

# 基于蚁群的无线传感器网络双簇头算法

苏 淼, 钱 海, 王煦法

(中国科学技术大学计算机科学技术系安徽省计算与通讯软件重点实验室, 合肥 230027)

**摘要:**在无线传感器网络分层路由协议 LEACH 的基础上,重新定义了“轮”的概念,把每一轮划分成 3 个阶段而不是传统的 2 个阶段。提出基于蚁群的双簇头算法,根据信息素浓度在每一簇中选择具有分工特征的主簇头和副簇头,分别进行数据收集、融合、传输等工作。与 LEACH 和 LEACH-C 算法进行实验比较,该算法较好地平衡了网络的能量消耗,延长了网络的生命周期。

**关键词:** 轮; 蚁群算法; 双簇头

## Ant Colony-based Double Cluster-heads Algorithm for Wireless Sensor Networks

SU Miao, QIAN Hai, WANG Xu-fa

(Anhui Key Lab of Software in Computing and Communication, Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027)

**【Abstract】**This paper redefines the conception of “round” on the basis of Low Energy Adaptive Clustering Hierarchy(LEACH) for wireless sensor networks, and partitions a round into three phases, instead of traditional two phases. A new algorithm named Ant Colony-based Double Cluster-Heads Algorithm(ACDCHA) is proposed through introducing the idea of ant colony algorithm. A master cluster head and a vice cluster head are selected in each cluster according to the pheromone concentration, tasks such as data collection, fusion, transition, etc. are allocated respectively to these two kinds of cluster head. Simulation results show that, compared with LEACH and LEACH-C, the algorithm can get a better balance of the energy dissipation and prolong the network lifetime.

**【Key words】** round; ant colony algorithm; double cluster-heads

### 1 概述

无线传感器网络由分布在一定空间范围的大量的传感器节点组成,节点具有无线通信能力和一定的信息处理能力。分布式无线传感器网络能够感知环境,具有广阔的应用前景(如军事、环境监测、医疗监护等),是当前网络研究的热点之一。

传感器节点一般体积较小,仅提供低能量和较弱的计算能力,而且能量一旦耗尽将不可再生。近年来许多研究集中于如何节省能量和延长网络的生命周期,其中分层和数据融合的结合是最有影响的方法之一<sup>[1]</sup>。根据分层路由协议,网络被划分成不同簇,每一个簇由一个簇头和簇成员组成,多个簇头形成高层的网络,簇头节点负责簇内信息的收集和融合处理,以及转发数据到基站(BS)。一般基站远离传感器节点区域,所以作为簇头节点比非簇头节点的能量紧张得多。

大多数分层路由协议是让资源受限的传感器节点承担簇头的任务。根据簇头产生方式的不同,可以把簇头产生算法分为 4 类。第 1 类是提前选择簇头并且在整个系统生命时间内不变的,这些节点将很快用尽它们有限的能量。一旦簇头耗尽了它们的能量,它不再工作,所有属于本簇的节点失去了通信能力。第 2 类如 LEACH<sup>[2]</sup>,它由节点根据某个阈值自主决定是否当选簇头,每轮产生的簇头没有确定的数量和位置,网络中的簇头分布在覆盖区域内非最优化,簇头直接与基站通信能耗较大,实际应用中很难平衡节点的能量消耗。第 3 类是通过节点之间的信息交互动态产生簇头,如

HEED<sup>[3]</sup>,它是一种完全分布式的成簇算法,随机选择簇头节点,选举概率与该节点的剩余能量直接相关,它仅适用于同构网络。第 4 类是由基站基于整个网络信息挑选簇头,如 LEACH-C<sup>[4]</sup>,每个节点把自身地理位置和当前能量报告给基站,基站根据所有节点的报告计算选出合适数量和最优地理位置的簇头集合,然后基站把簇头集合和簇的结构广播出去。这类方法需要基站知道全局的信息。

分层路由协议中簇头的产生方法、数量和位置决定了最终形成的簇的结构、大小和数量,也影响了节点的能量耗费进度和网络的生命周期<sup>[5]</sup>。

本文在 LEACH<sup>[2]</sup>算法的基础上,把每一轮(round)划分成 3 个阶段而不是通常的 2 个阶段,结合蚁群算法<sup>[6]</sup>的思想,提出了基于蚁群的双簇头算法(ACDCHA)。算法利用信息素浓度在每一簇中分别选择一个主簇头(Master-Cluster Head, MCH)和一个副簇头(Vice-Cluster Head, VCH),并进行工作分工。如图 1 所示, MCH 收集数据,并把融合后的数据发送给距离较近的 VCH,由 VCH 转发数据给较远的基站。MCH 仅与 VCH 通信交流,而不与基站直接通信,从而可以大大节省 MCH 的一次能量耗费,较好平衡了整个网络的能量负载,达到延长网络生命周期的目的。

**作者简介:** 苏 淼(1978 - ),女,硕士研究生,主研方向:计算智能; 钱 海,讲师、博士;王煦法,教授、博士生导师

**收稿日期:** 2007-07-15 **E-mail:** ssummm@mail.ustc.edu.cn

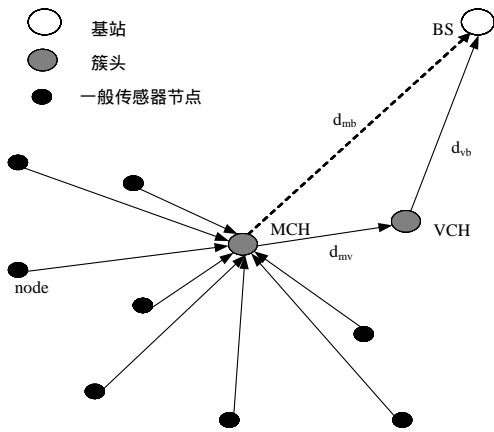


图1 双簇头示意图

## 2 基于蚁群的双簇头算法描述

本文修改了传统的无线传感器网络分层算法中“轮”的定义,把每一轮划分成3个阶段:建立(set-up)阶段,数据传输(steady-state)阶段和轮完成(finish)阶段,如图2所示。在无线传感器网络中运用简化的蚁群算法<sup>[6]</sup>,动态改变每对传感器节点间信息素浓度,通过对信息素浓度的评估来决定主、副簇头的选择及簇的形成。

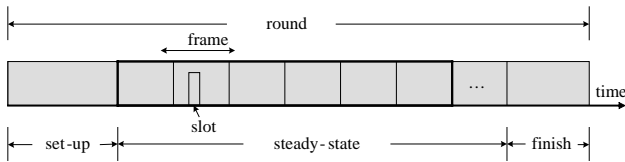


图2 ACDCHA 中轮的3阶段划分

本文假设如下:(1)基站(BS)固定并远离传感器;(2)网络中的所有节点是平等的且能量受限的;(3)所有节点能够到达BS;(4)对称的扩散通道;(5)簇头执行数据压缩。

### 2.1 MCH 和 VCH 的选择及簇的形成

本文算法的工作主要体现在 MCH 和 VCH 的选取上。

#### 2.1.1 MCH 的选择及簇的形成

初始时,BS 均匀抛撒蚂蚁,蚂蚁数为簇头数,收到蚂蚁的节点为第1轮的MCH。

MCH 广播消息,宣布自己为本轮的主簇头,收到此广播消息的节点建立与 MCH 之间虚拟的边(初始信息素浓度为0),并在此边上增加一定量的信息素  $Q/d$  ( $Q$  为一预先设定的参数; $d$  为两者之间的距离)。

如果节点  $i$  收到 MCH 的广播消息后,计算相应边上的信息素浓度:

$$\tau_{mi}(t) = \frac{Q}{d_{mi}} \times (1 - \rho) \quad (1)$$

其中,  $\rho = \frac{E_{init} - E_m}{E_{init}}$ ,  $\rho$  是信息素挥发率,与发送消息的节点的能量水平有关。

节点  $i$  根据与自己相连边上的信息素浓度大小确定自己属于哪一个簇,选择信息素浓度大的边向对应的 MCH 发送应答信息,加入该簇成为其成员。

数据传输结束时,成员节点向其主簇头发送信息报告其剩余能量,同时在相应的边上增加式(2)确定量的信息素:

$$\Delta\tau_{mi} = \alpha \times \frac{e_i}{e_{init}} \quad (2)$$

其中,  $e_i$  是节点  $i$  当前的能量水平;  $e_{init}$  是节点  $i$  的初始能量水平;  $\alpha$  是调节因子,本文取 10。

则此时 MCH 与节点  $i$  的边上的信息素浓度为

$$\tau_{mi} = \tau_{mi}(t) + \Delta\tau_{mi} = \frac{Q}{d_{mi}} \times \frac{E_m}{E_{init}} + \alpha \times \frac{e_i}{e_{init}} \quad (3)$$

在本轮的轮结束阶段,MCH 根据信息素浓度大小决定下一轮的 MCH,并向新的 MCH 发送“移交”信息。收到“移交”信息的节点即为下一轮的 MCH。

可见 MCH 的工作有:(1)向邻居节点发布消息,宣布自己为本轮的主簇头;(2)选择本轮的 VCH;(3)收集数据,并转发给 VCH;(4)计算下一轮的 MCH;(5)向下一轮的 MCH 发送“簇头移交”消息。

#### 2.1.2 VCH 的选择

MCH 收到节点的加入信息后,选择对应边上信息素浓度大且剩余能量大于给定阈值的成员节点作为本轮的 VCH,并向该节点发送 VCH 确定信息,收到此信息的节点即为本轮 VCH。在数据传输时间内,VCH 暂时休眠直到开始转发 MCH 的数据给 BS,以节省能量。

可见,由于 MCH 只把数据发送给簇内最近的 VCH,而大量的通信耗费交给 VCH 执行,因此避免了 MCH 既要耗费能量接收簇内其他节点的数据并融合,又要传送给较远的 BS 所带来的双重大能耗,从而节省了 MCH 的能量。

### 2.2 ACDCHA 算法<sup>[7]</sup>

由网络随机选择主簇头之后,进入第1轮,如图3所示。

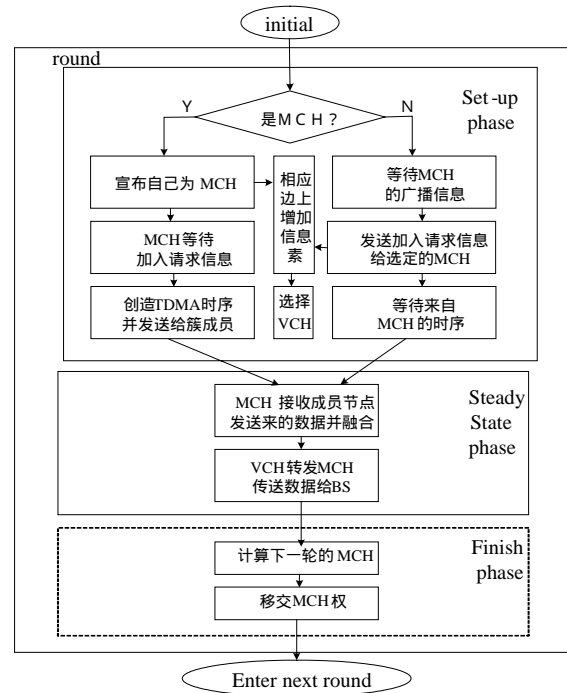


图3 ACDCHA 算法流程

#### 2.2.1 建立阶段

MCH 宣布自己为主簇头,向其他节点广播自己为 MCH 的消息,并在相应边上增加一定量的信息素。每个节点收到消息,根据相应边上的信息素浓度决定加入哪一个簇,并向该 MCH 发送加入信息,同时更新对应边上的信息素。MCH 再根据边上信息素浓度选择 VCH,同时创造时分复用(TDMA)时序并发送给簇成员。

#### 2.2.2 数据传输阶段

在数据传输阶段,簇内的所有节点按照 TDMA 时序向 MCH 发送数据。MCH 将数据融合之后把结果发给 VCH,再由 VCH 转发给基站。在持续工作一段时间之后,网络进入本

轮完成阶段。

### 2.2.3 轮完成阶段

在轮完成阶段,本轮 MCH 计算新一轮新的主簇头,并向新的主簇头发送“主簇头移交”信息。至此,本轮结束,进入下一轮的建立阶段。

## 3 算法分析

距离为  $d$  的节点间传输  $l$  bit 数据所需的能耗<sup>[4]</sup>为:

接收能耗:  $E_{Rx} = lE_{elec}$

传输能耗:  $E_{Tx}(l, d) = E_{Tx-elec}(l) + E_{Tx-amp}(l, d) =$

$$\begin{cases} lE_{elec} + l\varepsilon_{fs}d^2 & d < d_0 \\ lE_{elec} + l\varepsilon_{mp}d^4 & d \geq d_0 \end{cases}$$

其中,  $E_{elec}$  为电子能量;  $\varepsilon_{fs}, \varepsilon_{mp}$  分别是不同距离的放大能量。

如图 1 所示,假设传输  $l$  bit 数据,为简化算法,本文仅考虑节省主簇头的能耗,即仅需满足式(4)。

$$E_{Tx}(l, d_{mb}) > E_{Tx}(l, d_{mv}) \Leftrightarrow d_{mb} > d_{mv} \quad (4)$$

本文假设 BS 远离传感器节点区域,式(4)显然成立。另外,算法利用包含距离、剩余能量等意义的信息素传递节点的信息。在确定副簇头时,选择信息素浓度最大的簇内成员。由信息素的计算公式可知,由距离 MCH 较近且剩余能量较多的节点担任 VCH,一方面减少了 MCH 的能耗,另一方面不至于 VCH 能量枯竭。同样,确定新一轮主簇头时,也衡量了综合因素。所以理论上算法能够平衡网络中节点的能量消耗,达到延长网络生命周期的目的。

## 4 实验及结果

本文在 NS-2 网络仿真平台上,用 100 个传感节点随机分布在  $100 \text{ m} \times 100 \text{ m}$  的正方形区域内,分别对 LEACH、LEACH-C 及本文算法 ACDCHA 进行仿真。节点的初始能量为 2 J,簇头数为 5<sup>[4]</sup>。基站(BS)位于坐标(50, 175)处。所有节点均为静态的,仿真时间 1 000 s。仿真中的传输能量模型采用无线通信模型,仿真结果如图 4 所示。由图 4 可见本文算法节点存活情况明显优于 LEACH 和 LEACH-C。仿真中,ACDCHA 的第 1 个节点死亡出现在 270 s,第 2 个死亡节点出现在 370 s;而 LEACH 分别是 130 s 和 140 s, LEACH-C 则分别出现在 110 s 和 130 s。从这个意义上,本文算法显著延长了网络的生命周期。

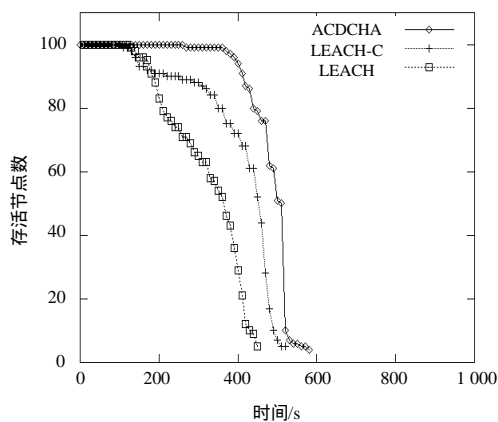


图 4 存活节点数随时间的变化

由图 5 可见,仿真时间 200 s 前,ACDCHA 与 LEACH-C 接收到的数据量基本相当,200 s 后 ACDCHA 的数据量少于 LEACH-C 但远多于 LEACH。分析其原因,可能是由于

ACDCHA 采用了双簇头的数据转发机制,降低了单位时间内的数据传输量。

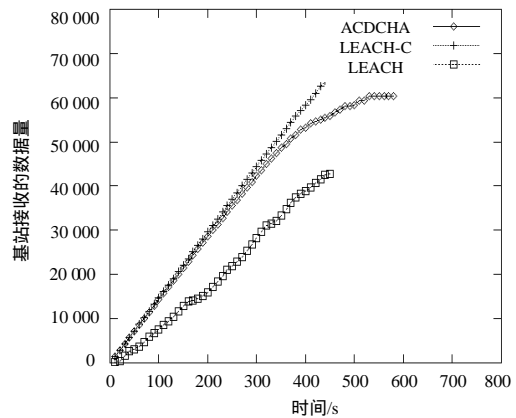


图 5 BS 接收数据量随时间的变化

由图 6 可见 ACDCHA 的网络能量消耗速度慢于 LEACH 和 LEACH-C。

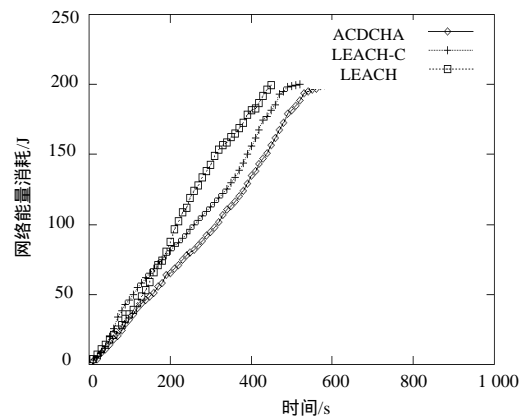


图 6 网络能量消耗随时间的变化

图 7 展示了基站接收的数据量与能量的关系,它表示每消耗 1 J 能量 BS 接收到的数据量。从图 7 可看出,在 130 J 前,ACDCHA 明显优于其他两种算法,130 J 后略逊于 LEACH-C。

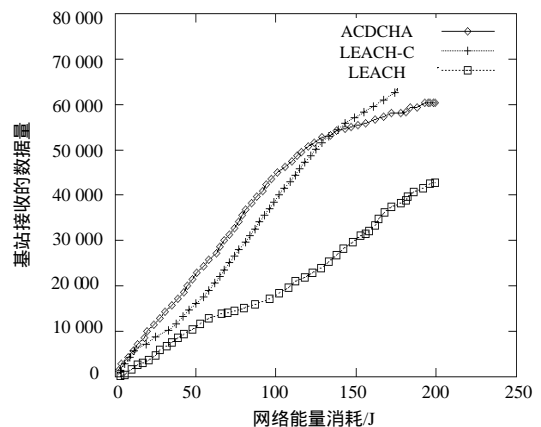


图 7 能量消耗与 BS 接收数据量的关系

综上所述,本文算法各项指标均优于 LEACH;虽然在数据量方面略逊于 LEACH-C,但能量消耗速度和节点存活情况均表现优异。

(下转第 192 页)