙向簇头对的次节点发送数据, 假设每个节点在各自的时间片 t 内可以完成数据采集和发送的任务, 当自己的时间片用完时即进入休眠状态. 次节点对收到的数据进行数据融合后, 以基于触发的方式及时向簇头节点发送数据. 引起触发的事件有两个: 一是次节点两次发送数据的时间超过了最大时间间隔 T_{max} , 二是次节点累积的数据量超过了某个限定值 w . 簇头节点收到数据以后, 按照 CSMA(载波侦听多路访问)的方式沿着聚合树的路径将簇内数据和下级节点传送的数据一并传递给上一级簇头节点.

单簇能量损耗为

$$E_4 = (k-1)(k-2)tE_{sleep} + (k-1)E_{TX}(l_{data}, d) + kE_{RX}(l_{data}) + hE_{RX}(l_{data}) + (k-1)l_{data}E_{eda} + E_{TX}(l_{data}, d_{BS}) + E_{TX}(l_{data}, d_{ch}) \quad (17)$$

其中, h 为从下级簇头收到的数据数目, d_{ch} 为次节点到簇头节点的距离, d_{BS} 为簇头节点到汇聚节点的距离.

簇头为

$$E_{ch4} = E_{TX}(l_{data}, d_{BS}) + (k-2)E_{RX}(l_{data}) + hE_{RX}(l_{data}) \quad (18)$$

次节点为

$$E_{snd4} = (k-1)E_{RX}(l_{data}) + (k-1)l_{data}E_{eda} + E_{TX}(l_{data}, d_{ch}) \quad (19)$$

延迟为

$$T_4 = (k-1)t \quad (20)$$

3.6 簇延续阶段

为了延长稳定簇通信的时间, 减少因大规模重建路由造成的能量消耗. 在簇头节点的能量小于门限值 E_m (初始能量的 20%)时, 将会在本簇内局部重新选择簇头和次节点, 选择剩余能量最大的节点作为簇头节点, 次节点的选择依照已述算法. 主次节点选择完成

后分别在簇内广播自身 ID.

单簇能量损耗为

$$E_5 = E_{TX}(l_{msg}, d) + kE_{RX}(l_{msg}) + E_3 \quad (21)$$

簇头为

$$E_{ch5} = E_{TX}(l_{msg}, d) \quad (22)$$

因为此次为簇内局部选择簇头和次节点, 所以不存在竞争关系, 延迟可以忽略.

3.7 能量反馈

数据通信进行一段时间以后, 距离 sink 最近的簇群将会因为负责大量的数据转发能量消耗过快, 形成网络“热区”, 造成簇群间能量消耗不均衡. 所以引入能量反馈机制, 设置能量阈值 E_x (E_x 为初始能量的 30%), 如果某个簇群的平均能量 \bar{E}_i 小于阈值 E_x , 则发送反馈消息 $msg_{feedback}$ 给自己所有的下一级簇头, 表示该簇群不再负责数据的转发, 由下一级簇头直接将数据发送给 sink 节点.

发送反馈消息簇头消耗的能量为

$$E_{ch6} = \nu E_{TX}(l_{msg}, d) \quad (23)$$

其中, ν 为下级簇头的个数.

接收反馈消息簇头消耗的能量为

$$E_{ch6} = E_{RX}(l_{msg}, d) \quad (24)$$

3.8 新一轮簇生成

当有一半以上的簇发送过能量反馈消息时, 按照 LEACH-DCHS 方法, 重新进行新一轮的簇生成.

重新进入 2.1 到 2.6 循环.

4 仿真评估与算法分析

这里通过 OMNET++ 仿真工具进行模拟并分析 EFCA 算法的性能. 利用 NED 和 INI 配置文件描述实验环境, 具体的仿真参数设定如表 1:

表 1 模拟实验主要参数

参数	值	参数	值
节点总数 N	150	分布区域 M	[150,150]
消息长度 l_{msg}	15Byte	数据长度 l_{data}	500Byte
信号处理功耗 E_{elec}	50nJ/bit	计算功耗 E_{eda}	5nJ/bit
空闲功耗 E_{idle}	15mW	休眠功耗 E_{sleep}	10uW

放大器功耗 ϵ_{fs}	10pJ/bit/m ²	放大器功耗 ϵ_{mp}	0.0015pJ/bit/ m ⁴
节电工作时间 t	0.05sec	信道速率 s	100kbps
最大时隙 T_{max}	1sec	BS坐标	(50,150)
延迟 τ	5 μ s	发送概率 τ	0.2
数据量限定值 w	3000Byte		

仿真比较了完成 LEACH, PEGASIS 和本文提出的 EFCA 三种算法在相同的数据收集情况下, 三种算法的数据的延迟情况, 即完成相同的数据收集任务所需要的时间; 网络的生命期. 仿真结果如图 1、图 2 所示.

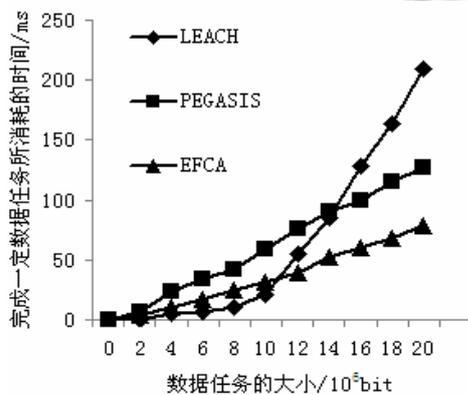


图 1 完成相同数据收集任务需要时间

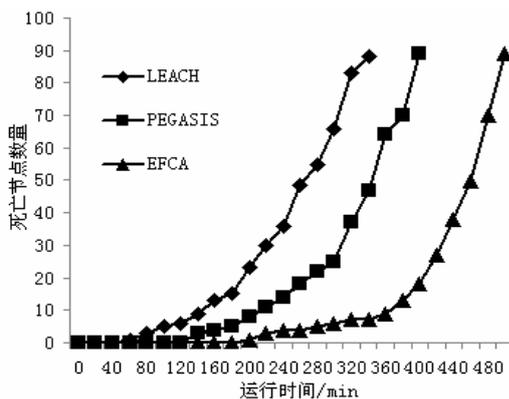


图 2 节点生命周期

4.1 网络时延分析

从图 1 可以看出当数据任务较小时, LEACH 由于采用簇头直接和汇聚节点通信的单跳传输方式, 在收集相同的数据任务时, LEACH 所用时间相对 PEGASIS 和 EFCA 都小, 而随着数据任务加大, LEACH 外沿节

点逐渐死亡, 可用节点数变少, 收集能力大幅下降. PEGASIS 采用链式传输, 传输路径过长, 数据收集速度相对较小. 而 EFCA 因采用了以簇为单位构造数据聚合树, 缩短了传输路径; 同时选择次节点负责处理簇内数据, 簇头节点负责处理下级簇头节点发送数据并行处理的方法, 在收集较大数据任务量时凸显出明显的优势, 所耗时间小于 LEACH 和 PEGASIS.

4.2 网络节能性实验分析

实验中采用约 60% 节点死亡的时间作为网络生存时间的评价标准. 因为此时剩余节点的能量已经很低, 网络的连通度也无法保证. 网络生存时间的对比如图 2 所示, EFCA 由于采用了能量反馈机制, 节点的能量消耗更加均衡, 失效节点出现时间延后, 网络生存期明显高于 LEACH 和 PEGASIS.

5 总结

本算法以簇为单位构造数据聚合树进行多跳传输, 解决了单跳传输外沿节点能耗过快和多跳传输时延过大的问题; 并引入次节点构造簇头对加快数据处理速度. 为延长簇通信的稳定期, 进行簇内簇头选举, 同时采用能量反馈机制, 均衡各簇之间能量消耗. 仿真结果表明, 在进行大批量数据收集时, 基于能量反馈的分簇路由算法, 比 LEACH 和 PEGASIS 所耗时间明显缩短, 同时有效延长了网络生命期. 但在本算法中, 进行能量反馈的最优门限值 E_x 不容易确定, 值得进一步探究, 以后将致力于此方面的研究.

参考文献

- 1 于宏毅, 李欧, 张效义. 无线传感器网络理论、技术与实现. 北京: 国防工业出版社, 2010.
- 2 孙利民, 李建中, 陈渝, 朱红松. 无线传感器网络. 北京: 清华大学出版社, 2005.
- 3 Heinzelman WB, et al. An application-specific protocol architecture for wireless microsensor sensor networks. IEEE Trans on Wireless Communications. 2002, 1(4): 660-670.
- 4 刘庆, 王培康. 无线传感器网络的安全分簇路由协议. 计算机仿真, 2009, 26(4): 167-171
- 5 陆立芳, 闫建国. 无线传感器网络路由协议的优化设计. 计算机仿真, 2010, 27(12): 125-128.
- 6 崔莉, 鞠海玲, 苗勇, 李天璞, 刘巍, 赵泽. 无线传感器网络研究进展. 计算机研究与发展. 2005, 42(1): 163-174.

(下转第 117 页)