

基于节点优先权和兴趣数据筛选的传感器网络分簇算法

李 晟¹ 刘林峰¹ 陈 行²

(南京邮电大学计算机学院 南京 210023)¹ (南京工程学院计算机工程学院 南京 211167)²

摘 要 无线传感器节点工作时仅由自身携带的电池供电,因此高效控制节点能耗可以延长整个网络生命周期,提高网络整体利用率。针对环境监测类场景,提出了一种基于节点优先权和兴趣数据筛选的分簇算法,该算法同时考虑了节点分布距离和剩余能量,将这两个属性换算成簇头选举的优先权权值。此外,在簇头确定后簇头通过相应的数据筛选方法对每一轮子节点发来的数据进行处理,有选择地将数据传到基站。仿真结果表明,该算法在能量控制方面有良好的表现,明显地延长了网络生命周期。

关键词 优先权,数据筛选,平均距离,剩余能量,网络生命周期

中图分类号 TP393 文献标识码 A

Clustering Algorithm of Sensor Network Based on Node Priority and Interest Data Screening

LI Sheng¹ LIU Lin-feng¹ CHEN Hang²

(School of Computer Science & Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China)¹

(Department of Computer Engineering, Nanjing Institute of Technology, Nanjing 211167, China)²

Abstract The working wireless sensor node is powered by its built-in battery. Therefore the effective control of energy consumption can greatly prolong the network life cycle and improve its utilization rate. In this paper, a clustering algorithm based on node priority and interesting data screening was proposed, which can be applied into environmental monitoring, etc. The node priority for being cluster head is calculated by the combination of the distance distribution of nodes and the remaining energy. Subsequently, the chosen cluster head will adopt the corresponding data screening method to handle the data obtained from the child nodes in each round. And the screened data will be sent to base station. The simulation results show that the proposed algorithm achieves better performance in the energy control and network life than others.

Keywords Priority weights, Data screening, Mean distance, Remaining energy, Network life cycle

1 引言

无线传感器网络(Wireless Sensor Network, WSN)一直以来都是国内外研究的热门领域。它的工作原理是通过已经部署的无线传感器节点搜集实时信息,通过无线信号将数据传送给远端的 sink 节点(基站)进行分析^[1]。由于无线传感器节点只具备简单的计算以及信息收发能力,通过有限的电池供电,因此节点的拓扑分布会直接影响无线传感网搜集信息的效率以及网络的生存周期。传感器节点通常距离基站较远,所以节点与基站的通信能耗远远大于节点间通信能耗,如果所有节点均与基站直接通信,就会造成节点整体能耗过快,大大缩短传感网的生命周期。分簇算法在节点中选取簇头,普通节点通过簇头间接与基站进行信息交互,再通过合理的簇头轮换机制来均衡节点的能耗,从而很好地延长整个网络的生存周期。无线传感器在运用于环境监测时,例如监测目标水体的 PH 值,判断该目标水体的 PH 值是否处于正常范

围,通常会规定一个明确的正常指标范围作为衡量的标准,基站再把传感器节点搜集的数据与监测水体的正常 PH 值标准进行比较。在此种场景下,如果监测的目标水体 PH 值在相当长的一段时间内都处于正常水平,簇头节点在传输每一轮数据时把所有的数据都给基站是没有必要的,并且会造成节点能量的浪费。因此,当无线传感运用于此类场景下,簇头节点有选择地向基站传输数据是很有意义的。

由于无线传感器网络中的节点计算能力十分有限,因此通过节点内部自行分簇,普通节点通过簇头节点与基站间接通信的分簇模型也成为如今拓扑算法研究中的热点。LEACH 算法^[2]可以看做是分簇算法领域的先驱,确立了从传感器节点中选取簇头,簇头节点再与邻近节点连接成簇,再由簇头与基站通信建立分簇拓扑模型。它通过控制区域内节点成为簇头的概率来平衡节点的能量消耗。该算法没有考虑节点间距离对节点通信能耗的影响,也没有考虑节点的剩余能量,这会大大缩短网络的生存周期。文献^[3]提出了一种基

本文受国家自然科学基金(61373139, 71301081),江苏省自然科学基金(BK2012833),中国博士后科学基金(2014M560379),南京工程学院引进人才科研启动基金(272340726101101)资助。

李 晟(1991—),男,硕士生,主要研究方向为无线传感器网络拓扑分布;刘林峰(1983—),男,副教授,硕士生导师,主要研究方向为无线传感器网络、机会网络。

于 LEACH 算法的改进算法,该算法将节点的剩余能量作为竞选簇头的依据,并且把平均最小可达能量作为节点选择簇头时的评判条件。文献[4]提出的 GAF 算法也是很具代表性的分簇算法,它提出了将节点所在区域划分成不重叠的网格的思想,但是该算法没有很好地做到节点的能耗均衡。文献[5]中由 P. Neamatollahi 等人提出的 DESC 算法在选取簇头时考虑了节点剩余能量,将节点中心密度作为非簇头节点选择簇头的依据,该算法形成的簇内,节点间距离较小,有效地减少了簇头节点与其子节点的通信能耗。

本文在簇头选取阶段综合考虑了节点间平均距离和节点剩余能量,通过相应算法将它们转化为节点竞选簇头的优先权;在数据传输阶段,对簇头即将发送的数据进行合理筛选,最终提出了一种新的基于簇头优先权和兴趣数据筛选的分簇算法 CABPS(Clustering Algorithm Based on Node Priority and Interest Data Screening)。本文第 2 节介绍算法的问题模型和能量模型,第 3 节详细介绍基于节点优先权和兴趣数据筛选的分簇算法并对其复杂度进行分析;第 4 节介绍仿真环境和仿真结果;最后对全文进行总结。

2 算法模型

2.1 问题模型

本文算法适用于大部分环境监测类场景,为了方便说明算法的流程,文中均以目标水域 PH 值监测为实例。这里将目标水域抽象定义为一个底部为正方形的立方体,根据节点深度将整个水域的节点划分成若干层。由于实际情况中水质的 PH 值在不同深度是不同的,但是处于同一深度水域的 PH 值的标准是相同的,因此对于水质 PH 值监测,该立方体中的每一层对于 PH 值的标准都不相同,每一层水域都可以看作是一个正方形平面区域来单独考虑,如图 1 所示。

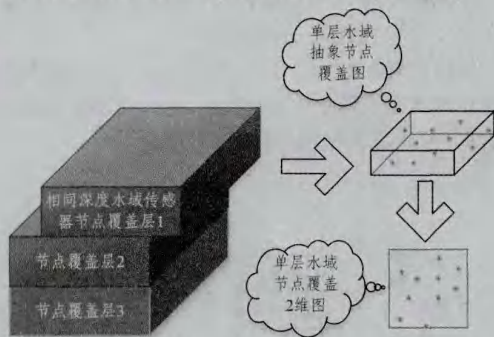


图 1 相同深度节点覆盖平面示意图

为了简化整个问题的描述,对算法所研究的无线传感器网络模型作如下要求:

(1)整个目标区域(布置传感器的区域)中的传感器节点是均匀分布的,且布置完毕后节点的位置不会发生改变,即节点的坐标值是固定的。

(2)在传感器网络的整个生命周期中,所有节点的初始能量相同,且所有节点剩余能量可以感知。

(3)传感网中的所有传感器节点都能够被单独识别。

2.2 能量模型

本文中节点信号传输时的能耗采用文献[2]中运用的能

量模型,发送端和接收端的能耗分别为:

$$E_{tx}(l, d) = E_{dec}(l) + E_{tx-amp}(l, d) = \begin{cases} lE_{dec} + l\epsilon_{fs}d^2, & d < d_0 \\ lE_{dec} + l\epsilon_{mp}d^4, & d \geq d_0 \end{cases} \quad (1)$$

$$E_{rx}(l) = lE_{dec} \quad (2)$$

其中, E_{dec} 表示节点每传输单位 bit 数据的能量消耗, ϵ_{fs} 和 ϵ_{mp} 都表示节点在单位面积 (m^2) 上传输单位 bit 数据的能耗, d_0 表示节点间距离的临界值,节点间距离大于或小于 d_0 时采用的信号传输模型不同。

3 算法描述

本文的分簇算法分为两个阶段:簇头选取阶段和数据传输阶段。在簇头选取阶段,节点会根据距离和能量信息计算出竞选簇头的优先权,优先权最大的将被选为簇头。在数据传输阶段,簇头节点将会对其子节点发回的数据进行合理筛选,减少最终传回基站的数据量,最终达到减少节点能耗的目的。

3.1 簇头选取阶段

上一节提到目标区域被抽象为一个正方形平面区域,其中的节点是均匀分布的。簇头的选取具体步骤如下:

Step 1 将无线传感器节点所覆盖的目标区域划分为若干完全相同的方格,每个方格之间是紧密排列没有间隔的。方格的数目等于整个目标区域即将选出的簇头数目。每个方格内有且仅有一个簇头。以每一块方格区域对角线交点的坐标作为每个方格区域的编号。每个区域内的节点发送的信息都会包含区域编号信息,不同区域中的节点是不可以相互连接的,这样做的目的是便于同一区域内的传感器节点相互识别,同时也避免了不同区域节点之间的相互干扰。最终整个网络中的簇头节点与跟它相连的节点都会属于同一个区域。由于每个方格内节点单独成簇,互不干扰,因此以下步骤均为单个方格中的选取情形,每个方格选取步骤相同。

Step 2 单个方格中的各节点根据距离信息计算出自身平均能耗标准 $P_{average}$, 计算方法如下。

Step 2.1 区域内所有节点(假设为 N 个)首先向本区域的其他节点广播自己的位置信息,这样其中的任意一个节点都可以知道自己与同区域其他节点之间的距离,记为: $d_1, d_2, d_3, \dots, d_{N-1}$ 。

Step 2.2 设该节点到同区域其他节点的平均距离为 $d_{average}$, 根据上一个步骤中的数据可以计算出平均距离

$$d_{average}, \text{计算公式为: } d_{average} = \frac{\sum_{i=1}^{N-1} d_i}{N-1}.$$

Step 2.3 定义第 i 个节点的平均距离为 $d_{average-i}$, 由于前面已经说明在无线传感器网络的整个生命周期中,节点的位置不会改变,因此每个节点的平均距离都是不会变化的。将 $d_{average-i}$ 作为上一节提到的能量模型中的节点间距离 d 并代入公式,可以计算得出当节点 i 在被选为簇头节点后下一轮的网络平均能耗标准 $P_{average-i}$, 由于每个网格中节点之间的距离不可能超过临界值 d_0 , 因此计算公式为:

$$P_{average-i} = l\epsilon_{fs}d_{average-i}^2 \quad (3)$$

Step 3 定义第 i 个传感器节点的剩余能量为 p_i , 第 i 个

节点成为簇头的优先权值为 $F_{weight-i}$, 计算公式如下:

$$F_{weight-i} = p_i - P_{average-i} \times (N-1) \quad (4)$$

其中, $F_{weight-i}$ 代表的是假定节点 i 在当选为簇头之后整个网络能耗的水平, 计算出每个节点的 F_{weight} 之后, F_{weight} 值最大的节点将会当选为簇头, 该区域其它节点将会与它连接形成一个簇。通过这个算法, 在每一轮选取的簇头节点将会是所有节点中担任该轮簇头完毕后剩余能量最多的节点。

3.2 簇头确定后结合数据筛选的数据传输阶段

簇头在整个簇中的作用是将子节点传来的信息转发给远端的基站, 在簇头已经选出以后, 同一方格内的非簇头节点将全部作为该方格内簇头节点的子节点并与其连接。数据传输过程中本文根据实际情况引入了兴趣数据筛选的具体过程:

第1阶段 确定节点编号及对应的矩阵

任意取一个方格内的所有节点单独进行分析, 在簇头选取阶段, 首先要确定节点编号, 每个节点的平均距离已经计算得知, 找出平均距离最小和最大的两个节点, 将平均距离最小的节点编号为 1, 将距离它最近的节点编号为 2, 此后运用贪心算法依次对方格内所有节点编号。

确定节点对应的矩阵时, 以平均距离最小的节点作为圆心、以这两个节点的节点距离为半径作一个圆, 随后作出该圆的任意一个外切正方形。然后对这个正方形进行均匀的网格划分, 保证每个节点都能够在一个小的网格里, 完成以后包含网格的正方形区域可以转换成一个 $r * r$ 的矩阵, r 是根据节点的数目以及节点的密集程度来确定的。令这个矩阵为 A , i 为行向量, j 为列向量, 那么方格中的所有节点在矩阵中的位置均可以用 $A[i][j]$ 的形式表示出来, 不包含节点的网格在矩阵中对应位置的数值定位 0。

第2阶段 对搜集的数据进行合理划分

为了直观地说明数据筛选的过程, 文中将簇头节点搜集的数据分为 3 类:

(1) 正常数据。正常数据表示该节点监测的水域的 PH 值处于正常指标范围内并且与上次传输的结果相比无太大变化。

(2) 数值超标数据。数值超标数据表示该节点监测水域

PH 值超出了已定的标准范围。(3) 变化超标数据。变化超标数据表示此次节点搜集的数据与上一次传输的数据相比, 变化程度超过了某一个正常的范围, 这个变化程度的范围事先已经规定。

节点本身在搜集到数据后可以判断搜集的数据属于哪一类, 在向簇头传递的数据中会有一个关于区别传递数据种类的标志位, 上述 3 种数据在标志位中分别定位为 1、2、3。下文将第二种和第三种数据统称为关键数据。

第3阶段: 数据筛选

以上 3 种数据中, 数据 2 和数据 3 是需要全部传输的。数据 1 全部传输会造成节点能量无端消耗, 全部筛选出去又会造成基站无法全面了解水质实时信息。为了解决该问题, 本文提出如下的数据筛选方案¹⁾:

初始轮次筛选方案: 将初始矩阵存储于所有节点中, 当某个节点被选为簇头后, 此轮数据传输过程结束前, 该簇头节点对应矩阵中的数值都置为 0。那么在第一轮除簇头节点外的所有节点对应的矩阵数值为 1, 矩阵其它数值都为 0。在这个第一轮的数据传输中, 簇头节点将所有的数据都传输到基站供相关人士分析。

非初始轮次筛选方案: 在第二轮采集数据后, 会有两种情况, 第一种是所有节点传回的依然是标志位都是 1 的数据, 簇头遍历新的矩阵会发现其与上一轮的矩阵相比没有变化, 在这种情况下簇头节点依旧是按照编号发送第一个节点的信息, 但是此后每隔 x 个节点再发送一次节点的数据(被选为簇头节点的编号此轮会被跳过), x 的值是根据水域特点和节点的疏密程度来确定的。

第二种情况就是某些节点传回的数据中包含了标志位为 2 或 3 的数据, 那么此时传回的数据由非初始轮次筛选方案中第一种情况理论下应当选出的数据、所有标志位是 1 的数据及该节点编号前后两个编号对应的节点的数据、标志位是 2 的数据这 3 部分数据的并集组成。

数据轮次更新的方案选择: 此后每新一轮数据中如果所有标志位再次出现都是 1 的情况时, 采用初始轮次筛选方案筛选数据, 否则采用非初始轮次筛选方案。

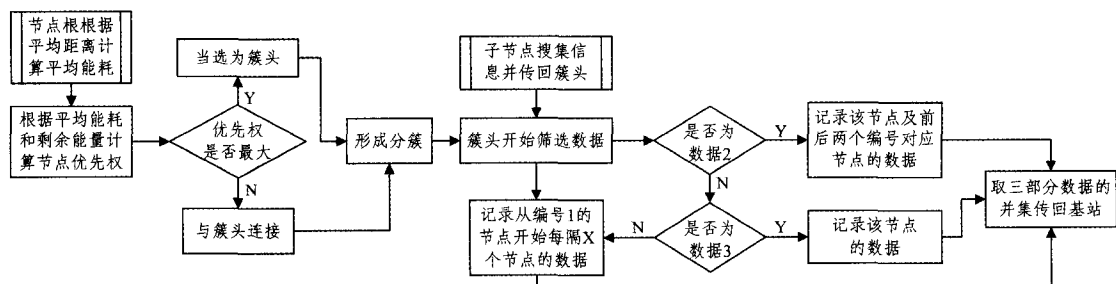


图2 算法流程

3.3 算法复杂度及期望分析

本文簇头选择算法首先要计算 n 个节点的平均距离, 之后又要比较计算 n 个节点成为簇头的优先权, 最终计算了 $2n$ 次, 因此算法复杂度为 $O(n)$ 。选取簇头时节点报文交互的条数为 $n^2 - 1$, 确定簇头节点后的报文交互条数为 $n - 1$, 所以最终交互的报文数为 $n^2 - n - 2$, 本文算法的报文复杂度为 $O(n^2)$ 。

设定整个区域的无线传感器节点个数为 N , 每个簇中包含簇头在内的节点数目为 t , 簇头数为 c , 无法预知 t 能否被 N 整除, 所以本文算法期望的簇头数目为:

$$c = \frac{N}{t} + \begin{cases} 0, & \frac{N}{t} \text{ 余数为 } 0 \\ 1, & \frac{N}{t} \text{ 余数不为 } 0 \end{cases} \quad (5)$$

¹⁾ 为了便于说明, 假定第一轮所有节点搜集的都是正常数据。

设定簇头向基站发送一条数据的能耗为 Q_c , 接受一条数据的能耗为 R_c , 子节点向簇头发送一条数据的能耗为 Q_s , 每个簇在一轮数据传输中的整体能耗为 Q_m , 每一轮簇头筛选除去的数据条数为 m , 那么根据上面簇头个数的期望值可以推算出每个簇在一轮数据传输中的整体能耗期望值为:

$$Q_m = (t-1) * Q_c + (t-1-m) * Q_s + (t-1) * R_c \quad (6)$$

4 仿真实验

4.1 仿真环境及参数设置

本文算法采用 C 语言进行仿真, 仿真平台为 Visual Studio 2013。仿真时设定 N 个无线传感器节点均匀分布在 $a \times a$ (m^2) 的正方形目标区域中。基站位于整个目标区域的中心, 坐标为 (b, b) 。为了简化实验的模型, 假定基站的通信范围能够完全覆盖整个目标区域。目标区域中除了基站以外的所有普通节点的初始能量都为 E_0 , 能量消耗均按照式(1)和式(2)所列出的模型计算。设定每一次发送的数据包长度为 l bit, 并且每一轮进行 k 次数据传输。模型中的参数设置如表 1 所列。

表 1 实验主要参数

参数	描述	取值
E_0	节点初始能量	0.5J
E_{elec}	节点传输单位 bit 能耗	50nJ/bit
ϵ_{fs}	节点单位面积上传输单位 bit 能耗	10pJ/(bit · m ²)
d_0	节点间距离临界值	85m
N	传感器数目	1000
a	仿真目标区域边长	100m
b	基站坐标	50
l	每次发送的数据包长度	1000bit
k	每轮数据传输次数	10
x	数据筛选间隔参数	1

一个分簇中节点数量太少会造成簇头数量过多从而加快网络整体能量的消耗, 分簇中节点过多又会加快簇头的能耗导致节点生命周期缩短, 所以实验时将分簇中节点个数设定为 7~15 个, 结合算法步骤 1 中的网格划分, 对本文算法采用不同节点个数最终运行的轮数进行分析, 如图 3 所示。

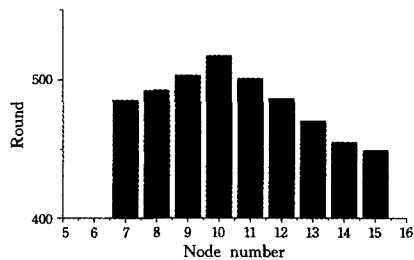


图 3 不同节点数形成簇对网络生命周期的影响

当每个簇中节点个数为 10 时, 传感网运行的轮数最多, 所以将每个簇的节点总数定为 10, 因此目标区域被划分成 100 个边长为 10m 的正方形区域, 每个区域有 10 个传感器节点, 其中每一轮将会有有一个节点将被选为该方格区域中的簇头。

4.2 仿真性能标准

为了体现本文所提的算法与其他算法在性能上的区别, 实验分别从下面两个角度进行比较。

(1) 网络生命周期

实验为了对比本文算法与其他算法对网络生命周期的影响, 对网络生命周期的定义如下:

从网络部署之后第一轮开始, 直到整个无线传感器网络中首次出现节点能量耗尽, 在这期间整个网络正常运行的轮数作为网络的生命周期。

(2) 网络整体能量

网络整体能量指当前轮次下整个网络中所有节点的剩余能量总和, 通过对网络每一轮整体能量的比较, 可以反映该算法均衡整个网络能耗的程度。

4.3 仿真结果分析

本文实验的具体内容包含 3 个部分: 实验一是通过设置 3 种数据在全部数据中所占的比例, 然后分别比较簇头节点使用数据筛选前后向基站发送的数据量。实验二是通过设置 3 种数据在全部数据中所占的比例, 分别比较在这 3 种比例下本文簇头选择算法采用数据筛选前后的网络生命周期。实验三是比较本文算法、LEACH 算法和 DESC 算法在相同实验环境下同一时期的网络节点总体的剩余能量和网络生命周期。

4.3.1 实验一

已知每个簇中包含簇头在内的节点数为 10。数值超标数据出现时, 需要将它本身以及它前后编号对应的两个节点的信息一同传回基站, 因此, 令数值超标数据所占比例为 P_c , 变化超标数据所占比例为 P_d , 由于两种关键数据的比例之和超过 80%, 经上述筛选方法筛选数据后, 簇头依旧需要将所有数据全部传回基站, 为了保证至少能够成功筛选除去一条信息, 当满足 $10 * P_c + 10 * P_d + 1 \leq 9$ 时, 数据筛选才会生效。

图 4、图 5 分别为 P_c 、 P_d 取 20% 和 10% 的情况下, 簇头节点筛选数据前后基站接收数据量的对比, 从两幅图可以看出, 数据经过簇头的筛选后, 在保证关键数据不遗漏的前提下, 基站接收到的数据比普通情况要少, 而且随着关键数据所占百分比的减少, 筛选数据的结果更为明显。当两种数据分别取 20% 和 10% 时, 簇头筛选后的数据发送量将大致分别减少 12.2% 和 22.6%。这在一定程度上减少了簇头节点的能耗。簇头发送的数据量在每一轮数据传输时出现波动的原因是由于两种关键数据与按编号默认传输数据的节点重合率不同造成的, 节点重合率越高, 筛选后发送的数据量就越少。

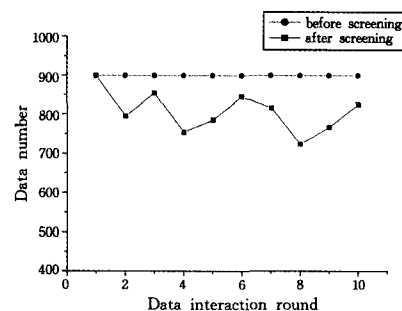


图 4 P_c 、 P_d 均为 20% 的数据接收量

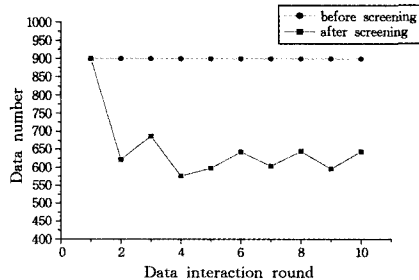


图5 P_c, P_d 均为 10% 的数据接收量

图6中曲线展示的是当所有数据都是正常数据时,簇头筛选数据后基站接收到的数据量。可以看出当簇头接收到的数据在一段时间内都是正常数据时,数据筛选的作用将发挥到最大,此时根据数据筛选间隔参数 x 的不同,筛选后的数据量也会不同,本文取 $x=1$,由计算可知在第一轮数据传输结束后,接下来的每一轮簇头筛选后传输的数据都将是正常传输模式下的 50%,这大大减少了簇头的能耗。

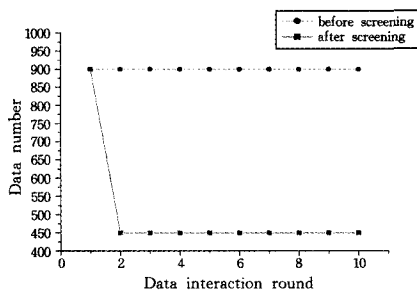


图6 P_c, P_d 均为 0% 的数据接收量

4.3.2 实验二

本实验通过仿真模拟出了当 P_c, P_d 分别取 0%、10% 和 20% 的情况下,本文簇头选择算法在选取簇头后簇头节点筛选数据前后的生命周期。从图7可以看出,当簇头节点不筛选数据直接将收到的数据发给基站时,整个网络的生命周期为 517 轮,当簇头节点筛选数据后再发送给基站,随着 P_c, P_d 取值的减小,生命周期分别增长为 553 轮、588 轮和 707 轮。这组数据说明筛选数据能在一定程度上降低每一轮簇头节点的能耗,并且随着 P_c, P_d 数值的减小,筛选数据后的网络的生命周期会逐渐变长。

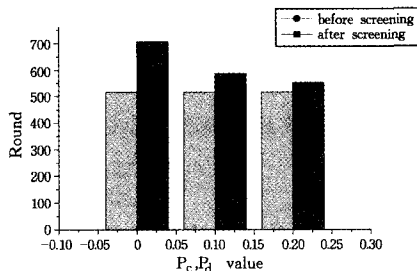


图7 不同数据比例下本文算法筛选数据前后的网络生命周期对比

4.3.3 实验三

本实验对比不同 P_c, P_d 取值时 3 种算法的生命周期,如图9所示。可以看出,在 3 种 P_c, P_d 取值情况下,执行 CABPS 算法下的网络生命周期均高于 LEACH 和 DESC。并且行 LEACH 和 DESC 算法下的生命周期并没有随着不同种类

数据量的变化而变化,CABPS 算法随着正常信息所占比重的增加,延长网络生命周期的时间更长。

当两种关键数据的比例不满足数据筛选发挥作用的条件 ($P_c + P_d > 80\%$) 时,每一轮簇头节点都需要将所有的数据传回基站,仿真记录了 LEACH 算法生命周期截止之前,CABPS 算法、LEACH 算法以及 DESC 算法每一轮的网络整体剩余能量的数据,如图8所示。从图中可以看出 CABPS 算法在生命周期内的总体节点能量均高于同时期的 LEACH 算法和 DESC 算法,这说明 CABPS 算法在控制节点能耗方面表现不错。另外,分析剩余能量曲线的变化趋势还可以看出采用 LEACH 算法下的总体剩余能量在生命周期内的变化呈由急到缓的曲线,DESC 虽然变化幅度小,但是变化趋势依旧明显,而 CABPS 算法下的总体节点剩余能量在生命周期内的变化非常平缓,接近 1 次线性变化,这说明 CABPS 算法在总体能耗均衡方面比 LEACH 和 DESC 更优越。结合实验二的结果综合分析得知:当两种关键数据的比例满足数据筛选有意义的条件时,随着两种数据比例的降低,CABPS 算法的性能将会继续提高,它比另外两种算法具有更好的自适应性。

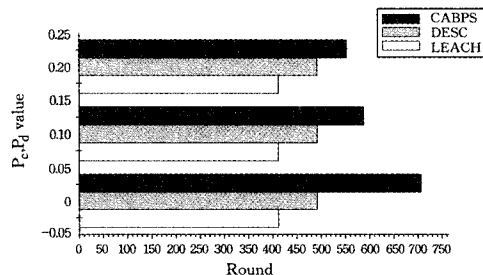


图8 不同 P_c, P_d 取值下 3 种算法的生命周期对比

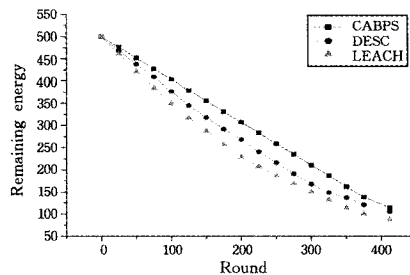


图9 网络整体剩余能量变化

结束语 无线传感器节点能量有限,合理地控制节点能耗才能延长网络生命周期。本文主要针对对现有分簇算法的不足,提出了一个综合考虑节点平均距离和剩余能量并且在数据传输阶段实行数据筛选的分簇算法 CABPS。该算法结合了每个节点的平均距离、剩余能量、信号传输能耗这 3 个因素,通过公式转换成每个节点自身竞选簇头的优先权值,根据权值的大小来合理地决定簇头。并且在数据传输阶段,簇头会对子节点发来的数据进行合理筛选,在保证关键数据不遗漏的前提下将更精简的信息传回基站。通过仿真验证,CABPS 算法相较于 LEACH 算法和 DESC 算法,一定程度上延长了网络的生命周期。同时,在网络节点整体能耗控制方面,CABPS 也表现出良好的性能。

参考文献

[1] 孙利民,李建中,陈渝.无线传感器网络[M].北京:清华大学出

- [2] Heinzelman W R, Chandrakasan A, Balakrishnan H. Energy efficient communication protocol for wireless microsensor networks[C]//Proc of the 33 rd Annual International Conference System Sciences. 2000(2):1-10
- [3] Younis O, Fahmy S. HEED: a hybrid, energy efficient, distributed clustering approach for ad hoc sensor networks[J]. IEEE Transactions on Mobile Computing, 2004, 3(4): 366-379
- [4] Xu Y, Heidemann J, Estrin D. Geography-informed energy conservation for ad hoc routing[C]//Proc of 7th Annual Int. Conf on Mobile Computing and Networking. Rome, Italy: ACM Press, 2001: 70-84
- [5] Neamatollahi P, Taheri H, Naghibzadeh M. DESC: distributed energy efficient scheme to cluster wireless sensor networks[C]//Proceedings of the 9th IFIP TC 6 International Conference. Vilanova Geltru on Wired/Wireless Internet Communications. , Spain, 2011: 234-246
- [6] Xuefei T, Xinzheng N, Shahzad A. Research on energy aware-topology strategy based on wireless sensor in internet of things [J]. International Journal of Computational Intelligence Systems, 2014, 7(6): 1137-1147
- [7] Dabirmoghaddam A, Ghaderi M, Williamson G. On the optimal randomized clustering in distributed sensor networks[J]. Computer Networks, 2014, 59: 17-32
- [8] Zeghlache D, Jouaber B, Slama I. Topology control and routing in large scale wireless sensor networks[J]. Wireless Sensor Network, 2010, 2(8): 584-598
- [9] Younis O, Fahmy S. HEED: a hybrid energy efficient, distributed clustering approach for adhoc sensor networks [J]. IEEE Transactions on Mobile Computing, 2004, 3(4): 366-379
- [10] Javad Akbari T. An energy-efficient topology construction algorithm for wireless sensor networks[J]. Computer Networks, 2013, 57(7): 1714-1725

(上接第 226 页)

- [25] Huang D, Chung H U, et al. A Cluster-Tree-Based GTS Allocation Scheme for IEEE 802. 15. 4 MAC Layer[C]//2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing(IMIS). IEEE, 2012: 524-528
- [26] Han S, Lee Y, et al. GTS Allocation Scheme of IEEE 802. 15. 4 for Energy-efficiency[C]//International Symposium on Computer Network and Multimedia Technology, 2009 (CNMT 2009). IEEE, 2009: 1-4
- [27] Lee H, Lee K, Shin Y. A GTS allocation scheme for emergency data transmission in cluster-tree WSNs[C]//14th International Conference on Advanced Communication Technology(ICAICT). 2012: 675-678
- [28] Jurcik P, Koubaa A. IEEE 802. 15. 4 OPNET simulation model [OL]. <http://www.open-zb.net>
- [29] Xia Feng, Hao Ruo-nan, Cao Yang, et al. ART-GAS: An Adaptive and Real-Time GTS Allocation Scheme for IEEE 802. 15. 4 [C]//AINTEC'11. New York: ACM, November 2011: 96-103
- [30] Zhang H, Xin S, et al. An Adaptive GTS Allocation Mechanism in IEEE 802. 15. 4 for Various Rate Applications[C]//Fourth International Conference on Communications and Networking in China, 2009(ChinaCOM 2009). IEEE, 2009: 1-5
- [31] Ho C, Lin C, et al. Dynamic GTS Allocation Scheme in IEEE 802. 15. 4 by Multi-Factor[C]//2012 Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing(IIH-MSP). IEEE, 2012: 457-460
- [32] Buttazzo G C. RateMonotonicvs. EDF: JudgmentDay [J]. Real-Time Systems, 2005, 29: 5-26
- [33] Hong Y, Center S R, et al. Adaptive GTS Allocation scheme to support QoS and multiple devices in 802. 15. 4[C]//11th International Conference on Advanced Communication Technology, 2009(ICAICT 2009). 2009, 3: 1697-1702
- [34] Rasyid M U H A, Lee B-H, Sudarsono A. PEGAS: Partitioned GTS Allocation Scheme for IEEE 802. 15. 4 Networks[C]//2013 International Conference on Computer, Control, Informatics and Its Applications(IC3INA). Nov. 2013: 29-32
- [35] Huang Y, Pang A, Hung H. An Adaptive GTS Allocation Scheme for IEEE 802. 15. 4[J]. IEEE Transactions on Parallel and Distributed Systems, 2008, 19(5): 641-651
- [36] Cho H W, Bae S J, et al. Utilization-Aware Dynamic GTS Allocation Scheme in IEEE 802. 15. 4[C]//2010 16th Asia-Pacific Conference on Communications(APCC). Oct 2010: 210-214
- [37] Huang Y, Pang A, Kuo T. AGA: Adaptive GTS allocation with low latency and fairness considerations for IEEE 802. 15. 4[C]//Proc. of IEEE International Conference on Communications (ICC). volume 9, Istanbul, Turkey, Jun. 2006: 3929-3934
- [38] Semprebom T, Montez C, Zomer G, et al. Guaranteed Time Slot Allocation for Periodic Messages with(m, k)-firm Constraints in IEEE 802. 15. 4 Networks[C]//2012 IEEE 17th Conference on Emerging Technologies & Factory Automation (ETFA). Sept 2012: 1-8
- [39] Koubaa A, Alves M, Tovar E. IEEE 802. 15. 4 for wireless sensor networks; Technical overview: TR-050702[R]. IPP Hurray, July, 2005
- [40] Sharath R, Saksham K, et al. A Survey and Comparison of GTS Allocation and Scheduling Algorithms in IEEE 802. 15. 4 Wireless Sensor Networks[C]//2013 IEEE Conference on Information & Communication Technologies(ICT). IEEE, 2013: 98-103
- [41] 蔡惠娟, 蒋文贤. IEEE 802. 15. 4 多时隙下 GTS 性能分析及配置优化[J]. 计算机应用, 2012, 32(12): 3499-3504
- [42] 张文强. IEEE 802. 15. 4 协议中 GTS 分配机制的研究和改进 [D]. 广州: 暨南大学, 2013
- [43] 孙利民, 李建中, 陈渝, 等. 无线传感器网络[M]. 北京: 清华大学出版社, 2005
- [44] 乔冠华, 毛剑琳, 郭宁, 等. 基于业务区分的 IEEE 802. 15. 4 协议分析及改进[J]. 计算机科学, 2014, 41(10): 149-153