

无线传感器网络中 Leach-c 路由协议的研究与改进

蒋 华, 刘伟强, 王 鑫

(桂林电子科技大学 计算机科学与工程学院, 广西 桂林 541004)

摘 要: 针对 Leach-c 协议中簇头节点与基站只能单跳通信, 频繁进行分簇的缺点, 在分析与仿真 Leach-c 路由协议的基础上提出了一种改进协议 Ileach-c (Improved low energy adaptive clustering hierarchy-centralized). 该协议在分簇时采用固定与非固定相结合的分簇的方式, 在建立簇间路由时利用 Dijkstra 算法实现簇头与基站最短路径多跳通信, 所有簇头将数据经最短路径多跳传输给基站, 达到了延长网络生存时间的效果. matlab 仿真结果表明: Ileach-c 协议比 leach-c 协议延长了网络生存时间, 使网络能量负载更均衡.

关键词: 固定分簇; 非固定分簇; 迪杰斯特拉算法; 最短路径

中图分类号: TP393

文献标识码: A

文章编号: 1000-7180(2014)12-0043-05

Research and Improvement of Leach-c Routing Protocol in Wireless Sensor Network

JIANG Hua, LIU Wei-qiang, WANG Xin

(School of Computer & Control, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: In order to overcome the shortcoming of frequent clustering and “One hop communication” between cluster header node and base station in Leach-c protocol, this paper proposed an Ileach-c (Improved low energy adaptive clustering hierarchy-centralized) protocol through analyzing and simulating the previous Leach-c routing protocol. The protocol combine fixed clustering approach and non-fixed clustering approach when it cluster. On the other hand, it achieve that cluster head node can be more hops communication in shortest path with base station by using Dijkstra algorithm. All cluster heads send messages to base station via shortest path by hops communication. The experimental results show that Ileach-c protocol is better than Leach-c protocol in reducing the network transmission cost and prolong the life of wireless sensor network and getting more balanced energy load.

Key words: fixed clustering; non-fixed clustering; Dijkstra algorithm; Shortest path

1 引言

作为典型的无线传感器网络协议, Leach-c 是 Leach 的一个经典改进协议, 它是由 Heinzelman 等人在 Leach 协议的基础上加以改进提出的基于基站控制的协议^[1]. 文献[2]中, 在 Leach-c 的基础上提出新协议 EBLEach-c; 文献[3]中, 在 Leach-c 协议基础上采用遗传算法代替模拟退火算法先取簇头节点; 文献[4]中, 在 Leach-c 协议的基础上改进节点信息发送机制, 它采用信息趋势函数 TINA 来决定

节点是否进行信息发送, 减少了网络信息通信量. 文献[5]中, 在 Leach-c 协议的基础上考虑了簇内多跳通信并且评估簇头能量损耗, 将其用于下一次簇头的选择; 文献[6]中, 在 Leach-c 协议基础上采用 susan 算子算法取代模拟退火算法选取簇头节点; 文献[7]中, 在改进 Leach-c 协议时提出重定向的思路, 利用重定向的方法建立了簇间路由; 上述研究考虑了簇头间多跳通信并延长了网络生存时间, 但没有考虑能耗最少最佳多跳路径通信.

本文在 Leach-c 协议的基础上提出了一种新的

收稿日期: 2014-01-12; 修回日期: 2014-03-09

基金项目: 国家自然科学基金重点项目(61262074)

改进协议 Ileach-c.

2 Leach-c 协议描述

Leach-c 协议是一种周期性协议,在它每个周期的开始阶段,所有节点把自己的位置信息和剩余能量值发往基站.基站在收到这些信息后,首先计算所有节点的平均能量值,把能量不低于平均能量值的节点作为候选节点.然后使用模拟退火算法从中选举出使目标函数较小的分簇方式,最后把分簇结果广播到网络中.如果节点收到的簇头集合中有自己的 ID,则把自己作为簇头;如果没有,则与相应的簇头建立联系,成为其簇成员节点并等待相应的 TDMA 时隙到来后向簇头传输数据^[8].无线传感器中节点发送信息所需的能量与其发送的距离的指数成正比,故发送距离越远发送能量消耗越多,信号干扰越大^[9].对于 Leach-c 协议,当监测范围较大时,因部份簇头离基站太远,每次发送信息所需能量过多,容易导致簇头节点死亡.另外,频繁的分簇导致网络节点频繁接受基站信息也降低了网络的生存时间.

3 改进协议 Ileach-c 描述

Ileach-c 协议与 leach-c 协议一样,也是分轮进行通信.但不同的轮数时,协议对节点有不同的处理,这涉及到固定分簇与非固定分簇相结合的问题.

设 N 为通信轮数变量,它在每一轮通信后会自加 1,初始值为 1. N_{avg} 为簇平均节点数,初始值为节点总数.每个节点均有唯一的 ID 号.

当 $N \bmod N_{avg} = 1$ 时,第一阶段为分簇阶段.所有节点先将自己的位置信息、能量剩余信息和 ID 号等信息发送给基站,然后基站计算机根据各个节点的位置信息和能量信息利用模拟退火算法得出分簇方案,选出簇头节点,统计每个簇的平均成员节点数并赋值给 N_{avg} 且令 $N=1$.第二阶段为簇间路由建立阶段.基站计算机根据第一阶段选出来的簇头节点运用 Dijkstra 最短路径算法计算出所有簇头与基站进行多跳通信的最佳路径,然后基站将分簇信息向所有节点广播,让所有节点知道自己是普通节点还是簇头节点,如果是普通节点它将知道自己属于哪个簇,如果是簇头节点,它将知道自己的下一跳簇头节点是谁,即在信息传输时簇头节点将自己收集到的信息发送给它的下一跳,直到信息最终被送到基站.第三阶段为数据传输阶段.所有普通节点将自身收集到的信息以及自身的编号信息与剩余能量

信息打包采用 TDMA 方式发送给簇头节点,然后簇头将收集到的所有信息融合后经过第二阶段计算出的最短路径多跳传送到基站.簇头将信息发送到基站后,从成员节点中选择剩余能量最多的节点担任下一任簇头节点并广播下一任簇头节点的 ID,然后进入下一个循环.

当 $N \bmod N_{avg} \neq 1$ 时,协议采用固定分簇方式,即沿用上一轮的簇结构不作变动,簇头也在上一轮通信结束时已选出,簇间通信路由也不作变动,直接进入通信第三阶段进行数据发送.

由上面协议描述可知网络每隔 N_{avg} 轮通信后重新进行分簇,而在 N_{avg} 轮通信轮内网络分簇固定不变,既保持了非固定分簇的健壮性也保留了固定分簇的节约能量的优点,图 1 为协议流程图.

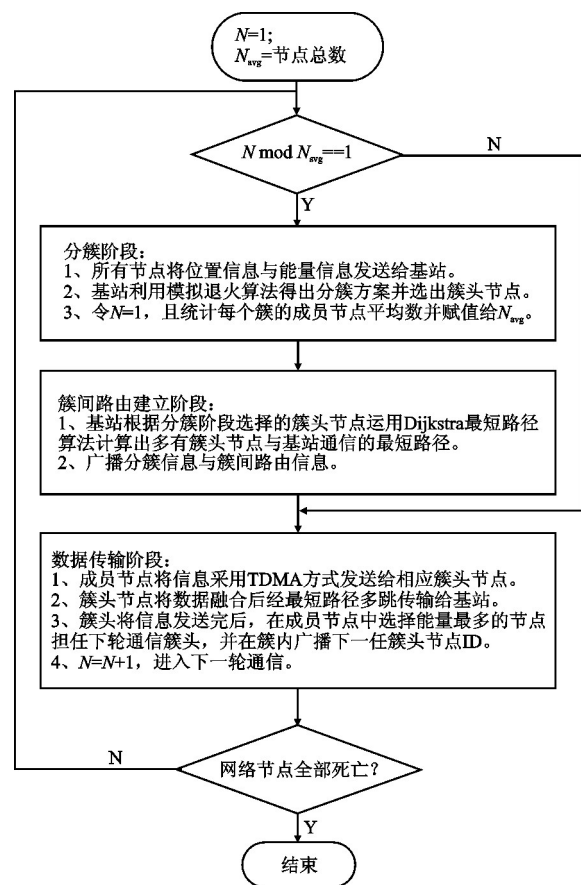


图 1 协议流程图

3.1 无线通信能量模型

本协议能量模型与融合模型采用文献[10].式(1)中 d 为传输距离, $E_{Tx}(l, d)$ 表示传感器节点发送 l bit 数据通过距离 d 时的能耗,由发射电路耗损和功率放大耗损两部分构成.功率放大耗损根据发送者和接收者之间的距离分别采用自由空间模型和多路径衰减模型. E_{elec} 为发射电路的耗损能量. ϵ_{fs} 和 ϵ_{mp}

分别表示两种信道模型下功率放大所需能量. 式(2)表示接收 l bit 数据的能量耗损, 仅由电路耗损引起. 式(3)为将接收到的 n 个节点发送过来的 $n \cdot l$ bit 数据与本身的 l bit 数据融合, 融合成 l bit 数据再发送出去.

数据发送:

$$E_{Tx}(l, d) = \begin{cases} E_{elec} \cdot l + \epsilon_{fs} \cdot l \cdot d^2 & d < d_0 \\ E_{elec} \cdot l + \epsilon_{mp} \cdot l \cdot d^4 & d \geq d_0 \end{cases} \quad (1)$$

$$\text{数据接收: } E_{Rx} = E_{elec} \cdot l \quad (2)$$

$$\text{数据融合: } E_{Gx} = E_g \cdot l \quad (3)$$

3.2 簇间路由的形成

Dijkstra 算法是经典的最短路径算法, 在带权图结构中, 它能计算出一点到其他各点代价最小的路径. Dijkstra 算法的基本思想是: 把带权图结构中所有节点分成两组, 第一组包括已确定最短路径的节点, 初始时只含有一个原点, 记为集合 S ; 第二组包括尚未确定的最短路径节点, 记为集合 $V-S$. 按路径权值递^①增顺序逐个把 $V-S$ 中的节点加到 S 中去, 直至从原点出发可以到达的所有节点都包括到 S 中. 在这个过程中, 总保持从原点到集合 S 中各节点的路径长度都不大于从原点到集合 $V-S$ 中任何节点的路径长度. 在 ILeach-c 协议中将簇头节点与基站的位置分布看成完全连通图 $T(V, \{E\})$, V 表示节点, E 表示边或权值. 基站看成是原点 V_0 , $E(a, b)$ 表示簇头 a 与簇头 b 的权值函数, 其意义是簇头 a 向簇头 b 发送 l 比特字节所消耗的能量, 加上 b 簇头接收 l 比特字节所消耗的能量, 根据能量模型函数(1)、(2), 可知, 权值函数的表达式如下:

$$E(a, b) = \begin{cases} 2E_{elec} \cdot l + \epsilon_{fs} \cdot l \cdot d^2 & d < d_0 \\ 2E_{elec} \cdot l + \epsilon_{mp} \cdot l \cdot d^4 & d \geq d_0 \end{cases} \quad (4)$$

又,

$$d = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (5)$$

即:

$$E(a, b) = \begin{cases} 2E_{elec} \cdot l + \epsilon_{fs} \cdot l \cdot [(x_a - x_b) + (y_a - y_b)] & d < d_0 \\ 2E_{elec} \cdot l + \epsilon_{mp} \cdot l \cdot [(x_a - x_b) + (y_a - y_b)]^2 & d \geq d_0 \end{cases} \quad (6)$$

说明: x_a, x_b, y_a, y_b 为 a, b 簇头对应的位置坐标.

ILeach-c 协议中 Dijkstra 算法选择簇头到基站

的最短路径具体步骤如下:

(1) 初始化, 包括对两个集合进行初始化与对权值的初始化. 基站计算机采用模拟退火算法形成了分簇方案并选举出簇头节点后进入 Dijkstra 算法应用阶段. Dijkstra 算法运行初始时, 集合 S 最初只包括基站 V_0 , 即 $S = \{V_0\}$, 集合 $V-S$ 包含所有簇头节点, 即 $V-S = \{V\}$, V 表示所有簇头节点. $V-S$ 中任意一个簇头节点 V_i 到基站 V_0 对应的权值为 $E(V_i, V_0)$, $E(V_i, V_0)$ 的初始化按式(5)进行, 簇头与簇头之间的权值 $E(V_i, V_j)$ 同样也按式(5)进行. $p(V_i)$ 是用来存放簇头节点的下一跳地址, 初始时所有簇头节点的下一跳地址为基站, 即 $p(V_i) = V_0$.

(2) 从 $V-S$ 集合中选取一个到基站 V_0 权值最小的簇头 V_m 加入 S 中, 并修改基站 V_0 到 $V-S$ 中所有簇头的权值. 修改方法是对于 $V-S$ 中任意一个簇头节点 V_k , 若连通图中 V_0 到 V_m 的权值 $E(V_0, V_m)$ 加上 V_m 到 V_k 的权值 $E(V_m, V_k)$ 小于 V_0 到 V_k 的权值 $E(V_0, V_k)$, 则用 V_0 到 V_m 的权值加上 V_m 到 V_k 的权值代替 V_0 到 V_k 原来的权值, 即: $E(V_0, V_k) = E(V_0, V_m) + E(V_m, V_k)$, 并将 V_m 作为 V_0 到 V_k 路径上的中间点, 数据传输时 V_k 的下一跳即为 V_m , 由 $p(V_k)$ 存储, 即 $p(V_k) = V_m$, 此时我们称 V_m 为 V_k 的下一跳, 反之, V_0 到 V_k 的当前最小权值不变. 即:

$$E(V_0, V_k) = \min\{E(V_0, V_k), E(V_0, V_m) + E(V_m, V_k)\} \quad (7)$$

(3) 若集合 $V-S$ 已空, 则算法结束, 权值函数的值即为基站到达各簇头节点最小代价. 若 $V-S$ 不为空, 则转步骤(2). 该算法流程图如图 2 所示.

经过上面三个步骤后 $E(V_i, V_0)$ 里存放的就是簇头节点 V_i 通过能耗最少的最短路径发送信息到基站的能量消耗量, $p(V_i)$ 里存储的是簇头 V_i 发送信息到基站 V_0 的能耗最少的最短路径路由. 簇头向基站发送信息时根据 $p(V_i)$ 的指引经过多跳传送, 最终将信息传输到基站且所耗能量最小. 比如: 簇头 V_a 要将信息 Mes 发送给基站 V_0 , 通过查询发现 $p(V_a) = V_b$, 即簇头 V_a 的下一跳是簇头 V_b , 那么簇头 V_a 将信息 Mes 发送给簇头 V_b , 信息到达簇头 V_b 后问题变成了簇头 V_b 要将信息 Mes 发送给基站 V_0 , 再通过查询发现 $p(V_b) = V_c$. 同理, 簇头又将信息 Mes 发送给簇头 V_c , 再通过查询, 发现 $p(V_c) =$

① 路径权值: 从一个节点到另一个节点所花费的代价.

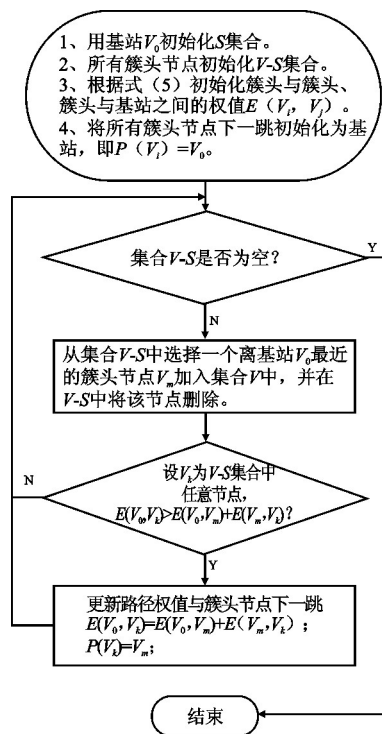


图2 路径生成流程图

V_0 , 所以簇头 V_c 将信息 Mes 发送给基站 V_0 , 这样信息 Mes 从簇头 V_a 开始, 经过 V_b 、 V_c 两跳最终到达了基站. 路径 $V_a \rightarrow V_b \rightarrow V_c \rightarrow V_0$ 就是簇头 V_a 将信息 Mes 发送到基站 V_0 的最短多跳通信路径, 而 p 的内容是由上面步骤(1)~(3)确定.

4 仿真结果分析

该文通过 Matlab 分别对 Leach-c 协议与 Illeach-c 协议进行了模拟仿真, 分别从网络的生存时间与能量负载均衡两方面来评价新协议的性能. 仿真的前提是所有节点能量相同, 所有节点位置不变, 所有普通节点每次发送的数据量相同, 节点死亡后不再添加新的节点. 实验参数设置如表 1 所示.

表1 实验参数设置

参数	取值
节点个数	100
监测区域	100×100
节点初始能量	1 J
数据包	4 000 bit
控制包	100 bit
E_{elec}	5 nJ/bit
ϵ_{fs}	20 pJ/(bit $\cdot$ m $^{-2}$)
ϵ_{mp}	0.003 pJ/(bit $\cdot$ m $^{-4}$)
E_g	10 pJ/(bit $\cdot$ signal $^{-1}$)
d_0	30 m

4.1 网络生存时间

图 3 表示采用 Leach-c 协议和 Illeach-c 协议原理进行无线传感器信息传输时节点的死亡情况, 最大轮数为 2 000 轮, 文中以 50% 节点死亡时所经过的轮数作为网络生存时间. 文中对第一个节点开始死亡轮数、20% 节点死亡轮数、50% 节点死亡轮数、70% 节点死亡轮数、全部节点死亡轮数进行了比较, 结果如表 2 所示.

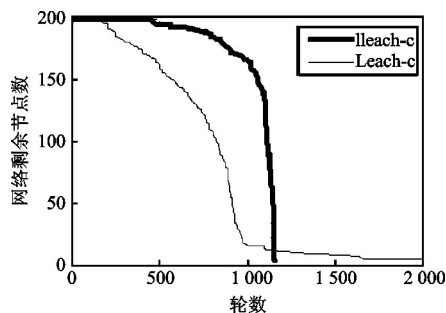


图3 网络生存节点数与轮数关系图

表2 两种协议不同生命期对比结果

死亡节点数	Leach-c	Illeach-c
1	133	451
20%	508	1 022
50%	823	1 112
70%	890	1 135
100%	2 000	1 155

从表 2 中的结果可以看出 Illeach-c 协议比 Leach-c 协议的生存周期提高了 35%, 第一个节点死亡时间更是大大的推迟了 239%, 20%、70%、100% 节点死亡时间也都得到了很大的改善. 之所以有如此大的改善是因为簇头节点发送数据到基站是经过最短路径多跳通信路径, 该路径能最大限度的节约通信能量.

4.2 负载均衡性

文中用某时刻的能量方差函数 $D_E(t)$ 来衡量网络负载均衡性, $D_E(t)$ 越小表示网络负载均衡性越好.

能量方差函数:

$$D_E(t) = \frac{\sum_{i=1}^N [E_i(t) - G(t)]^2}{N} \quad (8)$$

式中, $E_i(t)$ 表示标记为 i 的节点在 t 时刻的能量, $G(t)$ 表示 t 时刻网络平均能量, N 表示网络节点总数.

图 4 是两协议的能量方差与轮数关系图, 从图中可以看到 Illeach-c 协议的方差一直远低于 Leach-

c 协议的方差,表明 ILeach-c 协议能耗负载均衡性远高于 Leach-c 协议,有利于网络生命周期的延长.另外,观察图 4 中的 ILeach-c 协议的曲线可以发现该曲线先是比较平缓,而到 1 000 轮的时候却开始陡然下降,说明:相比于 Leach-c 协议,ILeach-c 协议中能耗能更加均衡的分配在每个节点上,导致大部份节点是在一个短时间段内同时死亡.

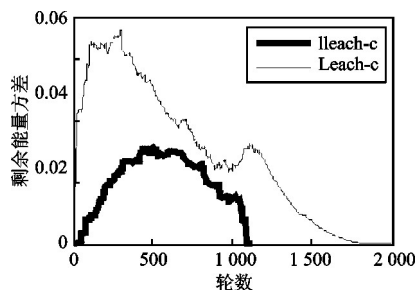


图 4 网络剩余能量与轮数关系图

从网络生命周期与能量消耗情况两方面均说明该文提出的协议具有如下优点:(1)采用固定分簇与非固定分簇相结合的方式,在不太影响网络健壮性的情况下提高了能量利用率;(2)利用 Dijkstra 算法得出能耗最少的多跳通信路径,不但均衡了网络能量,而且提高了能量利用率.

5 结束语

本文通过对 Leach-c 协议的改进,使得网络节点生存时间得到有效的延长,优化了网络能量负载.在 ILeach-c 协议中,虽然 Dijkstra 算法很复杂,但它是运行在基站计算机内,只将结果广播给传感器节点,无需消耗节点能量.所以,它继承了 Leach-c 协议的优点,使得网络生存时间有了进一步延长,能量负载能更加均匀地分布在网络中.但是 ILeach-c 协议运行时靠近基站的簇头节点负担较重,容易死亡,下一步研究方向是如何解决靠近基站的簇头节点容易死亡的问题.

参考文献:

- [1] Anand J, Tamilselvi J J, Janakiraman S. Analyzing the performance of diverse LEACH algorithms for wireless sensor networks[J]. International Journal of Advanced Networking and Applications, 2012, 4(3): 1610-1615.
- [2] Yan Xiaoying, Cheng Guojian, Sun Tao. WSN clustering routing algorithm of energy consumption balance [J]. Computer Engineering, 2012, 38(14): 79-81.
- [3] Rahmanian A, Omranpour H, Akbari M, et al. A novel genetic algorithm in LEACH-C routing protocol for sensor networks [C]// Canadian Conference on Electrical and Computer Engineering (CCECE). Canadian, 2011: 1096-1100.
- [4] Hanifi A, Amiri J, Kazimov T. Base station location for energy efficient in wireless sensor network [C]// Proceedings of the 2012 IV International Conference. Problems of Cybernetics and Informatics (PCI). Baku Azerbaijan, 2012.
- [5] Shuo Shi, Xinning Liu, Xuemai Gu. An energy -efficiency optimized LEACH-C for wireless sensor networks [C]// International ICST Conference on Communications and Networking. Orlando, Florida, 2012: 487-492.
- [6] 唐起涛, 刘蓉, 张燕, 等. 基于 SUSAN 算子的 LEACH-C 路由算法[J]. 微计算机信息, 2012, 28(10): 469-470.
- [7] 赵鹏飞, 金心宇. 一种无线传感器网络的新型集中式能量有效型路由协议[J]. 传感技术学报, 2006, 19(1): 30-49.
- [8] 蹇强, 龚正虎, 朱培栋, 等. 无线传感器网络 MAC 协议研究进展[J]. 软件学报, 2008, 19(2): 389-403.
- [9] 张招亮, 陈海明, 黄庭培. 无线传感器网络中一种抗无线局域网干扰的信道分配机制[J]. 计算机学报, 2012, 35(3): 504-517.
- [10] 崔永锋, 刘伟. 一种基于无线传感器网络的动态拓扑能量有效成簇增量算法[J]. 微电子学与计算机, 2012, 23(5): 1222-1232.

作者简介:

蒋华 男, (1963-), 教授, 研究生导师. 研究方向为数据库系统、信息安全.

刘伟强(通讯作者) 男, (1987-), 硕士研究生. 研究方向为无线传感器网络. E-mail: lwq7153910@163.com.

王鑫 男, (1976-), 副教授, 硕士生导师. 研究方向为网络与信息安全.