

## 基于数据关联性的无线传感器 网络簇内数据管理算法

向敏<sup>1</sup> 石为人<sup>2</sup>

**摘要** 无线传感器网络 (Wireless sensor networks, WSNs) 节点能量有限, 能量高效的数据管理和延长网络寿命是该领域的技术难题. 在以簇构建的传感器网络中, 利用节点的计算和分析功能, 提出了基于数据关联性的簇内数据管理算法. 簇头利用误差函数和模糊函数分析成员感知数据的关联性, 获取节点感知数据综合支持度, 由此将成员节点划分为冲突节点、补充节点和可靠节点, 对不同类别节点采用不同的调度规则以便降低簇内能耗和尽可能实现簇间节点能耗均衡, 并给出了簇头数据融合的处理方法. 仿真结果表明算法能够实现簇内数据分类管理, 并能有效降低簇内数据收发量和延长网络寿命.

**关键词** 无线传感器网络, 数据管理, 数据关联性, 综合支持度, 网络寿命

**DOI** 10.3724/SP.J.1004.2010.01343

### A Cluster Data Management Algorithm Based on Data Correlation of Wireless Sensor Networks

XIANG Min<sup>1</sup> SHI Wei-Ren<sup>2</sup>

**Abstract** The nodes of wireless sensor networks (WSNs) are extremely power constrained, so energy-efficient data management and prolonging the networks lifetime are the major challenges in this researching area. With node calculation and analysis function, a cluster data management algorithm based on data correlation is presented for the network formed with clusters. Cluster head analyzes its member data correlation using error function and fuzzy function, and gains the integrative supportabilities of its members' sensing data. Based on the integrative supportabilities, the members of the cluster are classified as conflict nodes, complementary nodes and reliable nodes, and the related nodes scheduling rules for the different types are given to reduce the energy consumption of the cluster and balance the energy consumption among members, and the data fusion mechanism of the cluster head is given. The simulation results demonstrate that the presented algorithm can realize the cluster data classification management, reduce the amount of the data transmission, and prolong the network lifetime.

**Key words** Wireless sensor networks (WSNs), data management, data correlation, integrative supportability, network lifetime

收稿日期 2009-10-16 录用日期 2010-03-17

Manuscript received October 16, 2009; accepted March 17, 2010

国家自然科学基金 (50804061), 国家科技重大专项基金 (2009ZX07528-003-09, 2009ZX03006-001-5), 重庆市教委自然科学基金 (KJ100511), 重庆市科委自然科学基金项目 (2009BB2278), 重庆邮电大学博士启动基金 (A2010-01) 资助

Supported by National Natural Science Foundation of China (50804061), National Science and Technology Major Project (2009ZX07528-003-09, 2009ZX03006-001-5), Chongqing Education Administration Program (KJ100511), Natural Science Foundation of Chongqing Science and Technology Commission (2009BB2278), and Scientific Research Foundation for Ph. D. of Chongqing University of Posts and Telecommunication (A2010-01)

1. 重庆邮电大学网络化控制与智能仪器仪表教育部重点实验室 重庆 400065  
2. 重庆大学自动化学院 重庆 400044

1. Key Laboratory of Network Control and Intelligent Instrument of Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing 400065 2. College of Automation, Chongqing University, Chongqing 400044

无线传感器网络 (Wireless sensor networks, WSNs) 节点能量有限, 能量高效的数据管理和延长网络寿命是该领域很多学者十分关注的问题.

无线传感器网络的每个节点都具有完整的运算和存储功能, 就节点本身来说, 它是一个独立的计算单元, 能够实现数据的管理, 可以根据网络结构或者任务需求动态调整自身行为, 包括数据感知、数据收发以及自身状态的控制. 目前在无线传感器网络中广泛应用的分簇算法, 将众多的节点分成不同的簇, 簇头实现对本簇成员数据的收发和对成员数据的分析. 无线传感器网络节点分布密度通常很大, 簇内很多成员具有相同或相近的感知范围, 导致其感知数据具有一定的关联性. 如果能够通过分析节点感知数据的关联性, 获取节点感知数据的可靠程度, 进而将这些节点数据分类管理, 如保留可靠度高的节点数据, 去除可靠度低的节点数据并暂时或永久关闭对应节点, 将能有效减少簇内通信能耗和提高簇头数据融合的精度. 在保持覆盖度和连通度不变的情况下适当关闭一些冗余节点, 将有利于减少网络能量消耗, 从而延长网络寿命. 然而如何划分节点类型是无线传感器网络的一个技术难点. 目前, 多数算法根据节点位置信息、节点感知半径实现冗余节点划分<sup>[1-3]</sup>. 文献 [4] 提出了一种基于概率而不是位置信息的冗余节点分析方法, 即节点可以根据自身感知半径内的邻居节点数量计算出自身成为冗余节点的概率. 然而, 由于绝大多数节点的感知硬件和通信部件是完全独立的两个模块, 即通信半径和感知半径不一致. 因此, 要知道节点感知半径中的邻居节点数量, 需要有专门的硬件进行判断, 这会加重硬件成本. 文献 [5] 提出基于时空域频率动态带宽的数据去时空冗余算法和相应的路由策略, 该算法有效去除了数据的时空冗余, 降低了网络的通信能耗. 在无线传感器网络中, 往往用多个传感器测量同一个物理参数, 甚至采用不同性能的传感器以达到“优势互补”, 从而提高测量的可靠性和测量精度<sup>[6]</sup>. 然而, 不同传感器所获取的测量值会有偏差, 这种偏差一方面与传感器本身的物理特性相关, 另一方面与数据分析中的算法有关. 事实上, 多数节点感知数据偏差各异; 同时, 随着能量下降, 感知偏差或数据的不确定性可能加大, 如果单纯依靠节点的感知范围或节点位置信息来决策出冗余节点, 可能降低网络的可靠性和有效性. 如果能根据节点感知数据的特性并结合节点位置信息准确划分节点类别, 如永久或暂时关闭不可靠节点, 精确控制冗余节点并对其实施活动和休眠调度控制, 这无疑会降低网络能耗和减少网络信道拥塞的风险, 并能提高簇头融合数据的精度, 提高网络的可靠性.

为此, 本文提出根据节点感知数据关联性, 簇头利用误差函数和模糊关联函数获取成员感知数据的综合支持度, 并结合综合支持度增量, 实现簇内节点数据的分类管理, 在不牺牲簇头融合数据精度的条件下, 减少簇内数据收发量, 从而改善网络能量效率并延长网络寿命.

## 1 算法的设计

无线传感器网络的数据与大量节点的感知数据密切相关, 因此在实施节点感知数据分类处理时, 应当充分考虑节点感知数据的特性. 对于分布在某一区域的众多节点, 当其监测对象相同或相近时, 节点感知数据的关联性具有强、中、弱的情况, 如果能够将这些节点分类处理, 保留感知数据有效的节点, 关闭感知数据可靠度低的节点, 一方面有助于提高测量数据的精度, 另一方面有利于降低簇内通信能耗. 此外, 在节点高密分布的网络中, 特别是网络运行初期, 绝大多

节点感知数据关联性很强,而且其感知数据多数有效.显然,在具有关联性很强的感知数据中很可能存在大量的冗余数据,如果能对这些冗余数据进行有效管理,减少冗余数据传送而带来的通信开销和信道竞争,这无疑会降低网络能耗并延长网络生存周期.

### 1.1 节点感知数据综合支持度

设  $X_i, X_j$  分别为节点  $n_i, n_j$  的感知数据,且各节点感知数据服从高斯分布,其概率分布函数分别为  $p_i(x), p_j(x)$ ,其中  $x_i, x_j$  为其一次感知数据,则点  $n_i, n_j$  的感知数据置信距离测度  $d_{ij}$  可表示为<sup>[6-7]</sup>

$$\begin{aligned} d_{ij} &= \operatorname{erf} \left( \frac{x_j - x_i}{\sqrt{2} \times \sigma_i} \right) \\ d_{ji} &= \operatorname{erf} \left( \frac{x_i - x_j}{\sqrt{2} \times \sigma_j} \right) \end{aligned} \quad (1)$$

其中,  $\operatorname{erf}$  为误差函数.由式(1)可知,  $0 \leq d_{ij} \leq 1$ ,  $d_{ij}$  越小说明节点  $n_i$  被节点  $n_j$  支持的程度越高.令  $r_{ij} = 1 - d_{ij}$ ,则对于有  $n$  个节点的网络,其关联矩阵  $R$  可表示为

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{bmatrix} \quad (2)$$

其中,  $r_{ij}$  表示第  $i$  个节点被第  $j$  个节点支持的程度,  $r_{ij}$  越大,支持程度越高.但式(2)无法综合权衡每个节点感知数据的有效性,文献[8]采用模糊理论中的相关性函数确定第  $i$  个节点感知数据被其他节点所支持的程度.令  $s(i|j) = r_{ij}$ ,相关性函数可定义为

$$s(i|j) = \frac{s(i|j)}{\max[s(i|j), s(j|i)]} \quad (3)$$

节点  $i$  的综合支持度可表示为

$$\beta_i = \min(s(i|j)), \quad j = 1, 2, \dots, n \quad (4)$$

显然,  $\beta_i$  越大,则该节点感知数据被其他节点支持的程度越高,其数据为可靠数据;反之,为不可靠数据,应当丢弃.

对于节点高密分布的网络,如果簇头每轮都接收来自其成员的数据并计算其综合支持度,一方面簇头计算处理能耗急剧增加;另一方,簇间通信能耗并没有改善,且导致簇头能量将急剧下降,进而过早失去功效.显然,这反而不利于簇内节点数据的管理.由式(4)计算得到的各节点综合支持度  $\beta_i$  满足  $0 \leq \beta_i \leq 1$ ,因此,可以根据该支持度确定节点类型.设参数  $\varepsilon_1, \varepsilon_2$  为综合支持度的两个阈值,满足条件:  $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$ .当节点综合支持度满足  $\beta_i < \varepsilon_1$  时,为冲突节点(Conflict node);当节点综合支持度满足  $\varepsilon_1 \leq \beta_i < \varepsilon_2$  时,为补充节点(Complementarity node);当节点综合支持度满足  $\beta_i \geq \varepsilon_2$  时,为可靠节点(Reliable node),即:

$$\begin{cases} \text{Conflict node,} & \beta_i < \varepsilon_1 \\ \text{Complementary node,} & \varepsilon_1 \leq \beta_i < \varepsilon_2 \\ \text{Reliable node,} & \beta_i \geq \varepsilon_2 \end{cases} \quad (5)$$

当接收到所有成员的感知数据后,簇头开始分析成员感知数据的特性.在感知数据的分析、处理过程中,冲突节点的

数据不利于提高簇头融合数据的精度,因此在簇头融合成员数据前,应当剔除冲突节点的数据.此外,在分布密度大的簇内,具有覆盖范围相同或相近的节点数目较多,在网络运行早期,节点剩余能量较多,多数处于正常状态,从而导致这些节点感知数据的关联性很强,其综合支持度都很高.如果这些节点的感知数据每轮都发送到簇头,这不利于降低簇内通信能耗.为降低簇内数据收发量,可以将具有高综合支持度的节点划分为活动节点和冗余节点(处于休眠状态),并设置活动与休眠的调度周期.令  $\varphi$  为冗余节点数目  $n_f$  占可靠节点总数  $n_h$  的百分比,  $T_r$  为簇头的工作轮回周期,冗余节点的休眠时间为  $T_{\text{off}}$ ,则有:

$$\begin{cases} \varphi = \frac{n_f}{n_h} \times 100\% \\ T_{\text{off}} = \nu \times T_r \end{cases} \quad (6)$$

其中,  $\nu = 1, 2, \dots$ , 该参数与节点剩余能量相关.节点在剩余能量较少时,感知数据的偏差较大,此时  $\nu$  取值应当偏小;反之,  $\nu$  取值较大.在实际应用中,由于环境的干扰,某些节点可能在一段时间出现感知数据偏差较大,其综合支持度降低,但这并不能说明该节点彻底失效,因此在网络工作早期,对于综合支持度较低的节点也应当考虑活动和休眠的调度机制.当其感知数据的综合支持度小于  $\varepsilon_1$  时,由簇头通知其在下一个轮回进入休眠状态.为计算方便,设冲突节点的休眠时间和冗余节点的休眠时间相同,即  $T_{\text{off}} = \nu \times T_r$ .

### 1.2 休眠节点数目确定

由以上讨论可知,在一个簇中需要进入休眠状态的节点包括两部分:冲突节点和冗余节点.式(4)可以计算得到各个节点的综合支持度,由于节点物理特性的差异性,其综合支持度不完全相同,而可靠数据最小综合支持度  $\varepsilon_2$  的选取对于节点分类有重要的影响.如果  $\varepsilon_2$  选取太小,大量节点感知数据被视为有效数据,难以对节点感知数据进行准确筛选,进而划分出大量冗余的节点并将其进行周期性活动/休眠调度.在大量冗余节点休眠阶段,簇头融合数据的精度会降低,并可能导致网络覆盖出现大量感知盲点从而降低了网络的有效性.如果  $\varepsilon_2$  选取太大,尽管确定了具有高可靠数据的节点,但冗余节点数目较少,使网络中仍然存在大量活动节点具有相同或相近的感知区域,其冗余数据依然存在,这对于减少簇内通信量和降低通信能耗并不理想.因此,可靠节点最小综合支持度  $\varepsilon_2$  的合理选取与网络覆盖和网络性能有着密切关系.对于节点密度较高的网络,当节点感知数据综合支持度大于 0.75 ( $\varepsilon_2 = 0.75$ ) 时,可以认为该节点数据为可靠数据;而当节点密度较小时,节点分布稀疏,各节点感知数据差异较大,导致各节点综合支持度减小,此时  $\varepsilon_2$  取值相应减小,具体需要根据实际的网络规模决定.

此处冗余节点并没有考虑节点的物理位置信息,而是从感知数据关联性计算得到.然而,冗余节点的关闭必须考虑感知区域的覆盖问题.假设网络中节点呈泊松分布,分布在半径为  $R$  的圆形子区域.设节点感知半径为  $R_s$ ,文献[3,9]给出了传感器网络所提供的服务质量期望  $Q$  满足:

$$Q = 1 - e^{-\lambda \pi R_s^2} \quad (7)$$

其中,  $\lambda$  为满足期望的服务质量的节点密度.设满足服务质量期望值  $Q$  的可靠节点数为  $n_r$ ,有  $\lambda = n_r / (\pi R^2)$ ,则在给定期望服务质量的条件下,可靠节点数目  $n_r$  须满足:

$$n_r \geq \frac{R^2}{R_s^2} \times \ln[(1 - Q)^{-1}] \quad (8)$$

由式 (8) 可知, 当确定了满足覆盖要求的最少可靠节点数目后,  $\varepsilon_2$  取值已经给出了限制条件. 如果可靠节点数目小于或等于  $n_r$ , 则不存在冗余节点; 反之, 存在冗余节点. 令  $(n_h - n_r)$  的可靠节点为候选冗余节点. 在实际运行中, 如果将所有的候选冗余节点全部关闭, 可能导致网络覆盖出现较多的盲区, 同时可能降低簇头数据融合的精度, 因此需要根据实际情况在候选冗余节点中选取部分为冗余节点. 在节点高密分布的网络运行早期, 多数节点感知数据的综合支持度较高, 此时存在较多的可靠节点, 则冗余节点数目  $n_f$  占候选冗余节点数目  $(n_h - n_r)$  的百分比  $\alpha$  取值满足:

$$\begin{cases} \alpha = \frac{n_f}{n_h - n_r} \times 100\% \\ n_r = \frac{R^2}{R_s^2} \times \ln[(1 - Q)^{-1}] \end{cases} \quad (9)$$

式 (6) 中的参数  $\varphi$  与式 (9) 中的参数  $\alpha$  具有关联性, 但后者更能反映冗余节点与候选冗余节点的关系, 因此在后面的讨论中使用  $\alpha$  来描述冗余节点所占比例. 由于节点分布的随机性和分布区域形状的不同, 为提高簇头融合数据的精度, 在实际应用中, 通常保留的可靠节点数目要大于  $n_r$ , 因此冗余节点数目需满足  $n_f < n_h - n_r$ . 当网络中节点数目极少时, 如果仍然考虑冗余节点控制, 则可能使簇头融合数据的精度急剧下降, 进而失去分类的意义. 比如, 某簇中存在 3 个节点, 如果去除一个冗余节点, 其融合数据精度的可靠性将无法得到保障.

根据容错理论, 若某一个传感器数据和 1/3 以上的传感器数据冲突, 该节点的数据应当删除<sup>[10]</sup>. 由式 (4) 计算得到的节点感知数据综合支持度反映了该节点感知数据与其他节点感知数据冲突的程度, 综合支持度越小, 则可以认为冲突越大. 因此, 对于综合支持度最小的 1/3 的节点可以认为其感知数据与其他节点冲突. 一个节点分布密度极高的网络, 在其运行早期, 多数节点工作正常, 出现异常感知数据的节点较少, 可以将  $\varepsilon_1$  取值为该簇中综合支持度最小的 1/3 的节点的最大值. 但对于节点分布稀疏的网络或者高密分布网络中存在较多死亡节点时,  $\varepsilon_1$  的取值需要减小, 否则会导致网络中活动节点数目急剧减少, 甚至会降低网络的可靠性. 针对一些特定应用,  $\varepsilon_1$  应当做适当的调整: 如果传感器测量偏差都在容许范围内且被观测对象的数据变化微小或缓慢, 如天然气产区泄漏的甲烷、硫化氢等气体浓度监测、冰川的物理参数监测等, 多数传感器具有较高支持程度,  $\varepsilon_1$  取值可以较大, 但不能超过该簇中综合支持度最小的 1/3 的节点的最大值; 而如果监测对象的数据变化幅度大且频率高或实时性要求较高的场合,  $\varepsilon_1$  取值应当较小. 当选定  $\varepsilon_1$  后, 不可靠节点也就完全确定了.

### 1.3 休眠节点控制规则确定

式 (6) 给出了不可靠节点和冗余节点休眠时间的计算方法, 但具体哪些节点应当休眠, 或者哪些节点应当首先考虑休眠, 哪些节点应当作为休眠候选节点仍需要相应规则. 对于一个可靠节点或不可靠节点, 其综合支持度改变情况各有不同. 令  $\Delta\beta_i(k)$  为节点综合支持度的增量, 其中  $i$  为节点编号,  $k$  为采样时间序号, 则

$$\Delta\beta_i(k) = \beta_i(k) - \beta_i(k-1) \quad (10)$$

$\Delta\beta_i(k)$  在一定程度上表明了各节点感知数据综合支持度的变化趋势. 由式 (10) 可知, 综合支持度增量有三种情况:

$\Delta\beta_i(k) > 0$ ,  $\Delta\beta_i(k) = 0$ , 和  $\Delta\beta_i(k) < 0$ . 根据节点感知数据综合支持度  $\beta_i(k)$  及其增量  $\Delta\beta_i(k)$  的关系, 节点行为控制分为以下几种情况:

1)  $\beta_i(k) < \varepsilon_1$  且  $\Delta\beta_i(k) < 0$ , 说明该节点感知数据综合支持度低且未来感知数据的综合支持度将继续变差, 该节点感知数据不参与簇头本次的数据融合, 簇头发送命令通知该节点暂时或永久性关闭, 休眠优先级  $\sigma = 3$ .

2)  $\beta_i(k) < \varepsilon_1$  且  $\Delta\beta_i(k) = 0$ , 说明该节点感知数据综合支持度低且未来感知数据的综合支持度将维持现状, 该节点感知数据不参与簇头本次的数据融合, 簇头发送命令通知该节点暂时或永久性关闭, 休眠优先级  $\sigma = 2$ .

3)  $\beta_i(k) < \varepsilon_1$  且  $\Delta\beta_i(k) > 0$ , 说明该节点感知数据综合支持度低但未来感知数据的综合支持度将增加, 该节点感知数据不参与本次数据融合, 休眠优先级  $\sigma = 1$ .

4)  $\varepsilon_1 \leq \beta_i(k) < \varepsilon_2$ , 说明该节点为补充节点, 需在下一个轮回继续向簇头发送感知数据, 休眠优先级  $\sigma = 0$ .

5)  $\beta_i(k) \geq \varepsilon_2$  且  $\Delta\beta_i(k) > 0$ , 说明该节点感知数据综合支持度高且未来感知数据的综合支持度可能继续增加, 该节点感知数据参与本次数据融合. 该节点可以为候选冗余节点, 休眠优先级  $\sigma = 3$ .

6)  $\beta_i(k) \geq \varepsilon_2$  且  $\Delta\beta_i(k) = 0$ , 说明该节点感知数据综合支持度高且未来感知数据的综合支持度可能保持不变, 该节点感知数据参与本次数据融合. 该节点可以为候选冗余节点, 休眠优先级  $\sigma = 2$ .

7)  $\beta_i(k) \geq \varepsilon_2$  且  $\Delta\beta_i(k) < 0$ , 说明该节点感知数据综合支持度高但未来感知数据的综合支持度将降低, 该节点仍然可以划分为候选冗余节点, 休眠优先级  $\sigma = 1$ .

根据节点综合支持度  $\beta_i(k)$  及其增量  $\Delta\beta_i(k)$  的关系, 休眠节点的控制规则如表 1 所示. 其中, 对于情况 1), 如果节点处于活动早期或剩余能量较多时, 可能由于环境有较大干扰, 出现异常数据, 这时可以将该节点休眠一段时间后重新查询数据并计算其综合支持度; 当然, 如果节点剩余能量不足, 其感知行为异常, 所得到的感知数据本身就不够准确, 该节点已经处于失效状态, 应当永久性关闭, 避免由簇头对其进行数据查询、处理所带来的通信能耗.

在实际应用中, 如果能够比较准确控制  $\varepsilon_1$ , 可以修改表 1 的控制规则, 减少判定条件, 如只要节点的感知数据综合支持度  $\beta_i(k) < \varepsilon_1$ , 都必须进行休眠调度控制, 则表 1 简化为表 2 的控制规则.

表 1 休眠节点控制规则

Table 1 The controlling rules of inactive nodes

	$\Delta\beta < 0$	$\Delta\beta = 0$	$\Delta\beta > 0$
$\beta \geq \varepsilon_2$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$
$\varepsilon_1 \leq \beta < \varepsilon_2$	$\sigma = 0$	$\sigma = 0$	$\sigma = 0$
$\beta < \varepsilon_1$	$\sigma = 3$	$\sigma = 2$	$\sigma = 1$

表 2 简化的休眠节点控制规则

Table 2 The simplified controlling rules of inactive nodes

	$\Delta\beta < 0$	$\Delta\beta = 0$	$\Delta\beta > 0$
$\beta \geq \varepsilon_2$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$
$\varepsilon_1 \leq \beta < \varepsilon_2$	$\sigma = 0$	$\sigma = 0$	$\sigma = 0$
$\beta < \varepsilon_1$	$\sigma = 3$	$\sigma = 3$	$\sigma = 3$

#### 1.4 冗余节点的选取与调度

由以上讨论可知,同处一个簇内的节点,将由簇头划分为可靠节点、补充节点和冲突节点.对于综合支持度及其增量满足  $\beta_i(k) < \varepsilon_1$  且  $\Delta\beta_i(k) < 0$  的不可靠节点,其感知数据不利于簇头提高融合数据的精度,这些节点不论位于什么位置,应当暂时或永久关闭,以便降低簇内通信能耗和提高簇头融合数据的精度.对于可靠节点,如果将某一子区域的所有  $(n_h - n_r)$  可靠节点均划为冗余节点,尽管活动的可靠节点数目依然满足式(8),但会造成网络实际覆盖面积减小,进而出现较多的盲点,这会降低网络的服务质量.此外,由于传感器测量偏差各异,在网络运行中,某些节点感知数据综合支持度或综合支持度增量出现频繁跳变,这可能导致候选冗余节点及其数目动态改变,从而使某些可靠节点长期扮演冗余角色,而另一些可靠节点一直处于活动状态,这显然不利于节点能耗均衡和延长网络寿命.因此,合理分配可靠节点担任冗余角色的次数,对实现簇内节点能耗均衡和延长网络寿命非常重要.

由于感知数据综合支持度低的冲突节点其数据被视为无效,因此这些节点感知范围或其覆盖区域对于网络已经无效;对于补充节点,在休眠规则中规定这些节点不参与休眠调度,因此这些节点对于网络的覆盖状况并不改变;而影响网络覆盖的是冗余节点.假定监测区域是半径为  $R$  的圆形区域,在该区域上存在  $n_h$  个可靠节点,每个节点的坐标均已知,节点感知半径均为  $R_s$ .如果由式(8)计算得到在给定服务质量下的最少节点数目满足  $n_r \geq n_h$ ,则所有可靠节点均需保留;反之,可以划分部分节点为冗余节点.显然,在尽可能提高网络覆盖度的前提下,将具有最多覆盖重叠的节点划分为冗余节点,将能减少由于关闭冗余节点带来覆盖度降低的风险.

设节点  $n_i$  和  $n_j$  的中心坐标分别为  $(x_i, y_i)$  和  $(x_j, y_j)$ . 设节点  $n_i$  与  $n_j$  的中心距离为  $g_{ij}$ , 则  $g_{ij}$  可表示为

$$g_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (11)$$

设二节点感知重叠部分面积为  $S_d(i, j)$ . 当  $g_{ij} > 2R_s$ , 则节点  $n_i$  和  $n_j$  没有重叠区域,其重叠区域面积为  $S_d(i, j) = 0$ ; 当  $0 \leq g_{ij} \leq 2R_s$ , 则节点  $n_i$  和  $n_j$  有重叠区域,其重叠区域面积  $S_d(i, j)$  可表示为

$$S_d(i, j) = 2R_s^2 \arccos \frac{g_{ij}}{2R_s} - \frac{g_{ij}}{2} \sqrt{4R_s^2 - g_{ij}^2} \quad (12)$$

如果  $g_{ij} = 0$ , 则节点  $n_i$  和  $n_j$  完全重叠,其重叠区域面积为  $S_d(i, j) = \pi R_s^2$ . 式(12)中由于存在反三角函数,在比较各节点感知重叠区域时,计算复杂.现采用曲线拟合机制,重叠区域的面积  $S_d(i, j)$  可近似表示为

$$S_d(i, j) = \left( \frac{\sqrt{3}}{2} - \frac{\pi}{6} \right) g_{ij}^2 - \left( \frac{\pi}{6} + \sqrt{3} \right) R_s g_{ij} + \pi R_s^2 \approx 0.34g_{ij}^2 - 2.26R_s g_{ij} + \pi R_s^2 \quad (13)$$

由于节点随机分布,要保证簇中节点在每个轮回都具有最大覆盖面积,必然要求某些感知重叠范围最小的候选冗余节点长期工作,这会造成这类节点能量消耗较快,不利于簇内节点能耗均衡.但在节点工作早期,在考虑覆盖面积最大化的条件下,可以将感知覆盖重叠范围较大的可靠节点划分为冗余节点.

在一个相对稳定的网络中,某些冗余节点在后续的工作中可能再次成为冗余节点,而另一些可靠节点长期处于活动

状态,造成其能耗较大而快速死亡,这显然不利于延长网络寿命,必须合理分配可靠节点成为冗余节点的次数,尽可能使簇内节点能耗均衡.但从覆盖度考虑,关闭具有较大覆盖重叠面积的冗余节点能降低出现较大盲区的风险,显然这两者出现矛盾.因此,如何调度冗余节点对网络性能的至关重要.

文献[11]提出了簇头采用连续担任本地控制中心的机制,将能有效降低簇头更换频率和将少簇内附加能耗.设冗余节点在担任簇头前休眠次数为  $T_x$ , 在式(6)中给出了休眠时间的计算方法,即  $T_{\text{off}} = \nu \times T_r$ , 则该冗余节点总的休眠时间  $T_a$  可表示为

$$T_a = T_x \times T_{\text{off}} \quad (14)$$

式(14)表明该冗余节点向簇头发送数据的次数与其他非冗余的可靠节点相比少了  $T_x \times \nu$  次.由于节点类别动态性较强,因此实时调整各个节点连续担任簇头的最优次数不太现实.但当前簇头在其连续担任本地控制中心次数到达最优值时,可以统计出在每个轮回中与其进行通信的成员节点平均数目  $n_a$ . 为简化计算,在簇头切换前,当前簇头将参数  $n_a$  转发到成为候选簇头的节点,该节点利用式(14)计算其总的休眠时间,并将休眠时所节省的能量折算为担任簇头的工作次数  $f_x$ , 则该簇头在原有计算的连续担任簇头最优工作次数  $f_o$  的基础上增加  $f_x$  次.  $f_x$  可由下式计算得到:

$$\begin{cases} E_s = T_x \times \nu \times (E_{\text{cpu}} + E_{\text{amp}} \times d_1^2 + E_{\text{elec}})k \\ f_x = \frac{E_s}{\{(2n_a - 1)(E_{\text{elec}} + E_{\text{cpu}}) + E_{\text{amp}}[d_0^2 + (n_a - 1)d_1^2]\}k} \end{cases} \quad (15)$$

其中,  $k$  为数据包的二进制长度,  $E_{\text{elec}}$  (nJ/bit) 为射频电路的能耗系数,  $E_{\text{amp}}$  (nJ/bit/m<sup>2</sup>) 为电路的放大器能耗系数,  $E_{\text{cpu}}$  (nJ/bit) 为处理器能耗系数,  $d_0$  为下层簇头到上层簇头或基站的平均距离,  $d_1$  为簇头与其成员的平均距离,  $E_s$  为冗余节点休眠期间所节省的能量.由此可重新计算得到该节点担任簇头的最优连续工作次数  $f_{\text{on}}$  为

$$f_{\text{on}} = f_o + f_x \quad (16)$$

当该节点成为簇头时,通过增加其连续工作次数,不仅能有效平衡节点能耗,而且减少了上一个簇头记录和分析各冗余节点工作次数所带来的存储代价和复杂计算代价,降低了为平衡节点能耗而强行关闭某些具有覆盖重叠面积较少的可靠节点所增加覆盖盲区的风险.

#### 1.5 簇头数据融合的机制

以分簇构建的网络,簇头需要完成本簇成员数据的融合处理.近年来,很多研究者先后提出了基于概率、最大似然估计等对冗余数据进行融合的方法<sup>[12-13]</sup>.然而这些算法对于数据的分布模型依赖性强,对于不确定的数据分布,其融合精度得不到保障.文献[14]提出了扩展加权平均法,其数据融合的精度较最优加权法和最大似然估计法高.然而,该算法获取加权系数的计算方式在传感器网络中很难实现.

节点感知数据综合支持度反映了该节点数据与其他节点感知数据的关联性,综合支持度越高,说明该节点数据与其他节点数据的关联性越强;反之,关联性越弱.因此,簇头可以直接将各成员节点感知数据的综合支持度作为加权平均算法中对应的各个加权系数.文献[15]提出根据节点感知数据综合支持度,剔除综合支持度小的节点数据,对保留节点的原始数据再采用最小二乘法进行融合,最终融合数据的精度

比单独运用最小二乘法和极大似然法高。现对该融合方法改进, 设节点总数为  $n$ , 感知数据为  $x_i$  ( $i = 1, 2, \dots, n$ ), 设各节点感知数据综合支持度为  $\beta_i$  ( $i = 1, 2, \dots, n$ ), 则融合数据  $y$  可表示为

$$y = \frac{\sum_{i=1}^n x_i \times \beta_i}{\sum_{i=1}^n \beta_i} \quad (17)$$

现以文献 [15] 给出的实验数据进行验证, 感知数据为  $x_i$  ( $i = 1, 2, \dots, 10$ ) 分别为: 1.00, 0.99, 0.98, 0.97, 0.5, 0.65, 1.01, 1.02, 1.03, 1.5, 测量方差分别为: 0.05, 0.07, 0.1, 0.2, 0.3, 0.25, 0.1, 0.1, 0.2, 0.3。由式 (1)~(4) 计算得到各个传感器感知数据的综合支持度  $\beta_i$  ( $i = 1, 2, \dots, 10$ ) 为: 1.0000, 0.9943, 0.9781, 0.9437, 0.0702, 0.2429, 0.9893, 0.9781, 0.9437, 0.0702。根据计算获得的综合支持度, 文献 [15] 将节点编号为 5, 6, 10 的数据剔除, 对剩余节点数据采用最小二乘法融合得到的数据为 0.9994。在剔除相同的节点条件下, 采用式 (17) 计算得到的融合数据为

$$y = \frac{\sum_{i=1}^n x_i \times \beta_i}{\sum_{i=1}^n \beta_i} = \frac{6.82715}{6.8272} = 0.999993$$

文献 [16] 给出的 10 个传感器数据出现明显的波动, 采用该文作者所提方法得到的融合数据为 99.7791, 而采用本文式 (17) 得到的融合数据为 100.195, 更接近真实值 100.00。

对比之下, 采用式 (17) 所得到的融合数据比文献 [15–16] 给出的融合数据精度更高。现将文献 [15] 给出的所有传感器节点数据代入式 (17), 计算得到最后的融合数据为:

$$y = \frac{\sum_{i=1}^n x_i \times \beta_i}{\sum_{i=1}^n \beta_i} = \frac{7.125435}{7.2105} = 0.9882$$

显然, 剔除综合支持度较低的数据后的融合结果与全数据参与融合后的结果相比, 前者融合数据的精度明显优于后者。从上面的计算结果和文献 [15–16] 的结果可以看出: 剔除支持度低的数据有利于提高融合数据的精度, 并且采用综合支持度作为加权系数实现感知数据融合是可行的。

## 2 算法的实现

设普通节点总数为  $n$ , 在簇头查询和接收其成员感知数据的第一轮和第二轮后, 簇头由式 (4) 快速计算出各个成员的感知数据综合支持度  $\beta_i(1)$  和  $\beta_i(2)$ , 并初步将成员分为综合支持度低、中和高三类, 同时记录各节点 ID 和与之对应的综合支持度。由式 (10) 计算得到各个节点综合支持度增量的变化情况, 并根据控制规则表通知其成员在下一轮的工作状态。当前簇头连续工作次数到达最优次数时, 由候选簇头在下一轮中替换当前簇头, 当前簇头将各个节点感知数据的综合支持度发送到候选簇头, 候选簇头在下一个轮回中开始按照相同的方式进行计算和处理各类节点。

簇头在进行新一轮综合支持度及其增量的计算和节点分类时, 需要重新接收各个成员的感知数据至少两个轮回, 当然可以根据实际情况调整该轮回次数。由于存在冗余节点和冲突节点的休眠情况, 簇头在其连续工作次数未到达最优值时, 其能耗将远低于非休眠方式。采用计数器记录冗余节点

总的休眠次数, 并将休眠所减少的能耗折算到其担任簇头的次数, 通过增加其担任簇头的次数实现簇内节点能耗均衡。

由于冗余节点的调度方式特殊, 簇头在整个过程中需要统计每轮回与其通信的普通节点数目, 并在其连续工作次数到达最优值时需要将每轮活动的平均成员数目通知给候选簇头。而候选簇头有两类: 冗余节点或活动的非冗余节点。如果候选簇头为非冗余节点, 则其在前面的工作轮回中没有进行休眠, 其能量消耗为正常情况, 因此该节点不需要增加其连续担任簇头的次数; 相反, 如果该候选簇头为冗余节点, 则根据式 (15) 和式 (16) 重新计算得到其连续担任簇头的最优值, 并将该值广播给簇内节点以及相邻簇的当前簇头。

## 3 算法性能评估

为评估所提出的算法性能, 在 Matlab 环境下构建了验证场景: 第一部分仿真验证算法的有效性, 第二部分验证算法对于网络能耗和寿命的改善状况。在后面的仿真验证中, 使用相同的节点感知数据模型  $x_i(t)$ :  $x_i(t) = 20 + 15 \times \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} + wgn(1, 1, -5)$ 。其中,  $t$  为采样时间,  $wgn(1, 1, -5)$  代表强度为  $-5$  dB 的传感器节点测量值中的一个白噪声向量。由于测量噪声是传感器内部噪声和环境干扰等多种相互独立因素引起的, 各个传感器的测量噪声为相互独立的白噪声, 尽管各节点测量值中的白噪声强度有所不同, 为了计算方便, 统一设置为  $-5$  dB, 这并不影响节点感知数据综合支持度的分析。

### 3.1 簇内数据管理算法有效性验证

为简化计算, 将所有节点配置在半径  $R$  为 60 m 的一个圆形区域, 且所有节点处于一个簇中。设节点感知半径  $R_s$  为 20 m。各节点感知数据的样本方差为 100 次测试计算结果。节点总数分别为 100, 200, 300, 经过 100 轮测试, 节点综合支持度和相应节点的平均数目统计如表 3 所示。

表 3 节点综合支持度和相应的节点数目  
Table 3 The integrative supportabilities of all nodes and the numbers of the related nodes

	$n = 100$	$n = 200$	$n = 300$
$0 < \beta < 0.4$	3	12	22
$0.4 \leq \beta < 0.7$	11	46	76
$0.7 \leq \beta < 1$	86	142	202

尽管仿真无法研究感知数据偏差与能耗关系, 但从表 3 可以看出, 多数节点感知数据综合支持度  $\beta > 0.7$ 。

设网络服务质量的期望值为  $0.7 \sim 0.95$ , 设冗余节点数目为  $(n_h - n_r)/2$ , 即  $\alpha = 50\%$ , 在 100 轮的测试中, 其平均冗余节点数目的变化情况如图 1 所示。

图 1 中冗余节点数目尽管表现出起伏状态, 但随着服务质量期望值的增加, 冗余节点数目整体呈现减少的趋势。

### 3.2 网络寿命验证与分析

由于节点分类算法中感知综合支持度的计算量相对较大, 不适合对网络中所有节点进行综合分析。因此本文所提算法需与分簇构建的网络相结合, 综合支持度的计算只针对各个簇, 并由簇头负责管理该簇成员的感知数据。首先验证算法在一个固定簇的性能。设簇的形状为  $R = 30$  m 的圆形区域, 节点随机分布, 各节点初始能量为 0.5 J, 收发数据的所有数据包长度为 1000 bits,  $E_{\text{amp}} = 0.659$  nJ/bit/m<sup>2</sup>,  $E_{\text{elec}} = 50$  nJ/bit, 普通节点  $E_{\text{cpu}} = 7$  nJ/bit。由于簇头需要计算成

员的感知数据综合支持度,令簇头的  $E_{cpu} = 14 \text{ nJ/bit}$ . 设节点总数  $n = 50$ , 节点感知半径  $R_s = 10 \text{ m}$ . 簇头采用连续工作机制, 由于节点分布在一个固定的簇中, 簇头担任次序以节点编号为基准, 即节点编号越小, 则越早担任簇头. 为比较算法测试结果的一致性, 选取一组由随机函数产生的分布相对均匀的节点位置坐标, 并直接令编号为 1 的节点担任初始化簇头. 簇头及其成员的位置分布情况如图 2 所示.

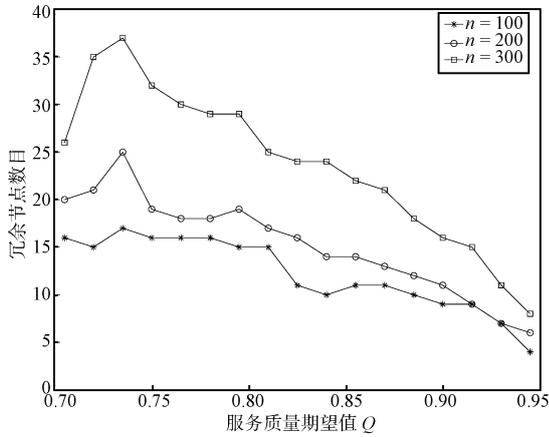


图 1 不同服务质量期望值所对应的冗余节点  
Fig.1 The numbers of redundant nodes with different expected values of  $Q$

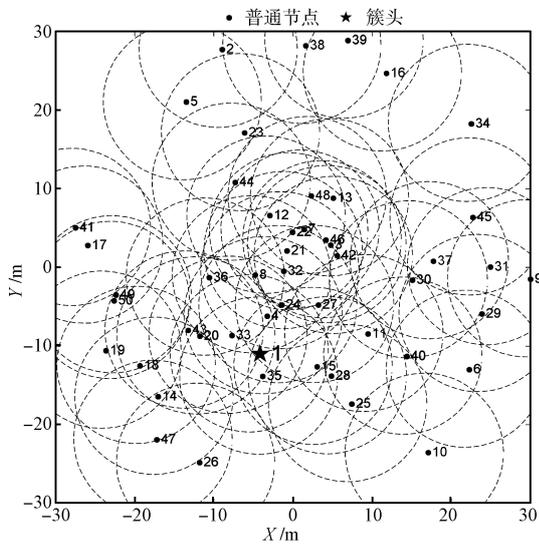


图 2 簇头及其成员的初始分布  
Fig.2 The initial distribution of the cluster heads and its members

图 2 中的由虚线所构成的圆形区域为各个节点的感知范围. 设该簇的簇头到中继簇的簇头平均距离  $d_0$  为  $50 \text{ m}$ , 并设节点间距离服从均匀分布, 由图 2 可知其距离最大值为  $30\sqrt{2} \text{ m}$ , 则簇头与成员的平均距离为  $15\sqrt{2} \approx 21 \text{ m}$ . 设  $Q = 0.8$ ,  $\varepsilon_1 = 0.4$ ,  $\varepsilon_2 = 0.85$ , 则  $n_r = 15$ . 由图 2 可直观看到在坐标原点附近存在大量的感知重叠范围, 如果这些节点为冗余节点, 网络运行中将关闭较多的节点. 设候选冗余节点数目所占百分比  $\alpha = 50\%$ , 且冗余节点必须是覆盖重叠范围较大的节点. 休眠节点的休眠周期  $\nu = 2$ , 且休眠与活动周期相同. 当网络中第一个节点死亡时, 网络中冲突节点数目和冗

余节点数目如图 3 所示. 各节点作为冗余节点的总次数如图 4 所示.

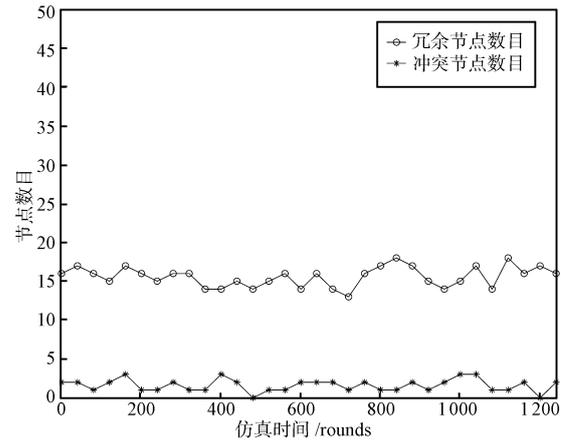


图 3 在网络第一个节点死亡前的冲突节点和冗余节点数目  
Fig.3 The numbers of unreliable nodes and redundant nodes before the first node dead

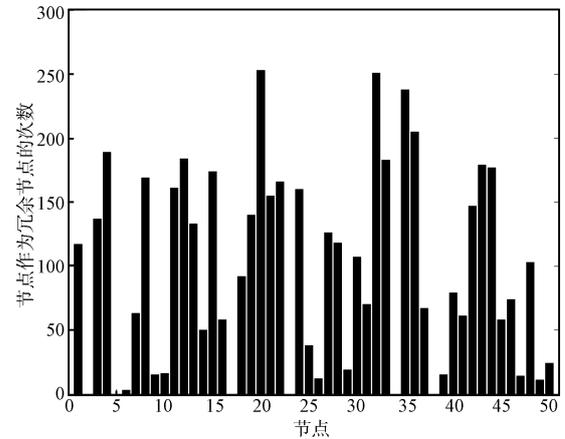


图 4 各节点作为冗余节点的总次数  
Fig.4 The total working times of each nodes acting as the redundant node

由于  $\varepsilon_1$  的取值较小, 图 3 中的冲突节点很少, 在整个仿真过程中其平均冲突节点有 2 个; 同时,  $\varepsilon_2$  和服务质量期  $Q$  取值较小, 在整个过程中冗余节点平均数目为 16, 占总节点数目的 32%. 对于一个规模较小的网络, 能周期性调度占总节点数目 32% 的冗余节点, 簇头信道利用率将会显著提高, 同时簇内的能量效率也将极大改善. 在多次仿真结果中, 第一个节点死亡时间在第 1220~1238 rounds 范围, 具体死亡节点出现在图中编号为 5, 9, 10, 26, 34 这几个节点中. 在图 2 中的节点位置分布中, 编号为 9, 10, 26, 34 节点处于边缘部分, 当簇头位于相对远的位置时, 这些节点在向簇头发送数据需要较多的能量; 同时, 这几个节点感知覆盖重叠面积较少, 特别是节点 5; 而 10, 34 这两个节点在整个运行过程中成为冗余节点的概率相对较小, 从图 4 中可以看到这些节点成为冗余节点的总次数较小, 其能量消耗较其他节点要快得多, 因此很快进入失效状态.

现不改变  $\varepsilon_1, \varepsilon_2$  取值, 将  $Q$  的取值范围设定为  $0.8 \sim 0.95$ , 并调整冗余节点数目与候选冗余节点数目的比值  $\alpha$ . 在

不同条件下第一个节点死亡时间的改变情况如图 5 所示.

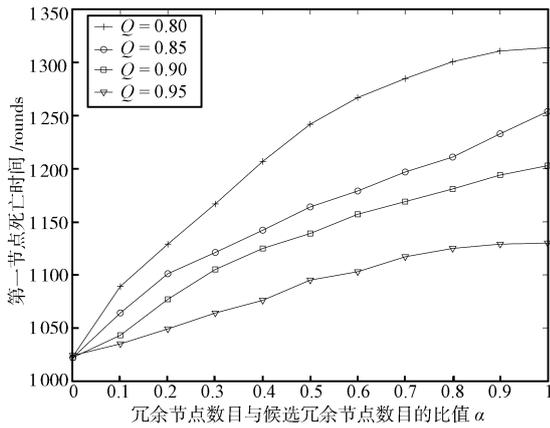


图 5 不同 Q 值下第一个死亡节点的时间  
Fig. 5 The time of the first node dead with the different values of Q

在图 5 中, 当冗余节点数目的百分比  $\alpha$  为 0 时, 实际上只关闭了冲突节点, 但由于冲突节点数目较少, 簇内节点工作情况与不考虑节点分类情况下相似, 因此第一死亡节点时间非常接近. 而当冗余节点数目不为零时, 随着服务质量期望值的提高, 冗余节点数目下降, 即参数  $\alpha$  减小, 则簇内活动节点数目增加, 导致簇内能耗加大, 对应簇的寿命 (第一个死亡节点时间) 缩短.

从上面的验证结果可以看出, 在给定的节点密度条件下, 随着服务质量期望值的提高, 网络寿命缩短. 但如果增加节点密度, 在相同区域或同一个簇, 其冗余节点数目增加的可能性较大. 现设 500 个节点分布在半径  $R = 100\text{ m}$  的圆形区域, 采用文献 [11] 提出的分簇算法实现网络拓扑管理. 节点的能耗参数与上面设定相同, 簇头的  $E_{cpu} = 14\text{ nJ/bit}$ , 节点感知半径可调, 有效范围为  $[10\text{ m}, 20\text{ m}]$ , 网络服务质量期望值  $Q$  最小为 0.8. 数据包长度为 1000 bits, 广播消息长度为 200 bits, 基站的坐标为  $(0, 0)$ . 设  $\epsilon_1 = 0.4$ ,  $\epsilon_2 = 0.85$ . 设服务质量期望值  $Q = 0.8$ , 节点感知半径  $R_s = 10\text{ m}$ , 节点分类和不分类所对应的网络寿命比较结果如图 6 所示.

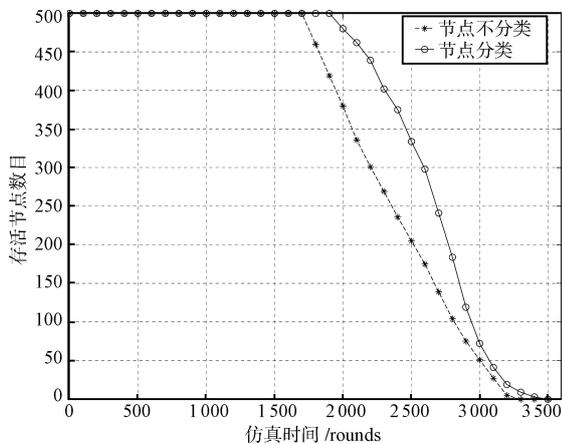


图 6 网络寿命对比结果  
Fig. 6 The network lifetimes with different mechanisms

在图 6 中, 采用节点分类后的网络, 第一个死亡节点的

时间较没有分类的情况明显延长了, 而且在仿真时间  $t$  处于 2000~2500 rounds 时, 分类后节点死亡变化情况比较平缓, 原因是处于最外围的簇存活节点数目仍然较多且具有较多的剩余能量, 导致冗余节点数量相对较多, 簇内通信能耗率相对较小. 但仿真时间在 2500~3000 rounds 的过程中, 分类后的节点死亡速度非常快, 其原因是采用分类算法的簇头计算其成员的综合支持度消耗较多能量, 而处于这段时间的各个簇头剩余能量本来不多, 因此综合支持度的计算加速了这些节点的死亡速度. 针对不同的服务质量期望值  $Q$  和不同的节点感知半径  $R_s$ , 网络寿命 (第一节点死亡时间) 对比情况如图 7 所示.

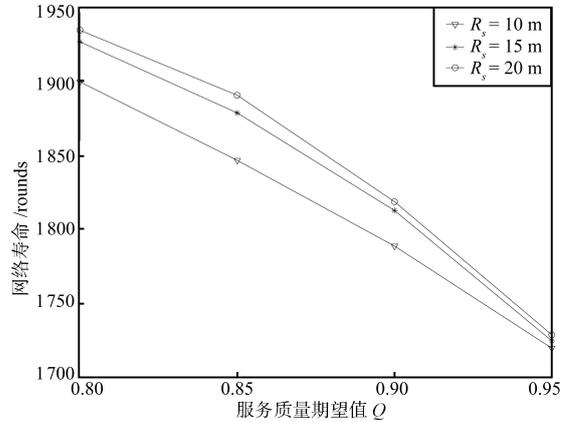


图 7 不同条件下的网络寿命比较  
Fig. 7 The network lifetimes under the different conditions

图 7 表明, 在服务质量期望值相同且较小的条件下, 节点感知半径越大, 网络寿命越长. 由式 (8) 和式 (13) 可知, 在一个给定区域, 感知半径越大, 所需的最少可靠节点数目越少; 同时各候选冗余节点覆盖重叠面积越大, 因此划分的冗余节点数目越多, 则簇内平均通信能耗率下降, 网络寿命延长. 但当服务质量期望值较大 ( $Q = 0.95$ ) 时,  $n_r$  取值较大, 则冗余节点数目急剧下降甚至没有; 同时, 由于  $\epsilon_1$  取值为 0.4, 冲突节点数目较少, 从而导致网络寿命都接近节点没有分类的情况. 但从另一方面考虑, 如果网络节点分布密度很大, 即使服务质量期望值较大, 仍然可能存在较多的冗余节点, 或者说当  $\epsilon_1$  取值较大时, 冲突节点数目增加, 则节点分布较多的簇其总体能耗依然会显著降低, 对应的网络寿命将会提高.

### 4 结论

本文提出了基于感知数据关联性的簇内数据管理算法, 该算法建立在节点感知数据的有效性基础上, 利用误差函数和模糊关联函数, 获取各个节点综合支持度, 由此将节点分为可靠节点、补充节点和冲突节点, 并结合各节点综合支持度的变化情况给出了休眠节点调度规则. 在可靠节点中, 以服务质量期望值为条件, 结合候选冗余节点感知覆盖重叠面积的大小确定出冗余节点, 给出了冗余节点的调度机制, 即各冗余节点记录其休眠次数并折算为应当增加担任簇头的次数, 从而为簇内节点能耗均衡提供了可能, 并降低了簇头的计算能耗. 最后, 通过设置不同的参数, 仿真验证了节点分类算法的可行性; 仿真结果表明该簇内数据管理算法能有效降低网络能耗, 并能显著延长网络寿命.

## References

- 1 Tian D, Georganas N D. A coverage-preserving node scheduling scheme for large wireless sensor networks. In: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications. Atlanta, USA: ACM, 2002. 32–41
  - 2 Lu J, Wang J S, Suda T. Scalable coverage maintenance for dense wireless sensor networks. In: Proceedings of the 3rd Annual IEEE Communications Society Conference on Sensor and Ad hoc Communications and Networks. Virginia, USA: IEEE, 2002. 651–660
  - 3 Liu Ming, Cao Jian-Nong, Zheng Yuan, Chen Li-Jun, Xie Li. Analysis for multi-coverage problem in wireless sensor networks. *Journal of Software*, 2007, **18**(1): 127–136  
(刘明, 曹建农, 郑源, 陈力军, 谢立. 无线传感器网络多重覆盖问题分析. 软件学报, 2007, **18**(1): 127–136)
  - 4 Gao Y, Wu K, Li F L. Analysis on the redundancy of wireless sensor networks. In: Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications. San Diego, USA: ACM, 2003. 108–114
  - 5 Li Guo-Hua, Liu Bao-Ling, Shen Shu-Qun. Research on data de-redundance for transmission in wireless sensor networks monitored the physical area. *Microelectronics and Computer*, 2005, **22**(9): 134–136  
(李国华, 刘宝玲, 沈树群. 无用于区域监测的无线传感器网络数据去冗余研究. 微电子学与计算机, 2005, **22**(9): 134–136)
  - 6 Yu Chun-Lai, Xu Hua-Long, Huang Shi-Qi. A relation matrix method for multisensor data fusion. *Aeronautical Computer Technique*, 2005, **35**(1): 23–26  
(禹春来, 许化龙, 黄世奇. 基于关系矩阵的多传感器数据融合方法. 航空计算技术, 2005, **35**(1): 23–26)
  - 7 Gao Fang-Wei, Liu Gui-Xi, Wang Lei, Zhang Jing. Multi-sensor fusion based on the support matrix and optimal weight. *Journal of Projectiles, Rockets, Missiles and Guidance*, 2007, **27**(4): 284–287  
(高方伟, 刘贵喜, 王蕾, 张靖. 基于支持度矩阵的一种多传感器融合方法. 火箭与制导学报, 2007, **27**(4): 284–287)
  - 8 Liu Jian-Shu, Li Ren-Hou, Liu Yun-Long, Zhang Zhen-Yao. Multi-sensor data fusion based on correlation function and fuzzy integration function. *Systems Engineering and Electronics*, 2006, **28**(7): 1006–1009  
(刘建书, 李人厚, 刘云龙, 张贞耀. 基于相关性函数和模糊综合函数的多传感器数据融合. 系统工程与电子技术, 2006, **28**(7): 1006–1009)
  - 9 Koskinen H. On the coverage of a random sensor network in a bounded domain. In: Proceedings of the 16th International Teletraffic Congress Specialist Seminar on Performance Evaluation of Wireless and Mobile Systems. Antwerp, Belgium: IAC, 2004. 11–18
  - 10 Liu Min-Hua, Xiao De-Yun. Method of multisensor data classification based on entropy. *Control and Decision*, 2006, **21**(4): 410–414  
(刘敏华, 萧德云. 基于信息熵的多传感器数据分类方法. 控制与决策, 2006, **21**(4): 410–414)
  - 11 Xiang M, Shi W R, Jiang C J, Zhang Y. Energy-efficient clustering algorithm for maximizing lifetime of wireless sensor networks. *AEU — International Journal of Electronic and Communications*, 2010, **64**(4): 289–298
  - 12 Wonjang B, Satish B. Optimal  $m$ -ary data fusion with distributed sensors. *IEEE Transactions on Aerospace and Electronic Systems*, 1995, **31**(3): 1150–1152
  - 13 Zhou Y F, Leung H, Yip P C. An exact maximum likelihood registration algorithm for data fusion. *IEEE Transactions on Signal Processing*, 1997, **45**(6): 1560–1573
  - 14 Tang Jin, Zhang Wen-Jie, Gao Yan, Fu Ming-Yu, Cai Zi-Xing. Fusion of different accuracy redundant data. *Acta Automatica Sinica*, 2005, **31**(6): 934–942  
(唐瑾, 张闻捷, 高琰, 付明玉, 蔡自兴. 不同精度冗余数据的融合. 自动化学报, 2005, **31**(6): 934–942)
  - 15 Liu Jian-Shu, Li Ren-Hou, Chang Hong. Multi-sensor data fusion based on correlation function and least square. *Control and Decision*, 2006, **21**(6): 714–717  
(刘建书, 李人厚, 常宏. 基于相关性函数和最小二乘的多传感器数据融合. 控制与决策, 2006, **21**(6): 714–717)
  - 16 Luo Ben-Cheng, Yuan Kui, Chen Jin-Long, Zhu Hai-Bing. Uncertainty analysis based dynamic multi-sensor data fusion. *Acta Automatica Sinica*, 2004, **30**(3): 407–415  
(罗本成, 原魁, 陈晋龙, 朱海兵. 一种基于不确定分析的多传感器信息动态融合方法. 自动化学报, 2004, **30**(3): 407–415)
- 向 敏 博士, 重庆邮电大学讲师. 主要研究方向为无线传感器网络和嵌入式技术. 本文通信作者. E-mail: xiangmin@cqupt.edu.cn  
(XIANG Min Ph.D., lecturer at Chongqing University of Posts and Telecommunications. His research interest covers wireless sensor networks and embedded technique. Corresponding author this paper.)
- 石为人 重庆大学自动化学院教授. 主要研究方向为智能控制与智能管理, 无线传感器网络及其应用和嵌入式系统. E-mail: wrs@cqu.edu.cn  
(SHI Wei-Ren Professor at the College of Automation, Chongqing University. His research interest covers intelligence control and management, wireless sensor networks, and embedded system.)