

基于新型聚类的无线传感器网络非均匀分层路由协议

陶志勇¹ 王和章²

(辽宁工程技术大学电子与信息工程学院 辽宁 葫芦岛 125105)¹

(辽宁工程技术大学研究生学院 辽宁 葫芦岛 125105)²

摘 要 针对成簇路由协议中由节点负载不均引起的能量空洞问题,提出了一种基于新型聚类的负载均衡非均匀分层路由协议(NHRPNC)。首先,利用改进的 LEACH 协议阈值函数选举区头,并对网络进行合理的非均匀分区;其次,对每个区头运用新型聚类算法实现区内非均匀分簇;然后,在每个簇内采用四步簇首选择机制来周期性地选择簇首;最后,在簇间多跳通信时,采用动态权重的方式优化多跳路径。仿真结果表明,与低功耗自适应集簇分层(LEACH)协议、分布式能量均衡非均匀成簇(DEBUC)协议以及基于动态分区的无线传感器网络非均匀成簇(UCDP)协议相比,NHRPNC 在网络生命周期方面可分别提高 257.5、33.74 和 12.83 个百分点,且具有良好的能耗均衡性。

关键词 能量空洞,负载均衡,非均匀分簇,四步簇首选择机制,动态权重

中图分类号 TP393 文献标识码 A DOI 10.11896/j.issn.1002-137X.2018.03.019

Non-uniform Hierarchical Routing Protocol Based on New Clustering for Wireless Sensor Network

TAO Zhi-yong¹ WANG He-zhang²

(School of Electrics and Information Engineering, Liaoning Technical University, Huludao, Liaoning 125105, China)¹

(School of Graduate Studies, Liaoning Technical University, Huludao, Liaoning 125105, China)²

Abstract Aiming at the problem of energy hole caused by uneven load energy in clustering routing protocol, a non-uniform hierarchical routing protocol based on new clustering for load balancing(NHRPNC) was proposed. Firstly, the algorithm utilizes the improved threshold function of LEACH protocol to select the region-heads and make the reasonable non-uniform partitions of the network. Secondly, the region-heads use the new clustering algorithm to achieve the non-uniform clustering in each region. Then, the four-step selection mechanism of cluster heads is adopted to select the cluster head in each cluster periodically. At last, for inter-cluster multi-hop communication, the multi-hop path is optimized by dynamic weight. Simulation results reveal that compared with LEACH (Low Energy Adaptive Clustering Hierarchy) protocol, DEBUC (Distributed Energy Balanced Unequal Clustering routing) protocol and UCDP (Uneven Clustering based on Dynamic Partition) protocol, NHRPNC can promote the percentage point of 257.5, 33.74 and 12.83 in the life cycle of the network respectively and has favorable performance in balancing the energy consumption.

Keywords Energy hole, Load balance, Non-uniform clustering, Four-step selection mechanism of cluster heads, Dynamic weight

1 引言

无线传感器网络(Wireless Sensor Network, WSN)是由大量廉价的具有通信、计算和存储能力的无线传感器节点组成的,其随机部署在监测区域内,通过无线通信的方式形成一个多跳的自组织网络系统,其目的是协同地感知、采集和处理网络区域中的信息,并且把相应的信息传送到观察者^[1]。目前,WSN 已应用在许多领域,如军事、环境等。但是,WSN 中的传感器节点由于采用电池供电,能量非常有限,并且在部署

完毕后无法实现二次供电,因此,如何降低网络能耗、延长网络生命周期成为了无线传感器网络路由协议的研究重点^[2]。

为了节约能耗,许多基于成簇的层次型路由协议被提出^[3]。分簇的拓扑结构降低了网络信息的冗余度,优化了数据的传输数量,减少了节点直接传输数据至基站的次数,降低了能耗,延长了网络寿命。LEACH 协议^[4]是最早被提出的一种分簇型路由协议,它能够有效地降低节点能耗和延长网络生命周期;但是,簇首的选举具有随机性,剩余能量相对较低的节点也可能当选为簇首,这会导致节点之间的能耗不均匀。

收稿日期:2017-01-19 返修日期:2017-04-19 本文受国家自然科学基金(61402212, 61240014),辽宁省自然科学基金(2015020100, 20170520098)资助。

陶志勇(1978—),男,博士,副教授,CCF 会员,主要研究方向为多媒体通信,E-mail:82456020@qq.com(通信作者);王和章(1992—),男,硕士生,主要研究方向为无线传感器网络,E-mail:986646540@qq.com。

文献[5]在簇首选举方面为了避免剩余能量较低或邻居节点数较少的节点当选簇首,采用加权概率的方式筛选所有节点,优化了簇首,有效地延长了网络寿命;然而,簇首与基站之间采用单跳通信,节点之间的能耗不均匀。为了解决簇间单跳的问题,文献[6]提出了一种能耗均衡的分簇路由协议——EBCRP,该协议在簇间通过考虑邻近簇首相对自身的距离和方向来确定中继簇首,实现了簇间多跳,均衡了节点之间的能耗;但是,中继簇首的选择并没有考虑其剩余能量,容易导致其能量消耗过快,节点过早死亡。文献[7-8]提出了一种基于非均匀分簇的路由协议,通过控制竞争半径来调整簇的规模,使距离基站较近的簇规模相对较小,这样簇内能耗相对降低,簇首就会拥有更多的能量转发数据;但簇首的选择只由概率和门限值决定,缺乏灵活性,无法保证簇首最优。文献[9]提出的 EBUCP (Energy-Balanced Unequal Clustering Protocol)协议利用 PSO (Particle Swarm Optimization)算法寻找最佳节点担任簇首,使簇首的分布更均匀,同时在簇间采用贪婪算法选择中继节点,均衡了节点间的能耗,延长了网络寿命;但 PSO 算法易陷入局部最优状态。文献[10]提出了一种能量高效均衡的分簇路由协议——DEBUC,该协议采用基于时间的簇头竞争算法,通过控制候选簇头的竞争范围实现网络的非均匀分簇,均衡了能耗,缓解了能量空洞,有效地延长了网络的存活时间。文献[11]提出了一种基于动态分区的非均匀成簇路由协议——UCDP,该协议首先将网络进行动态非均匀分区,使靠近基站的区面积较小;然后在每个区内综合考虑距离因子和剩余能量因子,实现区内非均匀分簇;最后在簇间,簇首和区头共同协作完成路由传输,显著延长了网络的生存周期。然而,重复的分区增加了不必要的控制能耗。

本文在研究 LEACH^[4], DEBUC^[10] 和 UCDP^[11] 协议的基础上,提出了一种基于新型聚类的负载均衡的非均匀分层路由协议。该协议首先在监测区域内选举区头,并对网络进行合理的非均匀分区;其次,每个区头运用新型聚类确定聚类中心和节点所属的簇,实现区内非均匀分簇;然后,在每个簇内采用四步簇首选择机制周期性地选举簇首;最后,在数据转发阶段,通过动态权重的方式优化多跳路径。实验结果表明,与 LEACH, DEBUC 和 UCDP 相比,所提算法能够显著延长网络寿命,均衡节点之间的能耗。

2 LEACH 协议

LEACH 是 Heinzelman 等人为无线传感器网络设计的一种低功耗自适应集簇分层型路由协议。该协议的运行过程以“轮”的概念表示,每轮分为两个阶段:簇建立阶段和稳定数据传输阶段。簇的建立阶段又可划分为 4 个部分:簇首选举、簇首广播、成簇和生成 TDMA (Time Division Multiple Address) 时隙。稳定数据传输阶段分为两部分:簇内通信和簇间通信^[4]。

簇建立阶段,每个传感器节点首先产生一个 0~1 之间的随机数,然后计算该随机数与阈值的的关系,若随机数小于阈值,则该节点当选为簇首。当选为簇首的节点在全网中广播

自己成为簇首的信息,未当选为簇首的节点根据接收信号的强度选择其中最大值所对应的簇首节点加入。簇形成后,簇首根据簇内普通节点的数目创建 TDMA 时隙表,并通告簇内成员节点。

LEACH 算法中阈值函数的计算公式为:

$$T(n) = \begin{cases} 0, & n \notin G \\ \frac{p}{1 - p \cdot (r \bmod (1/p))}, & n \in G \end{cases} \quad (1)$$

其中, p 为簇首节点的比例; G 为最近 $1/p$ 轮中未当选为簇首的所有节点的集合; r 表示网络当前的运行轮数。

稳定的数据传输阶段,簇内成员节点将感知的数据单跳传输至簇首,簇首将接收到的数据进行融合后单跳传输至基站。

3 系统模型

3.1 网络模型

假设网络由 N 个能量相同的传感器节点组成,其随机分布在 $M \times M$ 的监测区域内。对网络模型做如下设定^[10]:

- 1) 基站位于监测区域外部,传感器节点位于监测区域内部,部署后位置均不变;
- 2) 所有节点是同构的,即具有相同的功能,并且都有一个唯一的节点号标识 ID;
- 3) 链路是对称的,如果已知发射端的发射功率,则接收端可以根据接收信号的强度估算出两者之间的近似距离;
- 4) 每个节点能够感知自身的位置信息和当前的剩余能量;
- 5) 能够通过数据融合技术减小冗余度,进而减少数据传输的任务。

3.2 能耗模型

本文采用与文献[12]相同的能量消耗模型,即一阶无线电模型,忽略节点在计算、存储等过程中的能量消耗。当发射端和接收端之间的距离 d 小于 d_0 时,能耗遵循自由空间模型,与 d^2 成正比;否则,能耗遵循多径衰落模型,与 d^4 成正比。节点发送 k bit 的数据至距离为 d 的位置,所消耗的能量由两部分组成,即发射电路损耗 $E_{Tx-elec}(k)$ 和功率放大损耗 $E_{Tx-amp}(k, d)$ 。

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

其中, E_{elec} 为单位比特的发射电路损耗, ϵ_{fs} 和 ϵ_{mp} 为单位比特的数据传输单位距离时放大器的能量损耗, $d_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$ 为距离阈值。

节点每接收 k bit 的数据所消耗的能量为:

$$E_{Rx} = kE_{elec} \quad (3)$$

3.3 数据融合模型

本文算法通过数据融合技术减少所传输数据之间的冗余度,进而减少网络传输的数据量,达到节约网络能耗的目的。由于不同簇之间存在较大的差异,因此在仿真时不考虑簇与

簇之间的数据融合。假设网络的数据融合模型为:簇首接收每个节点发送的 l bit 数据,压缩为 $f n_i l$ bit 数据, f 为簇内融合系数, n_i 为簇内节点数。

4 基于新型聚类的非均匀成簇路由协议

LEACH 算法中,频繁构簇会造成不必要的能量消耗,从而降低能量效率;而基于 LEACH 改进的成簇路由协议在簇间多跳时由于节点负载不均又会导致能量空洞。为了缓解能量空洞,提高能量效率,提出一种基于新型聚类的非均匀成簇路由协议。该协议主要分为 3 个阶段:1)分区阶段,即根据节点的历史能耗因子以及节点到基站的距离选举区头^[11],通过区头在自身竞争半径内广播获胜信息,将网络非均匀分区;2)区内成簇阶段,即在每个区头上利用新型聚类算法选举区内聚类中心,并确定节点所属的簇,实现区内非均匀分簇,在每个簇内采用四步簇首选择机制周期性地选举簇首;3)数据传输阶段,分为簇内通信与簇间通信两部分。簇内通信采用单跳的方式;簇间通信采用单跳和多跳相结合的混合传输方式,当簇首到基站的距离小于一定范围时,簇首直接将数据传输至基站,否则,簇首根据下一跳节点的相对剩余能量因子和传输能耗因子,通过动态权重的方式建立簇间多跳路由,将数据转发至基站。

下面给出一些相关的定义。

定义 1(历史能耗因子) 历史能耗因子 e_cost_i 表示在前 $r-1$ 轮周期内节点 i 消耗的能量与该区内所有节点的平均消耗能量的比值。

定义 2(相对剩余能量因子) 相对剩余能量因子 $W(ECH_k)$ 表示中继簇首的剩余能量和网络中其余簇首的平均剩余能量的关系, $0 \leq W(ECH_k) \leq 1$ 。

定义 3(传输能耗因子) 传输能耗因子 $EN(u)$ 表示簇首节点与中继簇首及中继簇首与基站的位置关系, $0 \leq EN(u) \leq 1$ 。

4.1 非均匀分区

本文算法的目标是通过非均匀分簇来缓解能量空洞,即使得距离基站较近的区域簇规模较小,而距离基站较远的区域簇规模较大。若聚类之前不采用分区,则在聚类算法中选择聚类中心时需要为每个节点设置阈值,而阈值的设置只考虑了节点自身到基站的距离,并没有考虑其余节点的影响,因此可能会出现距离较近的两个节点同时被选举为聚类中心的情况,这样的聚类结果不利于均衡网络的负载,显然是不合理的;但若采用分区,则可以通过综合考虑区内所有节点到基站的等效距离在区内设置唯一阈值,能够有效解决距离较近的两个节点同时被选为聚类中心的问题,有利于均衡网络负载。因此,在聚类之前进行非均匀分区是必要的。

在节点聚类之前,NHRPNC 算法首先对网络进行合理的非均匀分区,有效均衡网络的整体负载。在本文算法中,区头的任务较重,它不但需要在自己的竞争范围内广播获胜消息,而且还需要接收区内节点的加入请求信息(包括节点位置信息),并根据位置信息利用新型聚类算法选举区内聚类中心和确定区内节点所属的簇,因此区头需要具有较高的剩余能量。

本文以节点的历史能耗因子作为选举区头的主要依据。

假设传感器网络被分为 M 个区,设在第 P 次分区后,任意分区 m 的集合 $T(P,m)$ 中有 $N(P,m)$ 个节点。其中,传感器节点 i 在第 r 轮开始时的能量为 E_i^r ,根据定义 1 给出节点 i 的历史能耗因子为:

$$e_cost_i^r = \begin{cases} 0, & r=1 \\ \frac{E_0 - E_i^r}{\frac{1}{N(P,m)} \sum_{k \in T(P,m)} (E_0 - E_k^r)}, & r>1 \end{cases} \quad (4)$$

当 $r=1$ 时,节点 i 没有任何能量消耗, $e_cost_i^r$ 为 0;当 $r>1$ 时, $e_cost_i^r$ 越大,说明该节点在前 $r-1$ 轮工作周期内的相对历史能耗越多,节点剩余能量越少。

结合 Younis 的成簇思想,根据节点的历史能耗因子对 LEACH 算法的阈值函数 $T(n)$ 改进如下:

$$T(n) = \begin{cases} 0, & n \notin G \\ \frac{p}{1 - p(r \bmod (1/p))} \cdot \frac{1}{e_cost_i^r}, & n \in G \end{cases} \quad (5)$$

其中, p 为节点当选为候选区头的概率, r 为网络当前的运行轮数, G 为最近 $1/p$ 次分区中未当选为候选区头和区头的所有节点的集合。由式(5)可知,节点的历史能耗越大,节点当选为候选区头的概率就越小。在 $T(n)$ 中加入 $e_cost_i^r$ 因子是为了使历史能耗较低的节点优先当选为候选区头。由于 $e_cost_i^1 = 0$,因此首轮时所有 $T(n)$ 均为无穷大,所有节点全部当选为候选区头,这会产生不必要的能量消耗,因此对 $e_cost_i^1$ 进行调整,令 $e_cost_i^1 = 1$ 。

但是,由于随机数具有不确定性,在选举候选区头时可能会造成其数目较少或候选区头之间的距离较小,导致区头数目较少,如所有的候选区头都在某一个候选区头的竞争范围内,且其节点剩余能量最大,则会造成整个网络中只有一个区头,不利于网络负载的均衡。针对这个问题做如下调整:若区头只有一个,则利用式(5)重新选举候选区头。

在确定了候选区头后,每个候选区头在自己的竞争范围内广播竞选区头的信息。为了有效均衡网络的整体负载,算法希望距离基站较近的区面积较小,区内能耗较少,因此为每个候选区头设置一个竞争半径 R_i ,随着候选区头到基站距离的减小, R_i 也减小。假设候选区头的最大竞争半径为 R_i^0 ,则候选区头 i 的竞争半径 R_i 的计算公式为:

$$R_i = (1 - c \cdot \frac{d_{\max} - d(i, BS)}{d_{\max} - d_{\min}}) \cdot R_i^0 \quad (6)$$

其中, c 为 $0 \sim 1$ 之间的常数,用于调节 R_i 的取值范围,在仿真时取为 $0.5^{[8]}$; $d_{\max}$ 和 $d_{\min}$ 分别为网络中节点到基站的距离和最小距离, $d(i, BS)$ 表示节点 i 到基站的距离。

候选区头通过在自己的竞争范围内广播竞选区头的信息能够获得其他候选区头的信息。候选区头在获得其他候选区头的信息后,选举其中剩余能量最大的候选区头作为区头,区头竞争范围内的其余候选区头则退出竞选,成为普通节点。当选为区头的节点在自身竞争范围内广播获胜信息,申明自己成为区头,竞争范围内的普通节点在收到获胜信息后向区头发送加入请求信息。对于接收到两个或多个获胜信息的普

通节点,为了均衡网络的整体负载,其选择竞争半径较小的区加入。

非均匀分区的边界如图 1 所示, $AH_1, AH_2, AH_3, AH_4, AH_5$ 为区头, A, B, C, D 为普通节点, EF 和 GH 为弧。由图可知,节点 A 只位于 AH_1 的竞争半径内,故 A 属于区 AH_1 ; 节点 B, C 和 D 位于两个或多个竞争半径内,为了均衡网络负载,普通节点选择竞争半径较小的区加入,故 B 属于区 AH_2 , C 属于区 AH_3 , D 属于区 AH_4 。由于区 AH_1 和 AH_2 重叠部分的节点如 B 归属于区 AH_2 , 区 AH_1 和 AH_3 重叠部分的节点如 C 归属于区 AH_3 , 因此区 AH_1 和 AH_2 的边界为弧 EF , 区 AH_1 和 AH_3 的边界为弧 GH 。

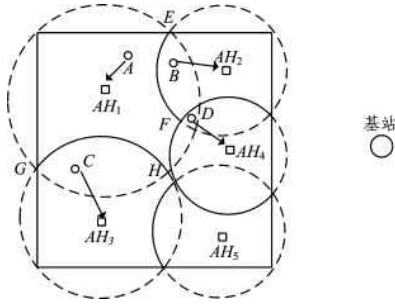


图 1 分区边界图

Fig. 1 Partition boundary

若初始分区后网络拓扑不变,则随着网络的运行,距离基站较近的区内节点的能量消耗较快,而距离基站较远的区内节点的能量消耗较慢。为了有效均衡分区之间的能耗,距离基站较近的区域面积应较小,而距离基站较远的区域面积应较大;但是,若每轮都重新分区,则会产生不必要的控制能耗,降低能量利用率。因此,本文算法在网络每运行一段时间后重新分区。重新分区的主要依据是距离基站最近的区内节点的平均剩余能量与其余区内节点的平均剩余能量的比值 $RA(r)$, 其计算公式为:

$$RA(r) = \frac{1}{nu_{1j} \in N_1} \sum E_j^r / \frac{1}{n - nu_{1j} \in N_1} \sum E_j^r \quad (7)$$

其中, N_1 为距离基站最近的分区, nu_{1j} 为区 N_1 中节点的总数, E_j^r 为第 r 轮开始时节点 j 的剩余能量。 $RA(r)$ 越小,说明距离基站较近的节点的平均剩余能量与距离基站较远的节点的平均剩余能量的差距越大,分区之间的能耗越不均衡,网络需要重新分区。

在分区完成之后,算法进入区内成簇阶段。

4.2 区内成簇

区内成簇阶段分为节点聚类 and 簇首选举。

4.2.1 节点聚类

现有的经典聚类算法如 K-means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) 等只是单一地根据节点之间的距离或节点的密度对节点聚类,在聚类过程中,聚类中心的随机选择严重影响着聚类效果,并且在确定节点所属的簇时需要不断地迭代更新,增加了算法的时间复杂度。为了解决这些问题,本文采用一种综合考虑距离和密度的新型聚类算法对节点进行聚类^[13]。

新型聚类算法的核心是通过设置阈值确定聚类中心,加强聚类效果。聚类中心应具有两个特点:有足够多的邻居节点;与其他聚类中心之间有足够大的距离。针对聚类中心的这两个特点,本文定义两个变量:局部密度 ρ_i 和距离 δ_i 。假设待聚类的节点集为 $S = \{x_i\}_{i=1}^{NN}$, $I_s = \{1, 2, \dots, NN\}$ 为对应的指标集, NN 表示区内节点的总数, $d_{ij} = \text{dist}(x_i, x_j)$ 表示节点 x_i 和 x_j 之间的欧氏距离。

局部密度 ρ_i 主要用来表示聚类中心的邻居节点个数。当聚类中心的邻居节点较多时,其局部密度较大;当聚类中心的邻居节点较少时,其局部密度也较小。假设节点 x_j 与节点 x_i 在同一个分区内,则根据局部密度的含义,其计算公式为:

$$\rho_i = \sum_{j \in N_i \setminus \{i\}} e^{-\left(\frac{d_{ij}}{d_c}\right)^2} \quad (8)$$

其中, d_c 为截断距离, N_i 为与节点 x_i 在同一个分区内的所有节点的集合。由上式可得,与 x_i 的距离小于 d_c 的节点越多, ρ_i 的值越大。

截断距离 d_c 的选取从某种程度上决定着聚类算法的成败,取值太大或者太小都会影响聚类效果。 d_c 的具体确定方法如下:假设任意分区 m 内共有 $N(P, m)$ 个节点,由于任意相异两点有两个重复的距离,因此共有 $NU = \frac{1}{2} N(P, m) \cdot [N(P, m) - 1]$ 个距离,将所有的距离按照升序排列,则得到:

$$d_1 \leq d_2 \leq \dots \leq d_{NU} \quad (9)$$

取 $d_c = d_{f(NU)}$, 其中 $f(NU)$ 为对 NU 进行四舍五入后得到的整数。若直接选取 d_c , 则对于规模不同的网络,合理的 d_c 取值在数量级上可能会有很大的差异,因此该聚类算法利用参数 t 确定 d_c 。当 t 确定后, d_c 也随之确定,在后面的仿真实验中, t 取经验值 0.02^[14]。

距离 δ_i 主要用于刻画此聚类中心与其他聚类中心的距离,一定程度上反映了类间距离的大小。距离 δ_i 越大,说明节点 x_i 作为聚类中心时与其他聚类中心的距离越大,两个类的差距越大,而这正是聚类的目标。设 $\{q_i\}_{i=1}^{N(P, m)}$ 表示 $\{\rho_i\}_{i=1}^{N(P, m)}$ 的一个降序排列的下标序,即满足:

$$\rho_{q_1} \geq \rho_{q_2} \geq \dots \geq \rho_{q_{N(P, m)}} \quad (10)$$

根据距离的含义,其计算公式为:

$$\delta_{q_i} = \begin{cases} \min\{d_{q_i, q_j}, j \leq i\}, & i \geq 2 \\ \max\{\delta_{q_j}, j \geq 2\}, & i = 1 \end{cases} \quad (11)$$

为了有效均衡分区间的能耗,距离基站较远的区内簇规模应较大;而距离基站较近的区内簇规模应较小,以节省簇首的簇内能耗,为多跳通信时的数据转发预留足够的能量。因此,聚类算法在每个分区内设置了不同的阈值,实现了不同的区内具有不同的簇规模,进而实现了非均匀分簇。

为了使距离基站较远的区内簇规模较小,而距离基站较近的区内簇规模较大,聚类算法以每个分区到基站的等效距离作为阈值选取的主要依据。假设分区 m 的所有节点集合 $T(P, m)$ 中共有 $N(P, m)$ 个节点,节点 x_i 到基站的距离为 $d(i, BS)$, 综合考虑区内所有节点,当节点到基站的距离越大,节点的局部密度越小时,分区到基站的等效距离越大,其

区内簇数目越少。等效距离 $D(P, m)$ 的计算公式为:

$$D(P, m) = \frac{1}{N(P, m)} \sum_{i \in T(P, m)} \frac{d(i, BS)}{\rho_i} \quad (12)$$

设全网中节点的最大局部密度和距离分别为 $\rho_{\max}$ 和 $\delta_{\max}$, $D_{\max}$ 和 $D_{\min}$ 分别表示等效距离 $D(P, m)$ 的最大值和最小值, 根据非均匀分簇的目标, 局部密度 ρ_i 的阈值计算公式为:

$$\rho_i^m = (1 - d \cdot \frac{D_{\max} - D(P, m)}{D_{\max} - D_{\min}}) \cdot \rho_{\max} \quad (13)$$

距离 δ_i 的阈值计算公式为:

$$\delta_i^m = (1 - e \cdot \frac{D_{\max} - D(P, m)}{D_{\max} - D_{\min}}) \cdot \delta_{\max} \quad (14)$$

其中, d, e 分别为调整局部密度阈值和距离阈值范围的常数。由式(13)、式(14)可得, 分区距离基站越近, 其等效距离越小, 阈值也越小, 而簇数目越多; 分区距离基站越远, 其等效距离越大, 阈值也越大, 而簇数目越少。

确定聚类中心的具体步骤为:

Step1 区头根据区内所有节点的位置信息计算任意两个节点之间的距离, 并确定 d_c ;

Step2 区头根据式(8)计算区内所有节点的局部密度, 并将节点按照局部密度降序排列, 然后根据式(11)计算节点距离;

Step3 区头根据式(13)、式(14)确定区内局部密度和距离阈值, 选举聚类中心, 并将聚类中心编号通过广播通告给区内各节点。

在确定非聚类中心节点所属的簇之前, 先定义用到的一些符号(n_c 表示聚类中心的个数):

$\{m_j\}_{j=1}^{n_c}$ 为聚类中心的节点编号;

$\{n_j\}_{j=1}^{N(P, m)}$ 为非聚类中心节点 x_j 的簇归属, 初始化为 0;

$\{c_j\}_{j=1}^{N(P, m)}$ 为节点所属簇的编号, 其初始化为: 若节点 x_j 为聚类中心且属于第 k 个簇, 则 $c_j = k$, 否则 $c_j = -1$ 。

由以上可知, 在聚类中心被确定之后, $\{m_j\}_{j=1}^{n_c}$ 也随之确定。而对于每个非聚类中心节点, n_j 的计算公式为:

$$n_j = \arg \min \{d_{ij}, \rho_i > \rho_j\} \quad (15)$$

区头根据非聚类中心节点的 n_j 确定 c_j , 即 $c_j = c_{n_j}$ 。 c_j 确定后, 区头将每个非聚类中心节点所属的簇通过广播通告给区内节点, 每个非聚类中心节点在收到广播信息后向聚类中心发送加入请求信息, 聚类中心在收到该信息后会返回一个确认加入的信息, 至此, 节点分簇完成。

在聚类完成后, 若聚类结果不变, 则随着网络的运行, 有的节点由于能量消耗较快而死亡, 原来的聚类并不适合此时的网络, 因此需要对节点进行重新聚类。设置能量阈值 EGO_i^{iter} , 当区内节点的平均剩余能量小于 EGO_i^{iter} 时, 节点重新聚类; 否则, 节点的聚类结果不变。若 EGO_i^{iter} 是静态的, 则当网络运行到后期时, 由于平均剩余能量变小, 节点会频繁聚类, 这会产生不必要的控制能耗, 因此设置 EGO_i^{iter} 为动态的。经过仿真实验可知(见图 4 和图 5), 根据式(16)设置阈值既能够减少聚类次数, 又可以最大化网络寿命。

$$EGO_i^{iter} = (EGO_i^{iter-1})^{1.5} \quad (16)$$

其中, $iter$ 为聚类次数; EGO_i^0 为初始阈值, 仿真中取为节点的

初始能量。

4.2.2 簇首选举

在聚类完成后, 虽然每个类中有确定的聚类中心, 但是在选举聚类中心时并没有考虑节点的剩余能量、簇内能耗以及普通节点的能耗均衡等因素, 只是从聚类的角度达到了聚类的目标: 类内距离尽可能小; 类间距离尽可能大。因此, 算法需要在每个类内重新选举簇首。

本文采用周期性选择机制选举簇首, 即每隔一段时间重新选举簇首。簇首的选择不但要考虑节点的剩余能量以及簇内总能耗, 而且还需要考虑普通节点的能耗均衡以及簇首的簇间能耗。

为了均衡节点的能耗, 簇首应具有较高的剩余能量。由能耗模型可知, 若保证簇内总能耗较小, 则普通节点到簇首的距离平方和应较小; 若要保证普通节点的能耗均衡, 则应将普通节点到簇首的平方距离的方差限制在一定的范围内; 若要使簇首的簇间能耗较小, 则簇首到基站的距离应较小。本文综合考虑以上 4 个方面, 采用下面的四步簇首选择机制选举簇首^[15]。其具体方法为:

Step1 保证簇首具有较高的剩余能量。将簇内节点的剩余能量按照降序排列。设簇内节点总数为 T_1 , 若 $1 \leq T_1 \leq 2$, 则取排名在首位的节点作为簇首; 否则, 取出排名前 $T_2 = \lfloor T_1/h_1 \rfloor$ 的节点作为待候选簇首, $h_1 \in [2, \frac{T_1+1}{2}]$ 。

Step2 保证簇内总能耗较小。分别将 Step1 中的待候选簇首作为簇首, 计算其簇内总能耗, 然后将所得的能耗按照升序排列, 取出排名前 $T_3 = \lfloor T_2/h_2 \rfloor$ 的待候选簇首作为候选簇首, $h_2 \in [2, \frac{T_2+1}{2}]$ 。

Step3 保证普通节点的能耗均衡。分别将 Step2 中的候选簇首作为簇首, 计算簇首的能耗方差, 然后将所得的方差按照升序排列, 取出排名前 $T_4 = \lfloor T_3/h_3 \rfloor$ 的候选簇首作为待定簇首, $h_3 \in [2, \frac{T_3+1}{2}]$ 。

Step4 保证簇首的簇间能耗较小。分别计算 Step3 中的待定簇首到基站的距离, 并将其按照升序排列, 将排名在首位的待定簇首作为最终簇首。

四步簇首选择机制能够十分灵活地选举簇首, 并且可以通过调整 h_1, h_2, h_3 的取值改变以上 4 个方面的权重, 从而适应不同的需求。本文在仿真阶段取 $h_1 = h_2 = h_3 = 2$ 。

重新选举簇首的主要依据为簇首的剩余能量与簇内普通节点的平均剩余能量的比值, 即:

$$V_k = E_{CH} / \frac{1}{q_k - 1} \sum E_i \quad (17)$$

其中, E_{CH} 为簇首的剩余能量, E_i 为普通节点的剩余能量, q_k 为第 k 个簇的节点数。当 V_k 小于阈值 V_t 时, 重新选择簇首; 否则, 簇首不变。本文在仿真阶段取 V_t 为 0.8。

簇首选举完成后, 获胜节点广播当选信息, 并且为簇内节点分配 TDMA 时隙。为了防止簇内节点向簇首传输数据时出现混乱, 簇内所有节点的时间应与簇首保持同步; 为了使节

点采集的数据具有实际意义,所有簇首的时间应与基站保持同步。

本文采用的时间同步分为两步:簇首与基站同步和簇首与簇内节点同步。由于监测区域内簇首的数目远小于节点数,因此簇首与基站的同步采用精度较高的 TPSN (Timing-sync Protocol for Sensor Network)协议^[16]。基站每隔一段时间(设在 T_1 时刻)向簇首发送同步信息,簇首在 T_2 时刻接收同步信息,并在 T_3 时刻发送同步应答,基站在 T_4 时刻接收同步应答,根据 4 个时刻的关系,可以得出基站和簇首的时间偏差 Δ 为:

$$\Delta = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \quad (18)$$

簇首根据 Δ 调整本地时钟,以达到与基站同步。簇首与簇内节点的同步采用 TPSN 和 RBS(Reference Broadcast Synchronization)^[16]相结合的同步机制。首先,簇首 A 在簇内广播同步信息,簇内节点记录到达时间,设节点 B 在 t_2 时刻收到同步信息并回复,根据式(18)计算 A 和 B 的时间偏差 Δ ;然后,簇首 A 再次广播同步信息,包括 t_2 和 Δ ,簇内节点收到同步信息后可以通过计算达到与簇首同步。至此,监测区域内的所有节点均达到与基站同步。

由于需要广播同步信息,因此时间同步会产生能量消耗。同步能耗分为两部分:簇首与基站同步时的能耗 E_1 和簇首与簇内节点同步时的能耗 E_2 。设同步信息为 l bit; d_{toBS} 为簇首到基站的平均距离,采用多径衰落模型; d_{toCH} 为簇内节点到簇首的平均距离,采用自由空间模型; NC 表示节点总数; K 为簇首数目。则:

$$E_1 = KlE_{dec} + K(lE_{dec} + l\epsilon_{amp}d_{toBS}^4) \quad (19)$$

$$E_2 = 2K(lE_{dec} + l\epsilon_{fs}d_{toCH}^2) + 2(NC - K)lE_{dec} \quad (20)$$

4.3 数据传输阶段

数据传输分为簇内通信和簇间通信,簇内通信采用单跳的方式,而簇间通信采用单跳和多跳相结合的混合传输方式,即当簇首到基站的距离小于阈值时,簇首直接将融合的数据传输至基站;否则,簇首将数据传输至中继簇首,由中继簇首转发至基站。

无线传感网路由协议中对簇间通信的主要改进是通过中继簇首转发数据。采用加权概率的方式来确定中继簇首,即同时考虑其剩余能量以及由其转发数据时的能耗。但是使用这种方法时两者的权重是静态的,而随着簇首剩余能量的变化,两者在选择中继簇首时的重要性也会变化,因此本文采用动态权重的方法优化簇间多跳路径。

为了延长网络寿命,中继簇首应具有较高的相对剩余能量。根据定义 2,中继簇首 x_k 在第 r 轮数据传输开始时的相对剩余能量因子 $W(ECH_k^r)$ 的计算公式为:

$$W(ECH_k^r) = \frac{ECH_k^r}{\frac{1}{P_r} \cdot \sum_{j \in U_s^r} ECH_j^r} \cdot \frac{ECH_k^r}{ECH_k^0} \quad (21)$$

其中, $U_s^r = \{k, d(k, BS) < d(i, BS)\}$,即中继簇首 x_k 为簇首 x_i 的前向簇首; ECH_k^r 表示中继簇首 x_k 在第 r 轮数据传输开始时的剩余能量, ECH_k^0 为节点的初始能量, P_r 表示 U_s^r 中元

素的个数。 $W(ECH_k^r)$ 越大,说明该中继簇首在此轮中的相对剩余能量越高。

根据自由空间模型可知,距离的平方直接影响着能耗的大小。簇首节点与中继簇首之间的距离越小,传输数据时的能耗便越低;中继簇首到基站的距离越小,转发数据时的能耗也越低。根据定义 3,簇首节点 x_i 经中继簇首 x_k 转发数据至基站的传输能耗因子 $EN(u_k)$ 的计算公式为:

$$EN(u_k) = 1 - \frac{d(i, k)^2}{\sum_{j \in U_s^r \setminus \{i\}} d(i, j)^2} - \frac{d(k, BS)^2}{\sum_{j \in U_s^r} d(j, BS)^2} \quad (22)$$

其中, $d(i, k)$ 为簇首节点 x_i 到中继簇首 x_k 的距离, $d(k, BS)$ 为中继簇首 x_k 到基站的距离。由上式可知,当 $d(i, k)$ 或 $d(k, BS)$ 较小时, $EN(u_k)$ 较大。

为了选择最优多跳路径,本文算法采用动态权重的方式。算法开始时,相对剩余能量因子和传输能耗因子所占权重相同,而随着网络的运行,节点的剩余能量逐渐变小,在选择中继簇首时,剩余能量因子的影响逐渐变大,传输能耗因子的影响逐渐变小。设节点相对剩余能量因子的权重为 α_r ,传输能耗因子的权重为 β_r ,根据相对剩余能量因子和传输能耗因子在网络运行过程中的重要性,其权重计算公式分别为:

$$\alpha_r = \alpha_{\min} + \frac{\alpha_{\max} - \alpha_{\min}}{r_{\max}} \cdot r \quad (23)$$

$$\beta_r = \beta_{\max} - \frac{\beta_{\max} - \beta_{\min}}{r_{\max}} \cdot r \quad (24)$$

其中, $\alpha_{\max}$ 和 $\alpha_{\min}$ 分别为相对剩余能量因子权重的最大值和最小值,在仿真时,分别取为 0.9 和 0.5; $\beta_{\max}$ 和 $\beta_{\min}$ 分别为传输能耗因子权重的最大值和最小值,在仿真时,分别取为 0.5 和 0.1; $r_{\max}$ 为网络运行的最大轮数, r 为网络当前运行的轮数。

簇首节点 x_i 选择中继簇首 x_k 的概率由式(25)计算:

$$relay_k = \alpha_r \cdot W(ECH_k^r) + \beta_r \cdot EN(u_k) \quad (25)$$

计算所有中继簇首 x_k 的 $relay_k$,并从中挑选概率最大的作为簇首节点 x_i 的中继簇首,直到中继簇首到基站的距离小于阈值。

5 实验验证及分析

为了验证本文算法的性能,利用 MATLAB 在相同的条件下仿真 LEACH^[4],DEBUC^[10],UCDP^[11] 和 NHRPNC 的多项性能,其主要参数如表 1 所列。

本文假设网络的生命周期为第一个节点死亡的轮数。

表 1 仿真参数表

Table 1 Parameters of simulation

参数名	数值
网络区域	200 m * 200 m
基站坐标	(100 m, 250 m)
节点数目	400
节点初始能量	0.5 J
发射电路损耗	50 nJ/bit
自由空间放大器系数	10 pJ/bit · m ⁻²
多径衰落放大器系数	0.0013 pJ/bit · m ⁻⁴
数据融合能耗	5 nJ/bit · signal ⁻¹
数据包大小	4000 bit
融合率	0.6

5.1 分区的必要性

为了证明本文算法在聚类之前采用分区的必要性,在相同的条件下仿真10次相同的网络,其结果如图2所示,横坐标代表仿真实验的次数,纵坐标为每次仿真实验的网络生命周期。

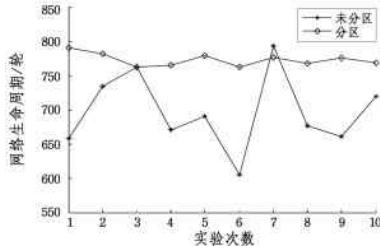


图2 实验次数与生命周期的关系

Fig. 2 Relationship between the number of experiments and the life cycle

由图可知,经过分区的算法的生命周期曲线基本位于不分区的上方,这说明采用分区能够延长网络寿命;分区的网络寿命波动性较小,而不分区的网络寿命波动性较大,这说明采用分区的网络更稳定,具有更好的网络监控质量。因此,本文算法在聚类之前需要对网络进行合理的分区。

5.2 RA 参数

本文算法用比值RA作为重新分区的依据,其目的是均衡分区之间的能耗。如图3所示,横坐标为RA的值;纵坐标为第一个节点死亡和最后一个节点死亡之间的轮数差,该数值越小表示能耗越均衡。由图可知,随着RA的增加,轮数差呈现先减小后增大的趋势,并且在RA=0.8时达到最低。因此,为了均衡分区之间的能耗,算法在仿真时取RA为0.8。

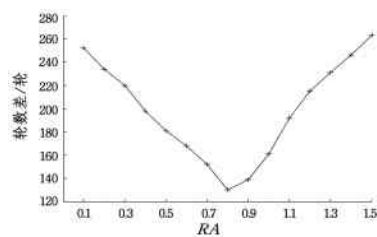


图3 RA与轮数差的关系

Fig. 3 Relationship between RA and wheel number difference

5.3 能量阈值

为了延长网络寿命,均衡节点能耗,本文算法每隔一段时间需要重新聚类,而其依据为区内节点的平均剩余能量与阈值的关系,因此阈值的设置至关重要。为了寻找较为合理的能量阈值,对以下4种不同的阈值公式进行仿真, $\lambda=0.6$ 。

线性阈值(LT): $EGO_t^{iter} = EGO_t^{iter-1} - 0.05$

常数迭代阈值(CIT): $EGO_t^{iter} = \lambda EGO_t^{iter-1}$

指数阈值(IT): $EGO_t^{iter} = \lambda e^{-0.5(iter-1)} EGO_t^{iter-1}$

幂次阈值(PT): $EGO_t^{iter} = (EGO_t^{iter-1})^{1.5}$

图4表示4种阈值公式的能量阈值随聚类次数增加的变化情况。由图可知,对于网络初期,当阈值公式为LT时,节点聚类次数较多,易产生不必要的控制能耗;而对于网络后期,当阈值公式为CIT时,节点频繁聚类,易导致过多的能量消耗。因此,IT和PT公式是较为合理的。

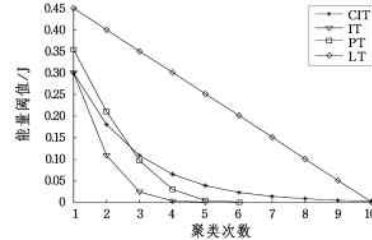


图4 4种能量阈值的变化图

Fig. 4 Changes of four energy thresholds

图5为本文算法采用4种不同阈值公式时的网络生命周期对比图。横坐标为仿真次数,本文在相同的条件下仿真10次;纵坐标为网络寿命,以轮数表示。由图可知,当阈值公式为LT,CIT和IT时,网络寿命较短且波动较大;而当阈值公式为PT时,网络寿命较长且波动较小。因此,综合考虑聚类次数和网络寿命,本文采用幂次阈值公式。

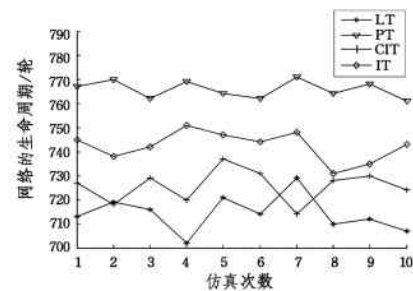


图5 4种能量阈值的网络生命周期对比图

Fig. 5 Comparison of network life cycle of four energy thresholds

5.4 生命周期

本文算法的核心是通过区内聚类生成非均匀簇的思想来缓解能量空洞问题,延长网络寿命。为了验证本文算法的有效性,下面比较4种协议的网络生存时间。图6显示了网络存活的节点数随仿真轮数的变化情况。从图中可以看出,本文算法相对于LEACH,DEBUC和UCDP3种协议具有较长的网络寿命,数据对比见表2。

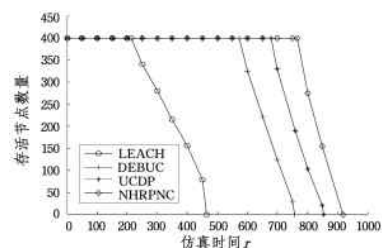


图6 网络生命周期对比图

Fig. 6 Comparison of network life cycle

表 2 4 种协议的节点死亡

Table 2 Node death of four protocols

路由协议	FND/轮	AND/轮	FEP/%	AEP/%
LEACH	214	459	257.5	99.56
DEBUC	572	768	33.74	19.27
UCDP	678	854	12.83	7.26
NHRPNC	765	916	—	—

其中 FND, AND 分别为网络中第一个节点死亡和全部节点死亡的轮数, FEP, AEP 分别表示本文算法在 FND 和 AND 上相对于其余 3 种协议的延长百分比。首先, 由于在不同的区内设置的阈值综合考虑了区内所有节点到基站的等效距离, 本文算法实现了网络中簇的非均匀分布; 同时结合簇间多跳的通信方式, 该算法有效地均衡了距离基站较近的簇和距离基站较远的簇的数据传输能耗, 一定程度上缓解了能量空洞。其次, 本文算法在簇内运用四级簇首选择机制灵活选举簇首, 降低了簇内能耗, 因此其第一个节点死亡的轮数明显晚于其他 3 种协议。

5.5 能量效率

图 7 对比了 4 种协议的网络总剩余能量随仿真轮数的变化情况。曲线的坡度越小, 说明网络能量消耗得越慢, 网络生存时间越长, 能耗越均衡。由图 7 可知, 本文算法的总剩余能量曲线一直位于其他 3 种协议曲线的上方, 说明本文算法的能耗速度慢, 这主要是因为本文采用了非均匀分簇和簇间多跳相结合的方式, 均衡了网络能耗。当网络运行到 300 轮时, LEACH, DEBUC 和 UCDP 的总剩余能量分别为 88.2 J, 123.5 J 和 133.1 J, 而本文算法的总剩余能量为 145.3 J, 说明本文算法能够有效减少网络能耗, 延长网络寿命, 其主要原因是本文采用了先分区后聚类的方式, 减少了频繁构簇的能耗。

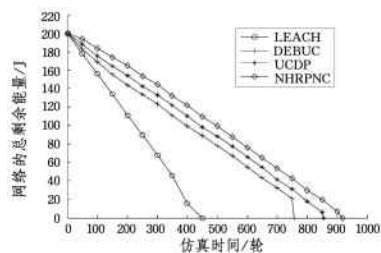


图 7 网络总剩余能量对比图

Fig. 7 Comparison of total network residual energy

为了进一步验证本文算法在均衡能耗方面的性能, 下面比较 4 种协议的节点平均剩余能量, 如图 8 所示。节点的平均剩余能量越高, 节点之间的能耗越均衡。

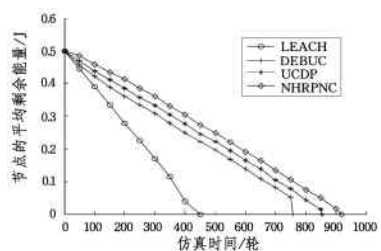


图 8 节点平均剩余能量对比图

Fig. 8 Comparison of average residual energy of nodes

由图可知, 本文算法的节点平均剩余能量一直比其余 3

种协议高, 说明本文算法在均衡网络能耗方面具有较好的性能, 这主要是由于本文算法不但实现了簇的非均匀分布, 而且还在簇间通过动态权重的方式优化了多跳路径。

结束语 为了解缓成簇路由协议中由节点负载不均导致的能量空洞问题, 本文在 LEACH, DEBUC 和 UCDP 协议的基础上提出了一种基于新型聚类的负载均衡的非均匀分层路由协议。该协议首先利用改进的 LEACH 协议阈值函数选举区头, 并将网络进行非均匀分区, 以均衡分区之间的能耗; 其次在每个区头上运用新型聚类算法将节点非均匀分簇; 然后在簇内采用四级簇首选择机制周期性地选举簇首, 以减少簇内总能耗和均衡节点能耗; 最后在簇间多跳通信选择中继簇首时, 通过考虑其相对剩余能量因子和传输能耗因子两者权重的动态变化, 优化簇间多跳路径, 以均衡簇首之间的能耗。实验结果表明, 所提算法能够显著延长网络寿命, 均衡节点之间的能耗, 缓解能量空洞。

虽然本文算法在仿真实验中具有较好的性能, 但实际监测环境的范围越来越大, 下一步将对本文算法作出改进, 使其更加适应于大规模的网络环境。

参考文献

- [1] DAHNIL D P, SINGH Y P, HO C K. Analysis of adaptive clustering algorithms in wireless sensor networks[C]// Proceedings of IEEE Conference on Communication Systems, Singapore: IEEE, 2010: 51-55.
- [2] HOANG D C, YADAV P, KUMAR R, et al. Real-time implementation of a harmony search algorithm-based clustering protocol for energy-efficient wireless sensor networks [J]. IEEE Transactions on Industrial Informations, 2014, 10(10): 774-783.
- [3] LI B, WANG W J, YIN Q Y, et al. An energy-efficient geographic routing based on cooperative transmission in wireless sensor networks [J]. Science China Information Sciences, 2013, 56: 4757-4762.
- [4] HEINZELMAN W R, CHANDRAKASAN A, BALAKRISHNAN H. Energy-efficient communication protocol for wireless microsensor networks[C]// Proceeding of the 33rd Annual Hawaii International Conference on System Sciences, Hawaii: IEEE Computer Society, 2000: 300-304.
- [5] LI Y, SUN L J, WANG R C, et al. Improvement of LEACH algorithm in wireless sensor network [J]. Journal of Computer Research and Development, 2011, 48(suppl.): 131-134. (in Chinese)
- 李悦, 孙力娟, 王汝传, 等. 一种改进的无线传感器网络 LEACH 算法[J]. 计算机研究与发展, 2011, 48(增刊): 131-134.
- [6] LI J Z, WANG H T, TAO A. An energy balanced clustering routing protocol for WSN [J]. Chinese Journal of Sensors and Actuators, 2013, 26(3): 396-401. (in Chinese)
- 李建洲, 王海涛, 陶安. 一种能耗均衡的 WSN 分簇路由协议[J]. 传感技术学报, 2013, 26(3): 396-401.
- [7] YE M, LI C, CHEN G, et al. EECS: An energy efficient clustering scheme in wireless sensor networks[C]// Proceedings of the 24th IEEE International Conference on Performance, Computing and Communications, Piscataway: IEEE Press, 2005: 535-540.

- [8] LI C F, CHEN G H, YE M, et al. An uneven cluster based routing protocol for wireless sensor networks [J]. Chinese Journal of Computer, 2007, 30(1): 27-36. (in Chinese)
李成法, 陈贵海, 叶懋, 等. 一种基于非均匀分簇的无线传感器网络路由协议[J]. 计算机学报, 2007, 30(1): 27-36.
- [9] JIANG C J, TANG X L, XIANG M. Unequal clustering routing protocol for wireless sensor networks based on PSO algorithm [J]. Application Research of Computers, 2012, 29(8): 3074-3078. (in Chinese)
蒋畅江, 唐贤伦, 向敏. 基于 PSO 的无线传感器网络非均匀分簇路由协议[J]. 计算机应用研究, 2012, 29(8): 3074-3078.
- [10] JIANG C J, SHI W R, TANG X L, et al. Energy-balanced unequal Clustering routing protocol for wireless sensor networks [J]. Journal of Software, 2012, 23(5): 1222-1232. (in Chinese)
蒋畅江, 石为人, 唐贤伦, 等. 能量均衡的无线传感器网络非均匀分簇路由协议[J]. 软件学报, 2012, 23(5): 1222-1232.
- [11] SUN Y Q, PENG J, LIU T, et al. Uneven clustering routing protocol based on dynamic partition for wireless sensor network [J]. Journal on Communications, 2014, 35(1): 198-206. (in Chinese)
孙彦清, 彭舰, 刘唐, 等. 基于动态分区的无线传感器网络非均匀成簇路由协议[J]. 通信学报, 2014, 35(1): 199-206.
- [12] WANG M Y, WANG X, JIANG H. Cluster-chain based low-energy consumption hierarchical routing protocol [J]. Computer Science, 2015, 42(11): 144-148. (in Chinese)
王梦莹, 王鑫, 蒋华. 基于簇首成链的低能耗层次路由协议[J]. 计算机科学, 2015, 42(11): 144-148.
- [13] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks [J]. Science, 2014, 344: 1492-1495.
- [14] WANG S L, WANG D K, LI C Y, et al. Clustering by fast search and find of density peaks with data field [J]. Chinese Journal of Electronics, 2016, 25(3): 397-402.
- [15] ZHAI C J, XU J M, LIU Y G. Energy-consumption balancing routing protocol based on regions [J]. Chinese Journal of Sensors and Actuators, 2016, 29(1): 81-87. (in Chinese)
翟春杰, 徐建闽, 刘永桂. 基于分区的能耗均衡路由协议[J]. 传感技术学报, 2016, 29(1): 81-87.
- [16] YANG C K. Time synchronization algorithms in Zigbee wireless sensor networks [D]. Shanghai: Shanghai Jiaotong University, 2015. (in Chinese)
杨成侃. Zigbee 无线传感器网络时间同步算法研究[D]. 上海: 上海交通大学, 2015.
- (上接第 97 页)
- [6] LI Z C. Research on node location algorithm for wireless sensor network in environmental monitoring [J]. Computer Applications and Software, 2010, 27(2): 241-243. (in Chinese)
李忠成. 环境监测无线传感器网络节点定位算法研究[J]. 计算机应用与软件, 2010, 27(2): 241-243.
- [7] CHEN H Y. The research of wireless sensor network positioning technology based on ranging technique [D]. Chengdu: Southwest Jiaotong University, 2006. (in Chinese)
陈红阳. 基于测距技术的无线传感器网络定位技术研究[D]. 成都: 西南交通大学, 2006.
- [8] ZAIDI S, EL ASSAF A, AFFES S, et al. Range-free node localization in multi-hop wireless sensor networks [C] // Wireless Communications and Networking Conference (WCNC). IEEE, 2016: 1-7.
- [9] LI L F, CUI W W, ZHAO X. Research on improved DV-Hop positioning algorithm based on jump distance correction [J]. Journal of Henan Institute of Science and Technology (Natural Science Edition), 2016, 44(1): 51-56. (in Chinese)
李琳芳, 崔微微, 赵欣. 基于跳距修正的 DV-Hop 改进定位算法研究[J]. 河南科技学院学报(自然科学版), 2016, 44(1): 51-56.
- [10] ZHANG W L, SONG Q X. Improvement of DV-Hop algorithm for genetic algorithm [J]. Journal of Chongqing University, 2015, 38(3): 159-166. (in Chinese)
张万礼, 宋启祥. 遗传算法的 DV-Hop 算法改进[J]. 重庆大学学报, 2015, 38(3): 159-166.
- [11] YANG G, YI Z, TIANQUAN N, et al. An improved genetic algorithm for wireless sensor networks localization [C] // 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA). IEEE, 2010: 439-443.
- [12] LI W, ZHOU W. Genetic algorithm-base localization algorithm for wireless sensor networks [C] // 2011 Seventh International Conference on Natural Computation (ICNC). IEEE, 2011: 2096-2099.
- [13] GU M S, YAN Y S, YOU L, et al. An improvement of localization algorithm based on particle swarm optimization and simulated annealing in wireless sensor networks [J]. Journal of Information & Computational Science, 2013, 10(5): 1497-1505.
- [14] FAN S P, LUO D, LIU Y L. DV-Hop location algorithm based on jump distance and improved particle swarm optimization [J]. Chinese Journal of Sensors and Actuators, 2016, 29(9): 1410-1415. (in Chinese)
范时平, 罗丹, 刘艳林. 基于跳距与改进粒子群算法的 DV-Hop 定位算法[J]. 传感技术学报, 2016, 29(9): 1410-1415.
- [15] GONG Y G. Study on the combination of particle swarm optimization and genetic algorithm [J]. Journal of Jining University, 2008, 29(6): 20-22. (in Chinese)
巩永光. 粒子群算法与遗传算法的结合研究[J]. 济宁学院学报, 2008, 29(6): 20-22.
- [16] HOLLAND J H. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence [J]. Quarterly Review of Biology, 1992, 6(2): 126-137.
- [17] KENNEDY J, EBERHART R. Particle swarm optimization [C] // IEEE International Conference on Neural Networks, 1995. IEEE, 2002: 1942-1948.