文章编号:1007-5321(2018)03-0101-06 DOI:10.13190/j.jbupt.2017-202

基于计算节点和转发节点的 WSN 自组织聚簇算法

薛寒寒, 王 柏, 张 雷, 黄 海
(北京邮电大学 计算机学院, 北京 100876)

摘要: 针对无线传感器网络(WSN)中数据计算需求和由簇首负载过重引起的热点问题和能量空洞问题,提出基于计算节点和转发节点的自组织聚簇算法(SCATN),对簇首功能进行分解,以计算节点满足数据计算需求,以转发节点进行数据转发,并通过分布控制解决热点问题和能量空洞问题. 聚簇过程采用自组织方式控制功能节点的生成、分布,从而解决分布不均匀和连接性问题. 同时,普通节点自主更换归属簇以及时、细粒度地调整计算节点负载. 仿真实验结果表明,与现有几种聚簇算法相比,SCATN 算法可有效地提高网络生存时间,增加基站的吞吐量,降低丢包率.

关 键 词: 无线传感器网络; 自组织聚簇; 热点; 能量空洞
中图分类号: TP393 **文献标志码**: A

A Self-Organized Clustering Algorithm Based on Computation and Transmission Node for WSN

XUE Han-han, WANG Bai, ZHANG Lei, HUANG Hai
(School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: A self-organized clustering algorithm based on computation node and transmission node (SCATN) for wireless sensor network (WSN) was proposed to satisfy computation requirement and solve the problem of hot spot and energy hole caused by cluster head overload. SCATN employs computation node and transmission node to undertake the cluster head's function data computation and transmission. The distribution probability of functional nodes is controlled to tackle with the problems raised. The generation and distribution of functional node is controlled by self-organized manner to solve the problem of distribution and connection. The ordinary nodes choose its belonged cluster to adjust the computation node's load. Simulation indicates that SCATN can effectively extend the network lifetime, improve the throughput at the sink and decrease the packet loss rate in comparison with several existing clustering algorithms.

Key words: wireless sensor network; self-organized clustering; hot spot; energy hole

随着物联网(IoT, internet of things)和大数据的快速发展,感知数据已从简单的温湿度数据发展到视频等多媒体内容,且感知数据量庞大,超出无线传感器网络(WSN, wireless sensor network)的传输能力,将大量数据传输到一个中心进行处理,从技术上和经济上都是不现实的. 现在的传感器节点有较强的计算和存储能力,可以在网络中进行数据处理[1]. 现有聚簇方法通常采用双簇首(CH, cluster head)的方法解决热点问题,采用非均匀聚簇的方法解决能量空洞问题,并假设 CH 的数据处理任务简

收稿日期: 2017-09-22
作者简介: 薛寒寒(1986—), 男, 硕士生, E-mail: hhxue86@126.com; 王 柏(1962—), 女, 教授, 博士生导师.

单,不能满足数据计算的需求. 张人上等[2]和刘壮等[3]采用双 CH 的方法,2 个 CH 分别负责簇内数据收集、处理和簇间数据转发. 戴志强等[4]对规模较大的簇选取副 CH 以减少 CH 的负担. Mo 等[5]采用非均匀聚簇的方法,根据到 sink 的距离确定每环的 CH 数目,使靠近 sink 的环有更多的 CH. Afsar 等[6]使靠近 sink 的 CH 之间距离更小,从而实现非均匀聚簇. Vural 等[7]提出了异步聚簇协议(ACP, asynchronous clustering protocol),根据各环的流量负载计算各环的 CH 基准概率,根据基准概率计算簇范围,使靠近 sink 的环内节点的 CH 基准概率大,簇范围小,从而实现非均匀聚簇. 吴标等[8]计算了不同网络区域的 CH 比例,通过 CH 发射功率的自适应调整实现非均匀聚簇. 非均匀聚簇算法虽然能够获得 CH 能耗的近似相等,但是由于靠近 sink 的网络区域 CH 节点较多,增加了该区域的总体能耗,从而不能很好地解决能量空洞问题. 此外,IoT 和大数据需要 CH 进行数据计算、网络管理和数据转发等任务,因此需要稳定的网络拓扑. 传统的双 CH 算法和非均匀聚簇算法不能满足数据计算需求,没有考虑计算节点的分布和双 CH 的相互影响.

针对上述问题,笔者提出基于计算节点和转发节点的自组织聚簇算法(SCATN, self-organized clustering algorithm based on computation node and transmission node). 该聚簇算法将 CH 功能分解到相互独立的计算节点(CCH, computation function of cluster head)和转发节点(TCH, transmission function of cluster head),根据数据计算需求确定 CCH 的分布,根据数据转发需求确定 TCH 的分布,并通过综合控制 CCH 和 TCH 的分布关系以缓解能量空洞问题. SCATN 在算法运行过程进行存在性和连接性检查从而确保 TCH 的连接性,并且通过普通节点自主更换归属簇及时调整 CCH 的负载.

1 系统模型

1.1 网络模型

网络模型如图 1 所示. 大规模 WSN 中,网络会形成以 sink 为中心的等宽环形区域,环内节点到 sink 有相同的跳数[9]. 节点可调整自己的传输功率,从而覆盖不同的通信范围[6-7]. SCATN 算法中节点有 3 个固定的通信范围 R_{com1}、R_{com2} 和 ExR(h)(其中 h 为节点的跳数). 网络中节点角色分为普通节点、CCH 和 TCH. 普通节点进行数据感知,并在范围 R_{com1} 内相互交换电量信息,以获得局部范围的最大电量,用于 CCH 的选举. CCH 负责簇内节点的数据收集和处理,转发数据到 TCH,通信范围为 R_{com1},如图 1 中 cch2 和 cch3 的簇范围,与 tch2、tch3 的转发数据范围均为 R_{com1}. TCH 负责转发数据到 sink,转发范围为 R_{com1},如图 1 中 tch2 与 tch3 转发数据到 tch4,tch4 转发数据到 sink. 为使 TCH 有较好的分布,每个 TCH 节点在范围 ExR(h)内进行存在性和连接性检查,并且在较小的附近范围 R_{com2} 内进行角色的更换,如图 1 中 tch5 的附近范围为 ExR(1),tch5 存在范围为 ExR(2),跳数不同存在范围不同.

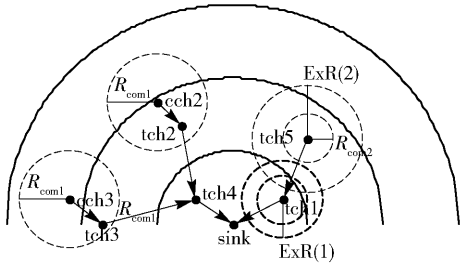


图 1 网络模型

1.2 基本概念

Vural 等[7]采用环形能量计算方法分析 CH 负载,忽略了 CH 的数据收集、计算负载,因而不太准确. TCH 的负载仅为数据转发负载,所以使用环形能量计算方法更加准确.

定义 1[7] 跳数是 h 的环内节点成为 TCH 的基准概率为 $t_{benp}(h)$(见式(1)),成为 TCH 的概率为 $P_{TCH}(i)$.

$$t_{benp}(h)=\frac{K^2-h^2+2h-1}{(2h-1)K^2}t_{benp}(1) \tag{1}$$

$$t_{benp}(1)=\frac{M}{C\sigma\pi R^2} \tag{2}$$

其中:K 为网络的最大跳数,M 为网络中初始 TCH 数目,$C=\sum_{h=1}^{K}\frac{K^2-h^2+2h-1}{K^2}$,$\sigma$ 为节点分布密度,R_0 为环宽.

Vural 等[7]中对网络进行流量分析,使各环 CH 转发的数据流量相等,从而求出各环所需的 CH 数量,并进一步确定各环 CH 的基准概率. 考虑节点的电量和 TCH 基准概率,$P_{TCH}(i)$的计算公式为

$$P_{TCH}(i)=t_{benp}(\mathrm{hop}(i))\frac{E(i)}{\varepsilon_{max}(i)} \tag{3}$$

其中:$\mathrm{hop}(i)$为节点 i 的跳数,$E(i)$为节点 i 的电

量，$\varepsilon_{\max}(i)$ 为节点 i 邻居范围内节点的最大电量.

定义 2　定义 ExR(h) 为跳数是 h 的环内 TCH 的存在范围，不同环的存在范围不同. 节点通过其 TCH 列表检查在其该范围内是否有 TCH 存在，如果没有就会启动 TCH 选举过程，如果有则不会生成新的 TCH，从而对 TCH 的拓扑进行控制. 由于各环的 TCH 分布随机且互不影响，所以各环 TCH 的分布符合泊松分布，该范围内 TCH 节点的平均个数 λ 为 $\sigma\pi \mathrm{ExR}^2(h)t_{\mathrm{benp}}(h)$. 由 $P(x=k)=\frac{\mathrm{e}^{-\lambda}\lambda^k}{k!}$ 可知，在 ExR(h) 范围内至少有一个 TCH 的概率为 $P\{n\geqslant 1\}\approx 1-P\{n=0\}=1-\mathrm{e}^{-\sigma\pi \mathrm{ExR}^2(h)t_{\mathrm{benp}}(i)}$，要保证存在范围内 TCH 一定存在，所以 $P\{n\geqslant 1\}$ 非常接近 1，可设为 99%[7]，可得

$$\mathrm{ExR}(h)\approx\sqrt{\frac{2\ln 10}{\sigma\pi t_{\mathrm{benp}}(h)}} \tag{4}$$

计算节点 CCH 的分布需求具有以下 3 个特点：①远离 sink 的区域存在更多的传感器节点，所以需要更多的计算节点；②在远离 sink 的区域进行数据计算，可以减少网络中传输的数据量；③在靠近 sink 的网络区域分布较少的 CCH，以减少该网络区域的能耗，从而均衡整个网络的负载.

定义 3　定义 $c_{\mathrm{benp}}(h)$ 为跳数是 h 的环内节点成为 CCH 的基准概率，成为 CCH 的概率为 $P_{\mathrm{CCH}}(i)$. 以 h_1 和 h_2 代表任意 2 个不同环的跳数，以 a_1、a_2 为对应环的面积，为实现整个网络的负载均衡，使各环的高能耗节点 TCH 与 CCH 的数目之和与环中节点的总数目成比例，即

$$\frac{a_1\sigma(t_{\mathrm{benp}}(h_1)+c_{\mathrm{benp}}(h_1))}{a_2\sigma(t_{\mathrm{benp}}(h_2)+c_{\mathrm{benp}}(h_2))}=\frac{a_1\sigma}{a_2\sigma}$$

得到 $t_{\mathrm{benp}}(h_1)+c_{\mathrm{benp}}(h_1)=t_{\mathrm{benp}}(h_2)+c_{\mathrm{benp}}(h_2)$. 由于 $0<t_{\mathrm{benp}}(h),c_{\mathrm{benp}}(h)<1$，所以可以设各环节点的 2 种基准概率之和为 1，得到

$$c_{\mathrm{benp}}(h)=1-t_{\mathrm{benp}}(h) \tag{5}$$

考虑节点的电量和 CCH 基准概率，$P_{\mathrm{CCH}}(i)$ 的计算公式如下：

$$P_{\mathrm{CCH}}(i)=c_{\mathrm{benp}}(\mathrm{hop}(i))\frac{E(i)}{\varepsilon_{\max}(i)} \tag{6}$$

定义 4　定义 $C_{\mathrm{eff}}(i)$ 为 CCH 节点 i 的影响力. $C_{\mathrm{eff}}(i)$ 的计算考虑以下 3 个因素：①节点 i 的电量；②节点 i 的基准概率；③节点 i 的簇内成员节点数目. 普通节点根据 CCH 节点的影响力自主更换归属簇，可使同环 CCH 的负载均衡，远离 sink 的环 CCH 有更多的簇内节点. CCH 从 hello 消息中获得簇内节点的电量和数目信息，$C_{\mathrm{eff}}(i)$ 为

$$C_{\mathrm{eff}}(i)=\frac{E(i)c_{\mathrm{benp}}(\mathrm{hop}(i))}{\mathrm{num}(i)} \tag{7}$$

其中 num(i) 为 CCH 节点 i 的簇内成员节点数目.

2　SCATN 算法

2.1　TCH、CCH 选择

网络中数据转发负载较大，所以首先进行 TCH 的选择. TCH 的选择根据自组织群体中个体行为原则进行连接性和存在性检查，从而保证 TCH 有较好的分布和连接. TCH 选择完成后，进行 CCH 的选择. TCH 和 CCH 选择的流程如图 2 所示.

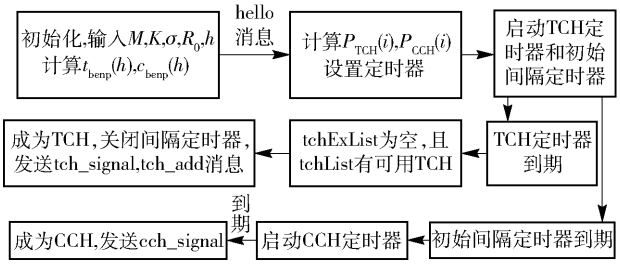


图 2　TCH、CCH 选择流程

算法步骤如下.

步骤 1　根据式(1)和式(2)计算节点的 TCH 基准概率，进而根据式(4)和式(5)分别计算 TCH 存在范围和节点的 CCH 基准概率.

步骤 2　跳数为 0 的 sink 节点在 R_{com1} 范围内周期性广播 tch_signal 消息，普通节点在 R_{com1} 范围内周期性广播 hello 消息，在 R_{com2} 范围内周期性广播 onForTCH 消息. hello 消息中包含节点的电量信息，节点收到 hello 消息后，计算其邻居范围内节点的最大电量，并根据式(3)和式(6)计算节点的 TCH 和 CCH 概率.

步骤 3　节点设置其 TCH 定时器为 $(1-P_{\mathrm{TCH}}(i))T$，设置其 CCH 定时器为 $(1-P_{\mathrm{CCH}}(i))T$，并设置其初始间隔定时器为 T.

步骤 4　节点启动其 TCH 定时器和初始间隔定时器. 当 TCH 定时器到期后，检查其 tchExList 确定 TCH 存在范围内是否有 TCH 存在，如果没有，检查其 tchList 列表内是否有可用 TCH 存在，如果有，则该节点成为 TCH，关闭其初始间隔定时器，不再参与 CCH 的选择. 并且，在 R_{com1} 范围内周期性发送 tch_signal 消息，在其存在范围内广播一次 tch_add 消息. 节点收到 tch_signal 消息后更新其 tchList 列

表. 节点收到 tch_add 消息后,如果跳数相同,则更新其 tchExList 列表,并停止 TCH 定时器.

步骤5 节点的初始间隔定时器到期后,启动其 CCH 定时器. CCH 定时器到期后,成为 CCH,在其簇范围 R_{com1} 内发送一次 cch_add 消息,并周期性发送 tch_signal 消息. 普通节点收到 cch_add 消息后,如果其 CCH 定时器没有到期,则停止其 CCH 定时器.

图1中,sink 发送 tch_signal 消息,sink 跳数为0,所以跳数为1的节点更新其 tchList,不会更新 tchExList. tch1 所代表节点的 TCH 定时器到期后,检查其 tchExList 和 tchList,若满足条件,则可以成为 TCH,然后发送 tch_signal 消息,跳数为1且在 ExR(1)范围内的节点收到该消息后更新 tchExList 并关闭 TCH 定时器,不会再成为 TCH,跳数为2且在 ExR(2)范围内的节点收到该消息后会更新其 tchList. 因为所有节点的 TCH 定时器时间都小于初始间隔定时器,所以初始间隔定时器到期后,TCH 选择结束开始 CCH 选择,cch2 所代表节点的 CCH 定时器到期后,成为 cch2,并在簇范围 R_{com1} 内发送 cch_signal 消息,收到该消息的普通节点关闭 CCH 定时器并加入 cch1.

2.2 TCH、CCH 角色更换

所提出的聚簇算法适用于大规模 WSN,节点根据自身及局部范围内动态变化的节点和网络拓扑信息,做出适应性调整. CCH 考虑电量和 CCH 基准概率信息进行角色更换,使电量较大、远离 sink 的节点能够优先成为 CCH,同时要保证新的 CCH 节点到 TCH 的连接性. CCH 节点 j 收到簇内节点 i 的 hello 消息后判断,如果 j 的 tchList 不为空,且 $E(j)c_{benp}(\mathrm{hop}(j)) > \theta_1 E(i)c_{benp}(\mathrm{hop}(i))$($\theta_1$ 为 CCH 角色更换阈值),则节点 j 代替 i 成为 CCH.

TCH 作为转发节点,为了保证路由的连续性,在附近范围 R_{com2} 内进行角色更换,以维护路由拓扑的稳定性. TCH 节点 i 收到邻居范围内同环节点 j 的 onForTCH 消息后判断,如果 j 的 tchList 不为空,且 $E(j) > \theta_2 E(i)$(θ_2 为 TCH 角色更换阈值),则节点 j 代替 i 成为 TCH.

2.3 节点更换归属簇

CCH 通过 cch_signal 广播其影响力,普通节点收到该消息后,自主判断是否更换归属簇,从而在簇结构不发生大的变化的情况下对 CCH 的负载进行及时和细微的调整.

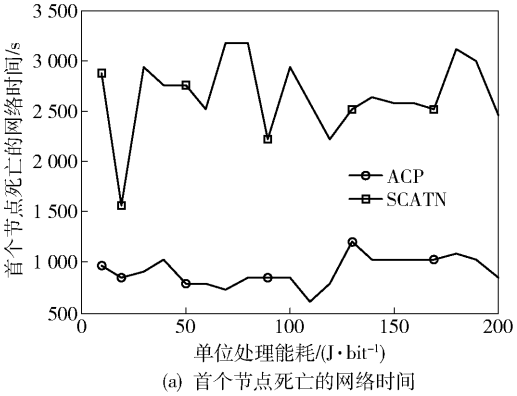
普通节点 i,其当前归属簇为 k,收到 CCH 节点 j 的 cch_signal 消息后判断,如果满足 $C_{\mathrm{eff}}(j) > \theta_3 C_{\mathrm{eff}}(k)$($\theta_3$ 为归属簇更换阈值),则节点 i 更换其归属簇为 j.

3 仿真结果及分析

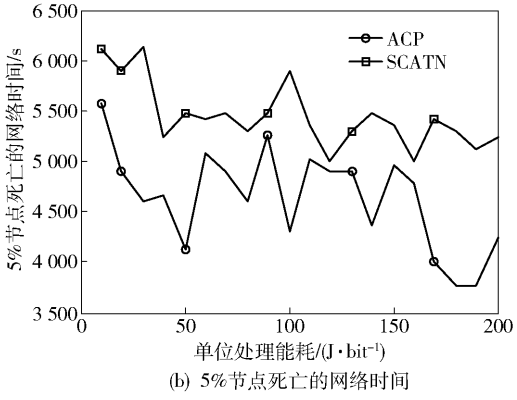
使用 Matlab 对算法进行仿真,在 400 m × 400 m 的区域中随机地部署1 000个节点,sink 位于网络区域的中心. 节点的初始电量在1~2 J 内随机分布. 在数据处理能耗从 5×10^{-9} J/bit ~ 200×10^{-9} J/bit 的多组实验中,TCH 的平均数目为51,占节点总数的0.051%,TCH 仅作为转发节点,会减少监测感知的节点数,但在大规模网络中,TCH 节点数目较少且分布均匀,节点 TCH 角色转换后会重新开始监测,所以对网络的影响较少.

网络生存时间用首个节点和5%节点的死亡时间表示. ACP 和 SCATN 的首个节点死亡的网络时间对比如图3(a)所示. 从图中可以看出,SCATN 的网络时间比 ACP 有显著提高,通过20组实验的统计,ACP 的网络时间平均值为907 s,SCATN 的网络时间平均值为2 658 s. 5%节点死亡的网络时间对比如图3(b)所示. 从图中可以看出,SCATN 的效果比 ACP 好,这是因为 ACP 中 CH 既承担数据转发,又进行计算,所以负载较重,网络生存时间较短. SCATN 的平均时间是5 452 s,ACP 的平均时间是4 636 s,二者差距比首个节点死亡的网络时间差距小,这是因为 SCATN 进行了拓扑控制,在不满足存在性和连接性条件的时候不进行 TCH 角色转换,会造成 TCH 节点死亡,而 ACP 中 CH 在电量较低时可以随时进行角色转换,不会导致节点直接死亡. 由于低功耗自适应聚簇分层型协议(LEACH, low energy adaptive clustering hierarchy)中 CH 数据单跳传输到 sink,所以在首轮就会出现节点死亡,网络生存时间较短. 而不均匀聚簇路由算法(UCR, unequal cluster-based routing)在设置数据传输距离与 ACP、SCATN 相同时,其 CH 节点没有分布控制和连接性保证,所以丢包率较高,大大减少了数据转发能耗,所以网络生存时间较长.

在单位数据能耗从 5×10^{-9} J/bit ~ 200×10^{-9} J/bit 的多组实验中,LEACH、UCR、ACP 和 SCATN 的平均丢包率对比如图4所示,平均吞吐量对比如图5所示. 从图4和图5中可以看出,与 LEACH、UCR、ACP 相比,SCATN 能够有效降低丢包率,提高



(a) 首个节点死亡的网络时间



(b) 5%节点死亡的网络时间

图 3　网络生存时间

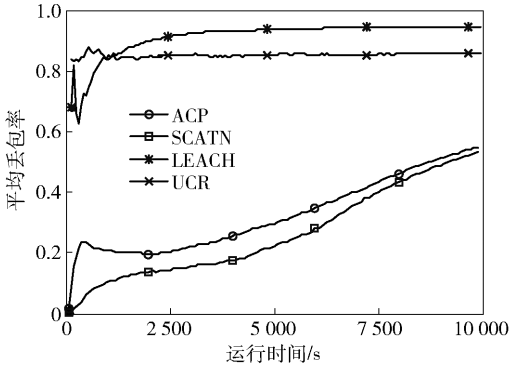


图 4　sink 丢包率

吞吐量. 这是因为 SCATN 对整个网络进行负载均衡控制,能够有效减缓能量空洞,并且 SCATN 对转发节点进行连接性和存在性检查,能够保证转发节点有更好的分布和连接性;而 LEACH、UCR 没有进行分布控制,CH 分布不均匀,且没有进行连接性检查,导致丢包率较高和吞吐量较低;ACP 在选择 CH 时通过调整簇范围保证连接性,但是在进行角色转换的时候没有分布控制和连接性检查. 随着网络运行时间的增长,不同协议的丢包率和吞吐量之间的差距逐渐减小,这是因为能量空洞问题固有,长时间运行后靠近 sink 的节点逐渐死亡.

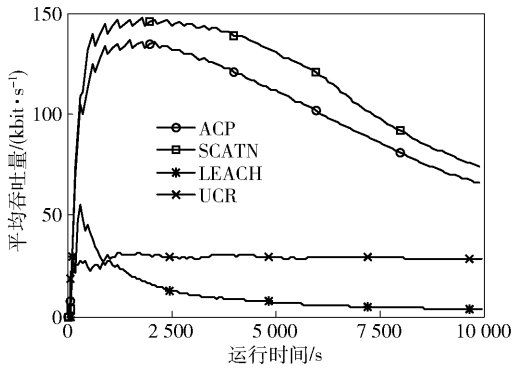


图 5　sink 吞吐量

4　结束语

作为大数据时代重要数据来源的 WSN,其感知数据越来越复杂,数据规模越来越大,超出了网络的传输能力,需要在网络中进行数据计算. 为此,笔者提出了 SCATN 聚簇算法,使用计算节点进行数据计算,使用转发节点进行数据转发,并控制它们的分布,从而实现整个网络的负载均衡,缓解热点问题和能量空洞问题. 仿真实验验证了在不同的数据计算能耗下该方法的有效性. 在以后的工作中,将关注节点休眠机制与聚簇算法的结合,由 CCH 进行节点的休眠调度,并进一步在聚簇结构的基础上进行服务发现、服务选择、服务中断恢复等服务计算的研究,以验证算法的有效性.

参考文献:

[1] Li Chao, Hu Yang, Liu Longjun, et al. Towards sustainable in-situ server systems in the big data era[C]//2015 International Symposium on Computer Architecture. New York: ACM, 2015: 14-26.

[2] 张人上, 曲开社. 基于双簇首网格调度的 WSNs 能量空洞缓解[J]. 传感器与微系统, 2014, 33(10): 133-136.
Zhang Renshang, Qu Kaishe. Energy hole alleviation of WSNs based on dual cluster head grid scheduling[J]. Transducer and Microsystem Technologies, 2014, 33(10): 133-136.

[3] 刘壮, 冯欣, 王雁龙, 等. 基于双簇头聚类分簇和数据融合的无线传感器网络路由算法[J]. 吉林大学学报, 2015, 53(5): 1013-1017.
Liu Zhuang, Feng Xin, Wang Yanlong, et al. An improved clustering routing algorithm based on double-CH clustering and data fusion in WSN[J]. Journal of Jilin University, 2015, 53(5): 1013-1017.

[4] 戴志强, 严承, 武正江. 一种新的无线传感器网络非均匀分簇双簇头算法-PUDCH 算法[J]. 传感技术学报, 2016, 29(12): 1912-1918.
Dai Zhiqiang, Yan Cheng, Wu Zhengjiang. New uneven double cluster head clustering algorithm for WSN-PUDCH algorithm [J]. Chinese Journal of Sensors and Actuators, 2016, 29(12): 1912-1918.

[5] Mo Yijun, Wang Bang, Liu Wenyu, et al. A sink-oriented layered clustering protocol for wireless sensor networks [J]. Mobile Networks and Applications, 2013, 18(5): 639-650.

[6] Afsar M M, Younis M. An energy-and proximity-based unequal clustering algorithm for wireless sensor networks [C]// LCN 2014. Washington DC: IEEE Computer Society, 2014: 262-269.

[7] Vural S, Navaratnam P, Wang Ning, et al. Asynchronous clustering of multihop wireless sensor networks [C]// ICC 2014. Piscataway: IEEE Press, 2014: 472-477.

[8] 吴标, 崔琛, 余剑, 等. 基于非均匀成簇的无线传感器网络多跳路由算法[J]. 计算机科学, 2017, 44(2): 157-162.
Wu Biao, Cui Chen, Yu Jian, et al. Multi-hop routing algorithm for wireless sensor networks based on uneven clustering[J]. Computer Science, 2017, 44(2): 157-162.

[9] 刘唐, 彭舰, 陈果, 等. 基于密度控制的传感器网络能量空洞避免策略[J]. 计算机学报, 2016, 39(5): 993-1006.
Liu Tang, Peng Jian, Chen Guo, et al. Avoidance of energy hole problem based on density control mechanism for wireless sensor networks[J]. Chinese Journal of Computers, 2016, 39(5): 993-1006.