

一种能量均衡的无线传感器网络分簇算法

耿晓义, 柴乔林, 张 擎

GENG Xiao-yi, CHAI Qiao-lin, ZHANG Qing

山东大学 计算机科学与技术学院, 济南 250061

Department of Computer Science and Technology, Shandong University, Ji'nan 250061, China

E-mail: gengxiaoyi@126.com

GENG Xiao-yi, CHAI Qiao-lin, ZHANG Qing. Clustering algorithm considering energy balance in wireless sensor networks. *Computer Engineering and Applications*, 2007, 43(33): 141-143.

Abstract: In model of multi-hop wireless sensor networks, if a cluster head is more close to the base station, it will have much more transmitting task, therefore the closer one will consume much more energy. Many classical clustering algorithms haven't focused on this problem. Based on the idea of periodically changing cluster head, this paper improves classical clustering. On the one hand, the cluster forming area has been dynamically controlled. Clusters closed to the base station have a small size, and then the little gathering staff balances the cost of transmitting. Meanwhile multi-candidate cluster heads keep the cluster structure to save the frequent cluster infrastructure energy costs. Through simulation, the new algorithm has better performance than classical method in balancing nodes' energy costs as well as extending the lifetime of the network.

Key words: Wireless Sensor Networks; base station; cluster head; candidate cluster head; communication range; life time

摘 要: 在无线传感器网络分簇算法的多跳网络模型中, 越靠近基站的簇首其转发任务越频繁, 从而造成耗能更多, 传统分簇算法中对于此问题的考虑较少。在传统算法的周期性更换簇头的思想基础上, 进行了改进, 一方面动态地对成簇范围进行控制, 使越接近基站的区域形成的簇规模越小, 减小收集簇内数据的任务, 平衡转发任务的能耗。同时引入在簇内选取多个候选簇首的机制来减少簇结构的更换, 降低频繁组簇的能耗问题。通过与传统分簇算法的仿真比较证明新算法有效地均衡了网络中节点的能耗, 延长了网络生命周期。

关键词: 无线传感器网络; 基站; 簇头; 候选簇头; 通信范围; 生存周期

文章编号: 1002-8331(2007)33-0141-03 **文献标识码:** A **中图分类号:** TP393

无线传感器网络(WSN)是一种分布密集、传感器节点随机布置的无线自组织网络。它的目的是协作的采集和处理网络覆盖区域节点收集的信息, 并报告给基站^[1]。从网络拓扑角度来说, 已提出的路由协议大概分两类: 平面路由协议与分簇路由协议^[2]。本文思想就是基于后者。由于传感器节点能量的有限性, 在分簇类算法中, 高效合理的簇结构的形成已是研究的热门问题。

1 问题提出

分簇算法中较为典型的有最大连通度算法^[3]、LEACH^[4]和TEEN^[5]等等。最大连通度算法的目标是尽量减少簇的数目, 节点和其相邻节点中具有最大连通度的节点被选为簇头, 但在选择簇头时算法未考虑能量因素。LEACH算法是通过等概率地随机循环选择簇头, 将能量负载平均的分配到网络每个节点上。但在多跳网络中, 离基站近的簇头肯定要担当更多转发数据的任务, 耗能不均, 单纯的轮流做簇头并不十分全面。同时, 每轮簇的拓扑结构都可能改变, 既费时又耗能。

本文的算法出于对节点能量负载均衡的考虑, 采用更普遍

的多跳模型, 同时又启用了多个候选节点来减少簇成员更换。首先簇头的选择上限定其当前能量应大于所有邻居节点的当前能量。候选簇头的选择需保证通信能力可覆盖全簇且每个候选簇头当前能量都应大于簇内能量的均值, 即当候选簇头成为簇头时也可以保证在簇内成员不变情况下进行新一轮数据采集。这样既平衡了能量负载又维持了簇成员的不变性。其次在簇的形成上, 为了均衡与基站距离较近的簇头节点因过多转发任务带来的能量损失, 应使这些簇头担负较小的收集本簇内数据的任务。在本算法中是按节点距离基站的远近使各簇的大小有所区别: 愈近, 簇成员数愈少; 愈远, 簇成员数愈多。

2 算法设计思想

2.1 簇头与候选簇头的选择

簇头选择是分簇算法中极为重要的一点^[6]。该算法设计采用簇头(Head)和多个候选簇头相结合方式。簇头的选择要保证其当前能量值要大于所有邻居节点当前的能量值($E_{current}$), 同时借鉴LEACH等传统算法中的轮换簇头的思想, 每轮更换簇头。为了尽量保持簇结构少做改变, 采用多个备选簇

头的方式,假设当前备选簇头个数为 M ,则其作用就是保证在 $M+1$ 轮数据采集中,簇的成员不改变。这样,在选择备选簇头时要满足如下条件(设 E_{cut} 为候选节点当前能量,NeighborList 为邻居列表,MemList 为成员列表):

- $E_{cut} \geq \bar{E}_{cut}$ (\bar{E}_{cut} 簇内节点当前能量均值);
- $E_{cut} \rightarrow$ NeighborList 包含 Head \rightarrow MemList 所有节点;
- 候选簇头个数为 M ,算法取 2。

算法采用的解决方法是对可入簇邻居节点范围进行限定。假设当前通信范围为 R ,如果候选簇头的选择是在距离簇头 λR 范围内($0 < \lambda < 1$),则成簇时只允许在簇头节点 $(1-\lambda)R$ 通信范围内的节点可入簇,如图 1 所示,中心点为本簇簇头节点,外层虚线为节点通信半径 R ,实线为成簇时限定的簇的半径 $(1-\lambda)R$,那么候选簇头只能在 λR ,即点划线范围内才有可能被选中。举例,假设 $\lambda=1/3$,则簇成员要在 $2R/3$ 范围内才能加入本簇,同时候选簇头要在 $R/3$ 范围内才可能被选中,这样当候选簇头担当簇头时,也能保证其实际通信能力能够覆盖全簇,可保持簇成员不变,不需每轮重新组簇。

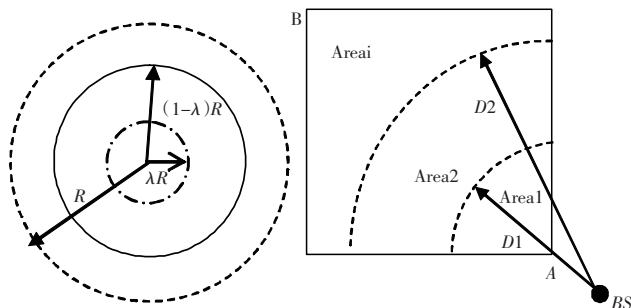


图1 候选簇头选择示意图

图2 区域划分示意图

2.2 簇的形成

文章前面提到多跳网络的能耗均衡问题。监测区域的节点分簇后,将数据发给各自簇头节点。在簇头节点将数据送到基站(BS)的过程中,也要经过多个簇头的转发。这样与基站距离近的节点肯定要担负更多的数据转发任务。本文算法设计中考虑将与基站距离近的簇成员个数减少,即依靠较少的收集数据的任务,使多出的能量用到数据转发任务中,从而在总体上平衡能量负载。

要满足簇成员个数的变化有多种方法,可直接限定个数,也可依靠能量等因素来设定。算法是依靠对节点通信范围的限定来满足成员个数的变化。假定所有节点有同样的通信能力。通常只要在节点通信范围内的节点均为一跳,即可加入簇。但为了达到簇的规模大小不一的目的,可对通信半径 R 进行限定,对于节点分布稠密的网络环境来说,可起到限定成员个数的目的。举一简单例子如图 2 所示,方形区域为节点的分布区域,基站(BS)在区域对角线上。其中在与基站相距 $D1$ 的弧形内的节点通信半径限定为 $\varepsilon_1 R$,相距 $D1$ 到 $D2$ 范围内的通信半径限定为 $\varepsilon_2 R$,其中要满足 $0 < \varepsilon_1 < \varepsilon_2 \leq 1$,即节点所处监测区域位置不同,其通信半径的设置不同,这样就可满足簇成员个数不同的目的。

理论上讲依靠节点传输数据到 BS 所需的跳数来划分较为合理。那么假定算法所用监测区域为一方形,BS 在方形对角线 BA 的延长线上,那么 B 点与基站 BS 的距离为监测区域与 BS 最远的距离 D_{max} ,估算跳数最大值 Hop_{max} :

$$Hop_{max} = \lceil \frac{D_{max}}{R} \rceil \quad (1)$$

依赖公式(1),可将区域划分块数 $AreaNO$ 与 Hop_{max} 建立如下关系:

$$AreaNO = \begin{cases} Hop_{max} & Hop_{max} < X \\ X & Hop_{max} \geq X \end{cases} \quad (2)$$

用一个 X 做界定,也是为了不至于划分区域过去繁杂。但是 X 值的确定是个很关键的问题,因为算法设计在不同的划分区域,为节点限定的通信半径是不一样的。若 X 值过大,则监测区域被划分块过多,半径划分过细,对于节点分簇操作带来麻烦;若 X 值过小,当 Hop_{max} 过大时,则距离 BS 的跳数大于 X 的多种跳数不同的节点也被划在一个区域内,可能会形成此区域过大,显然也不是一种合理划分。

以上从数学角度对方案的分析发现其诸多不便,因此本文简化了对区域的划分,在保持划分思想前提下,将区域划为基本均匀的三块。如图 2 监测模型中, B 点与 BS 相距最远距离为 D_{max} ,而 A 点与 BS 相距最近距离为 D_{min} 。另一种划分是将区域划分为三块,从地理位置上以一种基本等距的观点:

$$\begin{aligned} D1 &= D_{min} + \frac{D_{max} - D_{min}}{3} \\ D2 &= D_{max} - \frac{D_{max} - D_{min}}{3} \end{aligned} \quad (3)$$

按这种划分,方形监测区域被分为三个间距略为均等的部分。设监测区域按图 2 所示依据与 BS 的距离由近及远分为 Area1, Area2 和 Area3 三个区域,各个区域节点通信半径分别限定为 $R1, R2, R3$ 。任意节点 i ,若其与 BS 距离为 d_i 则有如下性质:

$$\text{节点 } i \in \begin{cases} Area1(R1) & \text{if } d_i < D1 \\ Area2(R2) & \text{if } D1 \leq d_i \leq D2 \\ Area3(R3) & \text{if } d_i < D2 \end{cases} \quad (4)$$

在不同的区域中对应不同的 R ,达到簇形成规模大小不同,在相对密集均匀的网络,簇成员个数也就不同。

2.3 数据传输

分簇的准备阶段完成后,就进入稳定阶段进行数据收集和传输,总体上分为簇内和簇间通信两部分。在本算法中簇内节点与簇首之间均为一跳距离,可直接通信,采用的是 LEACH 算法的 TDMA 模型,簇内所有节点按照 TDMA 时隙向簇头发送数据。簇头收集完本簇内数据就要涉及簇间通信,利用多跳数据转发来完成。离基站只有一跳的节点会最终将数据转发给基站。

3 算法详细设计

第一步:基站 BS 发一个查询包,收到包的节点不进行转发,也就是具有与基站直接通信能力的节点才可收到此包。这些节点在其邻居范围内进行当前能量($E_{current}$)的比较。其中 $E_{current}$ 最大的节点做为临时簇头发布 May_Head_Msg。收到包的节点选择收到的第一个 May_Head_Msg 的源节点做为自己临时簇头,发 May_Join_Msg,包含自己当前能量, ID, 地理位置 $Pos(x, y)$ 。临时簇头节点计算加入此簇的节点的平均能量 \bar{E}_{cut} ,然后在这些节点中挑选出离自己距离较近且当前能量($E_{current}$) $\geq \bar{E}_{cut}$ 的两个节点。设其中远一点的节点离簇头的距离为 dd (i 为该临时簇头节点 ID 号,则设

$$\lambda_i = \frac{dd(i)}{R} \quad (R \text{ 为节点的最大通信半径})$$

第二步: 临时簇头将 λ_i 值发给 BS。BS 将收到的不同的 λ_i 值中的最大值定为 λ 。根据 λ 值可以设定公式(4)中当前节点半径:

$$\begin{cases} R3 = (1-\lambda)R \\ R2 = p(1-\lambda)R, \quad 0 < p \leq p \leq 1 \\ R1 = q(1-\lambda)R \end{cases} \quad (5)$$

p, q 取值不同使各区域通信半径设置不同。仿真中可对不同的 p, q 值进行实验。

第三步: BS 广播 Area_Msg 包, 包含公式(3)中的 $D1, D2$ 、公式(5)中的 R_i 值、 λ 值, 及 BS 的地理位置 $Pos(x, y)$ 。在监测区域的节点收到包后计算自己与 BS 之间距离 d_i , 用公式(4)和(5)的规则选择自己 R_i 作为自己当前成簇可用通信半径 R_{cur} 。同时转发数据包, 并记录上一跳信息。

第四步: 各节点根据选择的 R_{cur} 进行邻居发现。广播 Hello_Msg 包, 包含节点 ID 和当前能量。若发现自己 $E_{current}$ 为邻居中最大的, 则宣布自己为簇头, 发 Head_Msg, 包含簇头地理位置、ID、 R_{cur} 。普通节点选择收到的第一个 Head_Msg 的源节点作为自己的簇首(Head), 发 Join_Msg, 包含 ID、 $E_{current}$ 、 $Pos(x, y)$ 。每个节点只允许加入一个簇, 当再收到其它节点的 Head_Msg 时, 则宣布自己为 Gate_node(网关节点)记录多个簇首信息, 并发 Gate_node_Msg, 通知簇首。

第五步: 当分簇初步完成, 簇首节点计算出的簇内节点能量均值 \bar{E}_{cur} , 根据各节点 Join_Msg 中的信息, 找出满足: $E_{current} \geq \bar{E}_{cur}$ 且与簇首距离小于等于 λR 的两个节点, 作为候选节点, 根据能量大小规定候选顺序。簇首发布消息 Cand_node_Msg, 簇内各节点记录候选节点信息。

第六步: 进入稳定阶段, 进行数据收集与传输。簇内按 2.4 节介绍的过程采用 TDMA 方式进行数据收集。簇间多跳转发, 利用第三步中保留的上一跳信息进行转发操作。

第七步: 当新一轮开始时, 各簇启用候选节点 1, 这些候选节点发 Be_Head_Msg, 各簇内节点更新自己簇首信息。然后开始第六步数据收集传输。这样三轮之后又进入第一步重新组簇。

以上为算法的全部过程, 其中对第二步公式 5 进行一下解释。前面 2.2 节提到对于候选簇头除了能量要求外, 还要求其能覆盖簇内所有节点。在算法中要求候选簇头与簇头距离在 λR 内, 那么候选簇头可以覆盖的实际区域为 $(1-\lambda)R$, 现在簇的半径范围在 $[q(1-\lambda)R, (1-\lambda)R]$ 内, 所以也可以满足覆盖度的要求。

在能量计算方面, 规定在距离为 d 的范围传输 k -bit 数据的能耗为^[7]:

$$E_{Tx}(k, d) = k * E_{elec} + \epsilon_{amp} * k * d^2$$

接收数据所消耗的能量为:

$$E_{Rx} = k * E_{elec}$$

4 仿真结果

仿真主要是对网络存活时间和出现死亡节点到节点全部死亡的时间间隔两个方面的比较。如果存活时间长, 且从区域开始出现死亡节点到最终节点全部死亡的时间间隔小, 就反映出网络中节点的能量均衡情况越好, 能量的使用越高效。

算法仿真在 400×400 的方形区域, 初始节点密度较大, 个

数为 200 个, 分布较为均匀。仿真中忽略由信号碰撞和无线信道干扰等随机因素带来的影响。节点死亡只发生在能量为零时。假定每个节点最初都具有 5J 的初始能量, 并且假设 $E_{elec} = 50 \text{ pJ/bit} \cdot \text{m}^2$; $\epsilon_{amp} = 100 \text{ pJ/bit/m}$; 采用 OMNET++ 软件^[8]进行仿真, 首先对前面公式(5)中不同的 p, q 取值进行仿真。取两组数据: $p=0.9, q=0.8$; $p=0.8, q=0.6$; $p=0.9, q=0.5$ 。图 3 为采用这两种数据进行仿真(①为第一组数据②为第二组③为第三组)时, 网络中生存节点个数(Numbers of nodes alive)随时间的变化。从图 3 仿真结果可以看出, 当 p 与 q 取值差增大时, 即取第二种情况时, 节点生存情况较好, 能量均衡效果更佳。将本算法与经典分簇算法 LEACH 进行比较。图 4 的仿真结果。两种算法基本思想都是节点轮流作簇头的原则。本算法也采用这一思想。仿真结果表明新算法比 LEACH 算法有更好的网络生存周期, 而且节点死亡整体跨度小, 能量均衡性更佳。

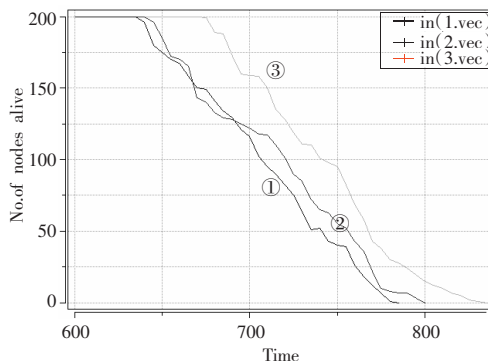


图 3 不同 p, q 值比较

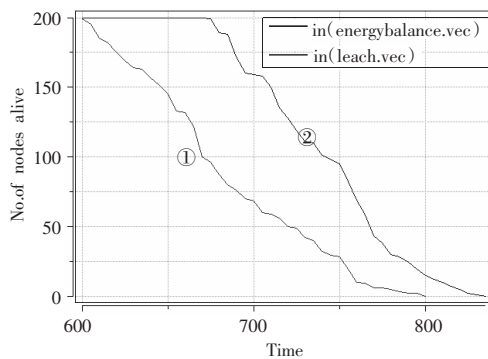


图 4 本文算法②与 leach①比较

5 结束语

论文通过对簇大小不同的控制, 以及对候选簇头的设计, 使算法在能量均衡方面有较好的效果, 延长了网络生命周期。但算法对于簇半径大小的设置及 λ 值的选择比较适用于节点密集且分布较为均匀的网络环境, 以后可以在这个方面进行更为全面的优化。(收稿日期: 2007 年 4 月)

参考文献:

- [1] Calhoun P R, Johansson T, Charles E, et al. DIAMETER mobile IP application, draft-ietf-aaa-DIAME-TER-mobileip-18[S], 2004-05.
- [2] 沈波, 张世永, 钟亦平. 无线传感器网络分簇路由协议[J]. 软件学报, 2006(7): 1588-1600.
- [3] 吴迪, 刘英学, 冯永新, 等. AD Hoc 网络中一种基于权值的分簇算法[J]. 小型微型计算机系统 2006, 27: 202-206.