基于通信与感知覆盖的WSNs节点调度算法

杨 訸 汪文勇 唐 勇

(电子科技大学计算机科学与技术学院 成都 611731)

摘 要 在随机部署的无线传感器网络中通常包含覆盖与通信冗余节点,这些节点不仅会造成大量的能量浪费,同时影响网络的性能。因此,如何对网络中的覆盖与通信冗余节点进行有效的调度是无线传感器网络研究的一个重要课题。提出了一种基于蜂窝模型的分布式节点调度算法(RCSC)。在蜂窝结构的基础上,RCSC算法通过添加"桥梁节点"和填补"空洞"来进一步优化工作节点集,使得整个网络达到全"通信覆盖"和全"感知覆盖"。最后,RCSC算法结合LEACH协议,对网络中的节点进行动态调度。经试验仿真证明,由RCSC算法构建出的网络拓扑中的工作节点数少且稳定,从而减少了由于冗余数据通信导致的额外能量消耗,延长了网络生存时间。

关键词 无线传感器网络,感知覆盖,通信覆盖,蜂窝模型

中图法分类号 TP393 **文献标识码** A

Node Scheduling Algorithm Based on Communication and Sensing Coverage in WSNs

YANG He WANG Wen-yong TANG Yong

(School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract In the randomly deployed wireless sensor networks (WSNs), there may always contain some redundant nodes of coverage and communication, which would not only cause huge waste of energy, but also affect the network's service quality. Therefore, how to effectively schedule the redundant nodes becomes a heated issue in WSNs. This paper proposed a distributed algorithm for active nodes with communication and sensing coverage in the randomly deployed WSNs (RCSC). Based on the cellular structure, RCSC optimizes the working set of nodes by adding "bridge nodes" and filling the "sensing hole", in order to ensure "communication coverage" and "sensing coverage" performance of the whole network to be achieved. Finally, combined with LEACH protocol, RCSC can schedule all the nodes dynamically. Simulation shows that in the network topology constructed by RCSC, working nodes are fewer while the network is stable, as a result, it extends the network lifetime by decreasing the extra energy consumption caused by random communications among nodes.

Keywords WSNs, Sensing coverage, Communication coverage, Cellular structure

1 引言

在现有的无线传感器网络(wireless sensor network, WSNs)的研究中,很多都是致力于解决传感器网络的部署和监测及覆盖与连接的关系等方面的问题。而这些问题的核心思想都是与覆盖问题有关的,它要求目标区域内的每一点至少被一个无线传感器节点覆盖,称为"感知覆盖",并同时保证网络内各节点间的通信连通性,称为"通信覆盖"。无线传感器网络很多是用于在野外森林等特殊环境下侦测信息的,其节点往往是被高密度部署,因此某些区域往往会同时被多个节点覆盖,造成无线信号冲突频繁,从而对网络性能、节点能耗带来负面影响。因此,对网络中传感器节点进行合理有效的拓扑控制是很有必要的,它能够在保证覆盖性能的前提下,减少工作节点数,并同时减少冗余数据通信导致的额外能量消耗,延长网络生存时间。总的来说,解决该问题的关键是使用尽可能少的节点来保持对目标区域的全覆盖和全连通。

目前,在针对WSN覆盖问题的节点调度方面,已经有很多文献做了相关的工作[1-10]。

一种是基于规则多边形的最优部署模式,即当传感器节点通信范围 Rc 和感知半径 Rs 的比值大于且等于 $\sqrt{3}$ ($Rc/Rs \geqslant \sqrt{3}$)时,部署节点在正六边形的中心具有最优的覆盖和连通效果。文献[1,2]在节点确定性部署情况下,研究了定点节点的全覆盖和 k($k=1,2,\cdots,6$)连通的最优模型,并进一步提出了 Rc 和 Rs 在任意比例下,部署节点达到全覆盖和 k ($k=1,2$)连通的最优方案。而当今更多的研究是在节点随机性部署的情况进行的,定点部署的最优模型为随机性部署研究提供了一个基本模型。文献[3]以全覆盖和 $k=1$ 连通为基本模型,也就是以蜂窝结构覆盖为理论依据,提出了从基点或基边出发,采用广播和时延的机制来选取近似最优连通覆盖集的BPS算法。文献[4]提出了另一种基于蜂窝模型的网络构造,

到稿日期:2012-09-10 返修日期:2013-01-30

杨 訸(1990—),女,硕士生,主要研究领域为无线传感器网络,E-mail:yanger9006@126.com;汪文勇(1967—),男,博士,教授,博士生导师,主要研究领域为下一代互联网管理与测量、下一代互联网应用、无线传感器网络;唐 勇(1973—),男博士,副教授,主要研究领域为无线传感器网络,计算机网络。

并使用理想情况下节点在六边形模型中所处的位置作为“战略点”，来确定实际情况中的工作节点位置。具体思想是将已选工作节点带有自己位置和上一跳工作节点位置的信息广播给周边节点，收到消息的节点根据消息中的位置信息计算战略点与自己的欧式距离，节点按与战略点欧式距离最近原则自荐为新的工作节点。而 DBC 算法[5] 对之前算法进行了改进，利用周边节点位置信息和统一求取战略点的方式，减少了蜂窝结构的扭曲。

另一种对于节点的部署并没有基于规则多边形的最优部署模型，而是提出一套新的判断机制，即每个节点根据周边节点的信息判断自身的工作状态。Tian 等人[6-8] 提出了判断节点是否冗余的方法，即判断该节点的感知覆盖区域是否已被其它工作节点覆盖。它通过感知范围内的邻居节点的位置信息来计算邻居节点对该节点圆周角的贡献，如果邻居节点的圆周角覆盖达到 2π，则说明该节点为冗余节点。节点判断自己为冗余节点后，则转为休眠状态，否则醒来成为工作节点。针对判断节点冗余，Huang 等人[9,10] 提出了一种更为精确的判断方法。它通过计算邻居节点的覆盖圆周覆盖度判断一个节点的探测范围是否被覆盖。其主要思想为：对于节点 i，若任意一个邻居节点 j 在其探测范围内的圆弧覆盖度不小于 2，则节点 i 为冗余节点。

本文将在前人研究的基础上提出一种新的算法：RCSC 算法。首先，节点基于二维平面的覆盖最优模型[11] 构建初始覆盖集，然后通过添加“桥梁节点”和填补“空洞”(感知空白区域)来优化初始覆盖节点，使得整个网络达到全连通覆盖和全感知覆盖。其中的空洞填补机制是基于 Huang 算法[9,10] 提出的。最后结合 LEACH 协议，将整个网络区域的节点动态调度起来，使节点的能耗尽可能地均匀，从而延长网络生存时间。

本文第 2 节对问题进行详细分析；第 3 节针对提出的问题进行求解；第 4 节呈现了整体的算法描述；第 5 节给出了仿真结果并加以分析；最后对本文进行了总结。

2 问题分析

本文首先假设网络中的所有传感器节点的通信模型和感知模型是圆盘形状，其通信或感知范围是以该节点为中心、以通信范围 Rc 或感知范围 Rs 为半径的圆形区域。

在二维无线传感器网络中，覆盖问题可描述为：“如何使用最少的传感器节点感知覆盖整个无线传感器网络，并同时使得这些节点通信?”。根据此描述，本文将 “覆盖”分为感知覆盖和通信覆盖。感知覆盖表示目标区域内的每一点都至少被一个无线传感器节点覆盖，如图 1 所示，11 个无线传感器节点(黑色点)分别以自己的感知范围(灰色圆盘)覆盖了整个目标区域(黑色边界线内)。而通信覆盖表示目标区域中的工作节点间都能相互通信，如图 2 所示，11 个无线传感器节点分别以自己的通信范围(灰色圆盘)覆盖了整个目标区域，其中两节点由红线相连表示这两个节点能够通信。

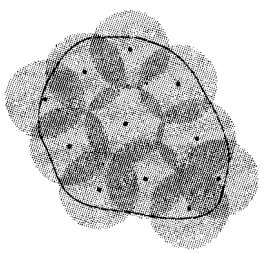

图 1 节点感知覆盖目标区域

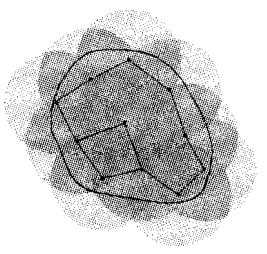

图 2 节点通信覆盖目标区域

为了使目标区域达到全“感知覆盖”和全“通信覆盖”，本文基于二维平面的覆盖最优模型来构建节点覆盖集，而该模型通常被称为蜂窝模型。该最优模型如图 3 所示，其中正六边形的中心到顶点的距离为 Rs，当 $Rc/Rs \geqslant \sqrt{3}$ 时，部署节点在正六边形的中心具有最优的覆盖和连通效果。

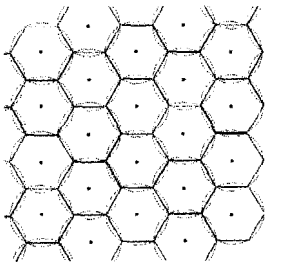

图 3 无线网络的最优覆盖模型

但本文研究的是节点随机性部署的网络，在该网络中，把每个节点布置在理想情况下的位置是不现实的。所以，基于蜂窝模型来选取工作节点的最终网络拓扑并不能使整个网络实现“全通信覆盖”和“全感知连通”。具体出现的问题如下：

1)在节点随机布置的情况下，如何选取工作节点使得初始节点覆盖集所形成的网络拓扑尽量类似蜂窝模型。

2)由于节点布置的随机性，也就是节点所处地理位置的不确定性，可能会造成初始节点在扩散过程中找不到合适的节点，从而导致节点扩散的中止。这样选出的初始节点覆盖集不能达到整个网络的连通。

3)由于工作节点构成的蜂窝模型会出现扭曲，可能不会使区域中的每个点都被覆盖，即会有空洞的出现。而空洞的存在会严重影响到整个网络的覆盖度，使整个网络不能达到全覆盖。

4)在定下首轮工作节点后，还需要使整个网络区域节点动态调度起来，即在工作节点失效后，需要休眠节点醒来作为工作节点。同时，所有节点的调度还需要使整个网络节点的能耗尽量均匀。

为了阐述方便，根据以上分析，本文将覆盖问题划分为 4 个小问题：

问题 A　初始节点覆盖集如何构建?

问题 B　初始节点集是否能达到整个网络的“通信覆盖”? 如果不能，如何解决?

问题 C　初始节点集是否能达到整个网络的“感知覆盖”? 如果不能，如何填补空洞?

问题 D　如何使整个网络区域节点动态调度?

3 问题求解

3.1 求解问题 A

本文采用同 DBC 算法[5] 相类似的方法，即第一个六边形的一条边确定了以后，整个网络的战略点位置都因此确定，不受实际中节点分布的影响。采用此种节点选取方法，可以减少这个初始节点集所构成的六边形的扭曲程度。

基于 DBC 中的方法，本算法结合蜂窝模型(见图 1)，提出了适合本文的节点的选取过程(见图 4)。通过初始发起节点 A 找到第二轮发起节点 $B1$、$B2$、$C1$、$C2$、$D1$ 和 $D2$。第二轮发起节点与初始发起节点可以继续向外扩散，如节点 $B2$ 结合节点 A 确定出将要扩散节点的位置，并找到节点 $B21$、$B22$ 和 $B23$。同理可得，任一节点结合上一跳节点都可能确定向外扩散的节点，从而覆盖整个网络。

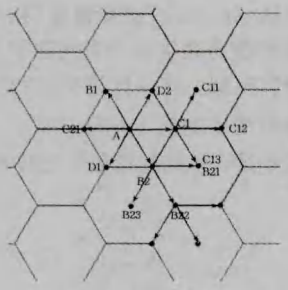

图 4　初始节点集选取

3.2　求解问题 B

在初始节点覆盖集选取的过程中，新的节点是通过上两跳工作节点的位置信息得到的。而在随机性部署网络中，某个工作节点结合上一跳节点所计算出的下一跳节点的位置可能在实际网络中根本找不到实际点的存在。这种情况称为节点扩散中止(见图 5)。

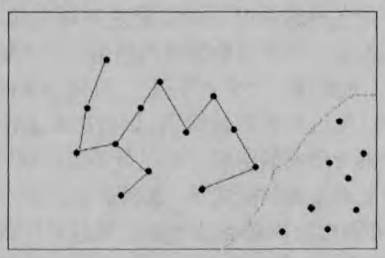

图 5　节点扩散中止

为避免节点扩散中止的情况，本文提出了“桥梁节点”的概念，如图 6 所示。在初始节点集的连通范围内随机选取一点作为新的工作节点，使得节点集能得以继续延伸，从而达到整个网络的“通信覆盖”。因此，桥梁节点就是为了使初始节点覆盖集能够继续扩散而成为工作节点的节点。

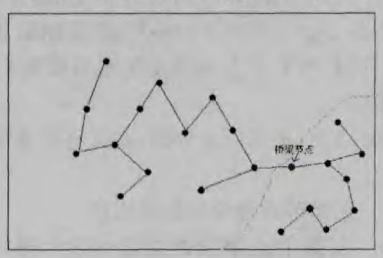

图 6　“桥梁节点”示意图

而为了使最终选出的工作节点尽量少，本算法对“桥梁节点”的选取制定了如下规则：

规则 1　某个未被初始节点集连通的节点在查看自己的连通范围内的节点后，如果存在工作节点，则选择该工作节点为“桥梁节点”。

规则 2　某个未被初始节点集连通的节点在查看自己的连通范围内的节点后，如果存在初始节点集连通范围内的非工作节点，则选择该工作节点为“桥梁节点”。

3.3　求解问题 C

为了使整个网络达到全“感知覆盖”，非工作节点需要判断自己是否为冗余节点。如果该节点是冗余节点，则休眠；如果不是，则醒来填补空洞。本节将问题 C 再分为如下两个小问题：

问题 C1　节点何时醒来判断自己是否冗余？

问题 C2　如何判断自己是否为冗余节点？

3.3.1　求解问题 C1——确定节点回退时间 T

非工作节点以随机的顺序判断冗余，若需要填补空洞，则成为工作节点，这样可能造成很大的节点浪费。对一个空洞的填补会由于节点填补的顺序不同而使节点数目不同。如图 7 所示，节点 1、2、3、4 依次判断自己未被完全覆盖后醒来填补空洞。但是，存在另一种更优的情况就是，如果图 7 中第 4 个醒来的节点最先醒来，即图 8 中标号为 1 的节点最先醒来，那么其它 3 个节点则可以不再填补这个空洞。

图 7　节点填补空洞情况 1

图 8　节点填补空洞情况 2

为解决上述问题，RCSC 算法给每个判断自己是否冗余的节点确定了自身的回退时间 T，即当 T 结束后，节点再进行冗余判断。

空洞最可能存在于几个工作节点的中间位置，即该节点到几个工作节点的标准差最小的地方。本文将节点醒来的先后顺序与其所处位置相关联，即每个节点到其连通范围内的工作节点的距离的标准差越小，那么这个节点的回退时间就越短。在分布式计算中，每个节点是并行计算自己的回退时间的，所以在一个特定区域内，第一个醒来的节点一般都能覆盖此空洞。

设：节点 i 连通范围内的工作节点数为 n。

节点 i 的回退时间：

$$T_i=\begin{cases}\alpha_{ij}=\dfrac{dist_{ij}}{Rs}\\ \bar{\alpha}=\dfrac{\sum_{k=1}^{n}\alpha_{ik}}{n}\\ \beta_i=\sqrt{\dfrac{\sum_{k=1}^{n}(\alpha_{ik}-\bar{\alpha})}{n}}\\ T_i=\beta_i * T_0\end{cases}\tag{1}$$

式中，$dist_{ij}$ 为节点 i 与节点 j 的欧氏距离。

3.3.2　求解问题 C2——节点的冗余判断

本算法是基于 Huang 算法[9,10]，它通过计算邻居节点的覆盖圆周覆盖度判断一个节点的探测范围是否被覆盖。但 Huang 算法只适用非边界节点的冗余判断，如果边界节点的冗余情况不单独判断，而按照非边界节点的判断方法来判断，那么所有的边界节点都不会被全覆盖，从而所有边界节点都会醒来而大大增加工作节点的数量。

对于边界节点，RCSC 算法在原算法上做了改进。本算法对边界节点的冗余判断分两部分：感知范围内区域的判断和边界线的判断。只有当两部分都被判断为冗余区域后，该边界节点才被定为冗余节点。对于感知范围内区域的判断，本文引入“虚拟节点”的概念。通过此方法，边界节点的判冗余方式可以与非边界节点的相同。而对于边界线的覆盖判断，本文进一步简化，将其近似为判断边界线中点是否被覆盖。

1)虚拟节点的确定

虚拟节点的创建见图 9，图中右边黑色圆为边界节点的

感知范围，左边灰色圆为该节点的虚拟节点的虚拟感知范围。

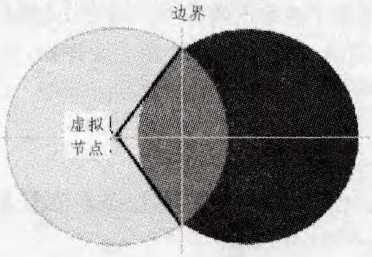

图 9　虚拟节点

设：边界节点为(x,y)，虚拟节点为(x',y)，(x,y')。以下为边界节点与其虚拟节点的对应情况：

$$\text{当}(x,y)\begin{cases}x\subseteq[0,R_s]\cup[AreaR-Rs,AreaR]\\ y\subseteq[Rs,AreaR-Rs]\end{cases}\xrightarrow{\text{虚拟节点}}(x',y)$$

$$\text{当}(x,y)\begin{cases}x\subseteq[Rs,AreaR-Rs]\\ y\subseteq[0,Rs]\cup[AreaR-Rs,AreaR]\end{cases}\xrightarrow{\text{虚拟节点}}(x,y')$$

$$\text{当}(x,y)\begin{cases}x\subseteq[0,Rs]\cup[AreaR-Rs,AreaR]\\ y\subseteq[0,Rs]\cup[AreaR-Rs,AreaR]\end{cases}\xrightarrow{\text{虚拟节点}}(x',y)\text{和}(x,y')$$

其中x'和y'的公式：

$$x'=\begin{cases}-x, & x\subseteq[0,Rs]\\ 2*AreaR-x, & x\subseteq[AreaR-Rs,AreaR]\\ x, & \text{others}\end{cases}$$

$$y'=\begin{cases}-y, & y\subseteq[0,Rs]\\ 2*AreaR-y, & y\subseteq[AreaR-Rs,AreaR]\\ y, & \text{others}\end{cases}\qquad(2)$$

其中，$AreaR$为网络区域的大小，即网络目标区域为$AreaR*AreaR$。

2)边界线中点的确定

边界线中点的创建见图 10。如果此点在某个工作节点的感知范围内，则近似为该边界线被覆盖。

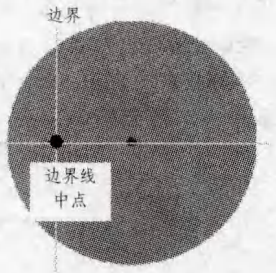

图 10　边界线中点

设：边界节点为(x,y)，边界线中点节点坐标为(x_m,y_m)

$$x_m=\begin{cases}0, & x\subseteq[0,Rs]\\ AreaR, & x\subseteq[AreaR-Rs,AreaR]\\ x, & \text{others}\end{cases}$$

$$y_m=\begin{cases}0, & y\subseteq[0,Rs]\\ AreaR, & y\subseteq[AreaR-Rs,AreaR]\\ y, & \text{others}\end{cases}$$

3.4　求解问题 D

当工作节点选取好后，节点的调度可与经典路由协议相结合。由于 RCSC 的工作节点调度机制与 LEACH[12] 有类似的网络生存时间划分，因此本文将结合 LEACH 协议来进一步完善 RCSC 算法。

LEACH 协议的这种生存时间划分与 RCSC 算法的时间划分非常类似，因此可以将 RCSC 算法的调度阶段插入 LEACH 协议的簇建立阶段之前。这样在每个时间片初期，首先由休眠机制进行节点调度，确定哪些节点进入休眠，哪些节点进入活跃状态。然后活跃节点进入 LEACH 协议的簇建立阶段，簇形成后再进入稳定工作阶段。RCSC 算法与 LEACH 协议相结合后的网络生存时间划分如图 11 所示。

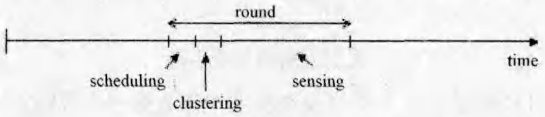

图 11　RCSC+LEACH 协议每轮时间划分

4　算法描述

假设网络区域形状为矩形，本算法执行的起始节点位于网络分布的拓扑图中心。RSCS 算法分为 3 部分，见图 12，分别如 4.3—4.5 节所述，而 4.6 节为 RCSC 算法结合 LEACH 协议的描述。

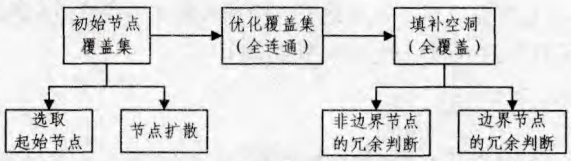

图 12　RCSC 算法的步骤

4.1　符号说明和名词解释

本算法将用到的名词和符号如表 1、表 2 所列。

表 1　名词解释

undecided 节点	节点初始状态
on_duty 节点	工作节点，完成对环境的感知与数据的传输
sleep 节点	休眠节点
ready_to_sleep 节点	只接收消息
ready_on_duty 节点	接收消息和发送询问消息
战略点	理想情况下节点在蜂窝模型中所处位置
父战略点	节点的上一跳工作节点的战略点
子战略点	节点扩散的下一跳节点的战略点
空洞	没有被任何工作节点覆盖的区域
冗余节点	节点的感知覆盖区域已经被其它工作节点覆盖

表 2　符号说明

id	节点自身的 ID 号
Rs	节点的感知半径
Rc	节点的通信半径
A、B、C…	节点标号
N(A)	节点 A 的全部一跳邻节点组成的集合
$dist_{ij}$	节点 i 和节点 j 的欧式距离
a、b、c…	战略节点标号
thrd	工作节点允许存在其它工作节点的最大区域半径
adb	战略点虚拟边境外扩大小

4.2　基本假设

• 算法起始节点位于网络区域中心，其他节点随机散布在二维空间的网络区域中；

• 节点的通信范围和感知范围都是以节点为中心、Rc 或 Rs 为半径的圆；

• 所有节点具有 GPS 系统，地理位置已知。

4.3　构造初始节点集

本节将初始节点集的选取算法分为发起节点的选取和节点的扩散。

4.3.1　选取发起节点

初始节点选取如图 13 所示。

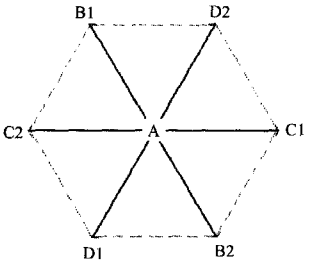

图 13　初始节点选取

(1)发起节点 A 将自身地理位置作为第一个战略点 a，在 $\sqrt{3}R_s$ 中找距其最远的邻节点 $B1$(距离为 $dist_{AB1}$)作为 ready_on_duty 节点，并利用相似三角形原理，依式(4)求解在此方向上的子战略点 $b1$，将战略点 a、战略点 $b1$ 的地理位置信息告知节点 $B1$；

$$\frac{dist_{AB1}}{\sqrt{3}Rs}=\frac{x_A-x_{B1}}{x_A-x_{b1}}=\frac{y_A-y_{B1}}{y_A-y_{b1}} \tag{4}$$

(2)将 $B1$ 作为端点，A 作为中点，依式(5)求解子战略点 $b2$，并在节点 A 的一跳邻居节点中选取距离子战略点 $b2$ 最近的节点作为 ready_on_duty 节点 $B2$；

$$\begin{cases}x_{b2}=2x_a-x_{b1}\\ y_{b2}=2y_a-y_{b1}\end{cases} \tag{5}$$

(3)结合 a、$b1$，利用余弦定理，依式(6)求解出子战略点 $c1$、$d1$；

$$\begin{cases}\dfrac{2(\sqrt{3}Rs)^2-[(x_{b1}-x_{c1})^2+(y_{b1}-y_{c1})^2]}{2(\sqrt{3}Rs)^2}=\cos(\dfrac{2\pi}{3})\\ (x_a-x_{c1})^2+(y_a-y_{c1})^2=(\sqrt{3}Rs)^2\end{cases} \tag{6}$$

(4)在节点 A 的一跳邻节点中选取距离子战略点 c、d 最近的节点作为对应战略点的 ready_on_duty 节点 $C1$、$D1$，并将战略点 a 及对应战略点的地理信息告知 $C1$、$D1$；

(5)同理，结合 a、$b2$，求解 ready_on_duty 节点 $C2$、$D2$；

(6)结束。

4.3.2　节点扩散

节点扩散如图 14 所示。

1)ready_on_duty 节点 I 向 $N(I)$ 发出问询，在距其 thrd 范围内是否有骨干节点，若时长 τ 内没有收到骨干节点反馈消息，则标识自己为 on_duty 节点，并标记 $N(I)$；否则，标识自己为 ready_to_sleep 节点，跳转至 5)；

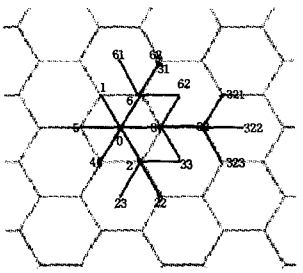

图 14　节点扩散

2)I 将结合战略点 i 及父战略点 $i.f$ 的坐标，按式(5)和式(6)求解出 3 个子战略点 $i.s1$、$i.s2$ 和 $i.s3$ 的坐标；

3)分别判断 3 个子战略点坐标是否满足式(7)，若满足则成为合理的战略点。若合理战略点个数为 0，转至 5)；

$$\begin{cases}x_{\min}-adb<x_{i.s}<x_{\max}+adb\\ y_{\min}-adb<y_{i.s}<y_{\max}+adb\end{cases} \tag{7}$$

4)在 $N(I)$ 中选择在父战略点一跳范围内离战略点最近的节点作为合理子战略点的 ready_on_duty 节点，I 将对应战略点 i、合理子战略点 $i.s$ 的位置信息告知，并通知其成为 ready_on_duty 节点，开始节点扩散算法；

5)结束。

4.4　优化初始覆盖集

1)on_duty 节点广播消息，标记感知范围 Rs 内的 undecided 节点为 ready_to_sleep；

2)剩下的 undecided 节点随机回退一段时间后醒来，查看自己的连通范围内是否存在非 undecided 节点，如果存在则选取“桥梁节点”(如果有 on_duty 节点，则选 on_duty 节点；如果没有，则选 ID 号最小的 ready_to_sleep 节点)。自己变为 on_duty 节点，并同时唤醒该桥梁节点为 on_duty 节点；

3)新的 on_duty 节点广播标记消息，标记感知范围内的节点为 ready_to_sleep 节点；

4)如果在回退过程中，undecided 节点收到了标记消息，则放弃成为 on_duty 节点。

4.5　填补空洞区域

节点在回退时间 T(依式(1))结束后，醒来判断自己是否冗余。

4.5.1　非边界节点的冗余判断

1)ready_to_sleep 节点 i 遍历 $N(i)$ 内的节点，假设此时选择的节点为 j；

2)节点 j 计算张角 $\angle_{j\to i}$；

3)节点 j 遍历 $S(j)$，计算每个节点对节点 j 的张角；

4)计算张角 $\angle_{j\to i}$ 是否被 $S(j)$ 中的节点覆盖，如果覆盖，则发送 wake0 消息给节点 i，如果没有被覆盖，则发送 wake1 消息给节点 i；

5)只要节点 i 收到一条 wake1，则醒来成为工作节点；若没有收到 wake1，则休眠。

4.5.2　边界节点的冗余判断

1)创建虚拟节点(依式(2))；

2)结合虚拟节点判断自己是否冗余，方法同边界节点；

3)如果冗余，继续判断其边界线中点是否被某一工作节点覆盖；如果不冗余，则醒来成为 on_duty 节点；

4)判断边界线中点节点坐标 (x_m,y_m)(依式(3))，如果 (x_m,y_m) 在某个工作节点的感知范围内，则近似为边界线被覆盖；

5)如果边界线中点被覆盖，则休眠成为 sleep 节点；如果没有被覆盖，则醒来成为 on_duty 节点。

4.6　节点调度

1)on_duty 节点进入 LEACH 协议的簇头建立阶段，簇形成后进入稳定工作阶段；本轮结束后跳转到 2)；

2)查看轮转次数是否达到冗余判断阶段，如果没有，跳 3)，如果达到，跳 4)；

3)查看是否有死亡节点，如果有，则标记为死亡节点，其余节点恢复为 normal 节点，跳 1)；

4)通知休眠节点判定自己是否为冗余节点，如果是冗余节点则继续为 sleep 状态，如果不是则醒来成为 on_duty 节点，跳 1)。

5　仿真结果及分析

为了评估本文提出的算法的性能，在 $Rc/Rs\geqslant\sqrt{3}$ 的条件

下对 RCSC 算法与其它算法进行了比较。为了能够计算仿真过程中的覆盖率，本文采用的方法是染色法。在节点分布图上，对感应范围内的工作节点的感知范围着色，其黑色区域为被覆盖区域，白色区域为未被覆盖区域。然后通过计算黑色区域的像素大小来确定感应区域的覆盖率大小。

首先，仿真过程体现了 RCSC 算法的运行过程（见图 15－图 18）。在网络目标区域中，网络中心的节点向外扩散形成初始节点覆盖集（见图 15），又由于区域中的某些节点没有存在于初始节点覆盖集的通信范围内，因此通过添加“桥梁节点”使整个网络节点达到“通信覆盖”（见图 16），最后某些处于感知空洞附近的节点醒来填补空洞，使整个网络达到“感知覆盖”（见图 18）。从图 18 可以看出，该算法解决了 Huang 算法无法对边界节点进行休眠判定的缺陷，大大降低了工作节点数量。并且，该算法在冗余判断之前已经选出了初始覆盖集节点，这样与 Huang 算法对比，更是降低了其通信复杂度。

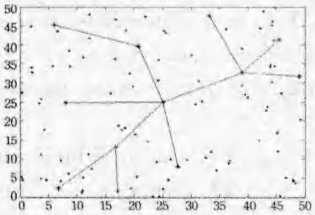

图 15　初始节点覆盖集

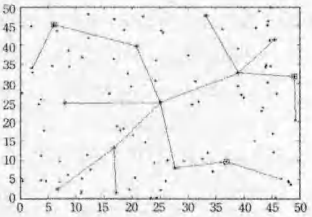

图 16　加入“桥梁节点”

图 17　达到“通信覆盖”

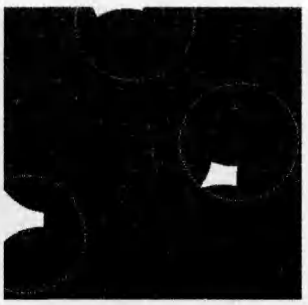

图 18　达到“感知覆盖”

5.1　覆盖率比较

该仿真比较了 RCSC 算法、CCP 算法和 ECBNSS 算法的覆盖率（见图 19）。从图 19 可以看出，它们都能够在给定条件下达到 99.9%以上的覆盖率。

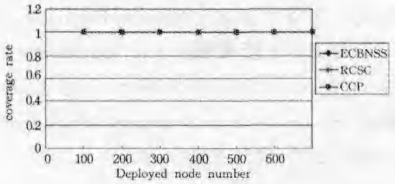

图 19　节点数 vs 覆盖率

5.2　工作节点数比较

图 20 表示了节点数目从 100 变化到 700，步长为 100 时，RCSC 算法、CCP 与 ECBNSS 算法所需要的工作节点数。仿真结果表明，当 $Rc/Rs \geqslant \sqrt{3}$ 时，该算法与 CCP 算法者最后选出的工作节点数相当，但该算法的计算复杂度优于 CCP。CCP 算法复杂度为 $O(\Delta^3)$，而本算法的复杂度为 $\max(O(\Delta\lg\Delta), O(\Delta), O(\Delta^2))$。并且，RCSC 的算法复杂度 $O(\Delta^2)$ 主要来自于空洞填补过程，而在这个过程期间 Δ 已经被控制得很小，所以 RCSC 的算法复杂度优于 CCP。而与 ECBNSS 相比，该算法的工作节点数减少了 10%左右。

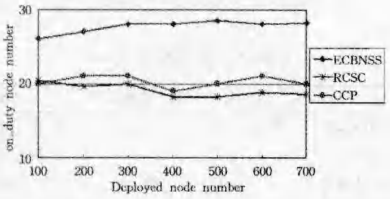

图 20　节点数 vs 工作节点数

而图 21 表示了当节点数目为 200、感知半径从 6 变化到 12、步长为 2 时，RCSC 算法与 ECBNSS 算法所需要的工作节点数。仿真结果表明，当 $Rc/Rs \geqslant 2$ 时，该算法与 ECBNSS 相比，工作节点数平均减少了 9%左右。

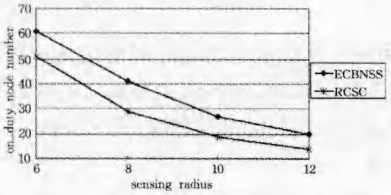

图 21　感知半径 vs 工作节点数

从图 21 仍可以看出，本算法的工作节点不随总节点密度增加而增加，工作节点基本保持稳定。因此，RCSC 算法可用于高密度的节点布置网络中，其可以很好地控制工作节点数，从而延长整个网络的寿命。

5.3　网络生存时间比较

在 LEACH 中，网络能耗主要包括簇建立阶段节点间的通信能耗 Ec 和稳定工作阶段的能耗 Eg。Ec 取决于网络节点数和预期簇首节点数，Eg 则包括节点采集数据、成员节点向簇首节点发送数据、簇首节点接收数据、簇首节点进行数据汇聚以及簇首节点向汇聚节点发送数据的能耗。在 RCSC 算法＋LEACH 协议中，除了这两部分能耗，还要考虑节点调度阶段节点间的相互通信。

本仿真选用第一顺序无线电能量模型（First order radio model）[13]，该模型有如下几个明显特征：

1）该模型网络中所有传感器节点完全相同。

2）汇聚节点位置固定并且离网络中的节点较远。

3）无线电信号在各个方向上消耗的能量相同。

在该模型中，当无线收发器发送 Kbit 的数据时，节点发送 Kbit 消息所消耗的能量为式（8），节点接收这个消息所消耗的能量为式（9）[13]。

$$E_{TX}(k,d)=\begin{cases}E_{elec}*K+\varepsilon_{fs}*K*d^2 & ,d<d_0\\ E_{elec}*K+\varepsilon_{amp}*K*d^4 & ,d\geqslant d_0\end{cases} \tag{8}$$

$$E_{RX}(k)=E_{elec}*K \tag{9}$$

式中，ε_{amp} 和 ε_{fs} 是信号放大器的放大新倍数，E_{elec} 为无线通信电路消耗的能量，d 为信号传输距离。

在计算方面，簇首节点主要考虑与一般工作节点和汇聚节点的通信能耗，还有对数据进行融合的能耗。而对一般工作节点，其主要能耗则是与簇头节点的通信能耗。本文暂不考虑 RCSC 中调度节点的节点耗能。节点无线通信半径等于该节点与相应簇首节点之间的距离。每轮的时间间隔 $T=10$s，节点每隔 1s 向簇首节点发送一个大小为 2000bit 的报告消息。簇首接到本簇内成员节点的消息后，以 20:1 的比例进行数据汇聚，再将处理后的数据发送给汇聚节点。仿真模型

具体参数如表 3 所列，E_d 为数据融合的能耗，p 为选择簇首的概率。

表 3　仿真参数表

参数	数值	参数	数值
区域范围 AreaR	50m * 50m	ε_{amp}	0.0013pJ/bit/m^4
节点数量 N	100	ε_{fs}	10pJ/bit/m^2
汇聚节点坐标	(0,0)	E_{elec}	50nJ/bit
节点初始能量 E0	2J	E_d	50nJ/bit/signal
数据包长度	2000bit	p	0.1
感知半径	10m	Ts	10s

由于 LEACH 协议运行的周期为 10s，而频繁地进行冗余节点判定反而会加重网络负担，造成能量浪费，因此在仿真中每隔 20 轮才进行一次冗余节点判定，即休眠机制的周期为 200s。

图 22 为系统总能量消耗随时间变化的曲线图。从图中可以看出 RCSC+LEACH 协议比原始 LEACH 协议能量消耗要慢得多，因此，相同的系统能量下，RCSC+LEACH 能维持更长的时间。

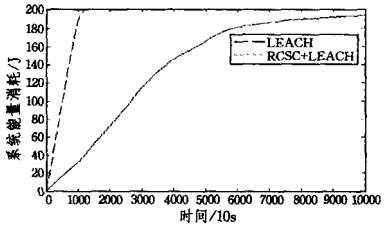

图 22　系统能耗随时间变化图

图 23 为网络剩余有效节点随时间变化的曲线图。从图中可看出 RCSC+LEACH 协议比原始 LEACH 协议的网络剩余有效节点数的衰减速度要慢得多。这是由于在 LEACH 中，所有具有能量的节点都一直处于工作状态。RCSC+LEACH 由于之前具有一个调度阶段，因此每一轮只有一部分的节点在工作，而其它节点处于非常节能的休眠状态。当某些工作节点失效后，休眠节点才会醒来成为工作节点，开始为整个网络感知数据和通信数据。

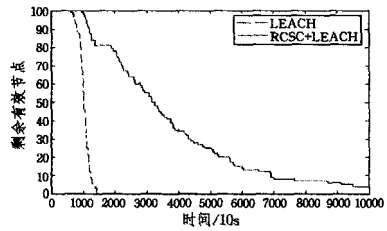

图 23　剩余有效节点随时间变化

图 24 为网络覆盖率随时间变化的曲线图(其中 RCSC+LEACH 曲线为零散时间点计算结果连成的折线)。从图中可看出，相比原始 LEACH 协议，RCSC+LEACH 协议的网络覆盖率的衰减速度要慢得多。

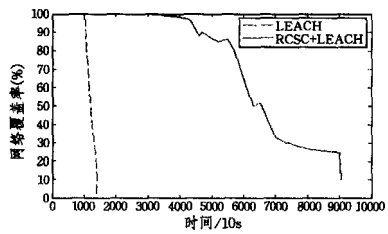

图 24　网络覆盖率随时间变化图

结束语　本文在保证网络区域“感知覆盖”和“通信覆盖”的基础上，提出了一种基于蜂窝模型的分布式节点调度算法(RCSC)。RCSC 通过类似 DBC 算法中对蜂窝模型的构造方式，得到了初始节点覆盖集。然后通过添加“桥梁节点”使得整个网络达到“通信覆盖”。最后对 Huang 算法进行改进，提出一种更优的空洞填补机制，使得整个网络达到“感知覆盖”。仿真实验结果表明，由 RCSC 算法所选取的工作节点数量少并且稳定，受网络节点密度的影响小。在 RCSC 算法基础上建立路由(如 LEACH 协议)，能够减少由于冗余数据通信导致的额外能量消耗，并延长网络生存时间。

参 考 文 献

[1] Bai Xiao-le. Deploying Wireless Sensors to Achieve Both Coverage and Connectivity[C]// Proceeding of the 7th ACM International Symposium on Mobile Ad hoc Network and Computing. 2006:131-142

[2] Bai Xiao-le, Zhang Chuan-lin, Xuan Dong. Optimal Deployment Patterns for Full Coverage and k-Connectivity (k<=6) Wireless Sensor Networks[J]. IEEE/ACM Transaction on Networking, 2010, 18(3)

[3] Durresi A, Paruchuri V K, Iyengar S S. Optimized Broadcast Protocol for Sensor Networks[J]. IEEE Transaction on Computers, 2005, 54(8)

[4] Paruchuri V, Durresi A. Adaptive Coordination Protocol for Heterogeneous Wireless Networks [C] // Communications, 2007. ICC'07. IEEE Intenational Conference. 2007:4805-4810

[5] Xiang Yu, Liu Xiao-juan. A Distributed Algorithm for Virtual Backbone Construction with Cellular Structure in WSNs[J]. International Journal of Distributed Sensor Networks 2012, 2012

[6] Tian D, Georganas N D. Connectivity Maintenance and Coverage Preservation in Wireless Sensor Networks[C]// Proceedings of CCECE 2004. IEEE Canada, Sheraton Fallsview, Canada, 2004

[7] Tian D, Georganss N D. A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks[C]// Proceedings of the lst ACM Workshop on Wireless Sensor Networks and Applications (WSNA'02, in Conjunction with ACM MobiCom'02). Atlanta, Georgia, USA, 2002

[8] Tian D, Georganas N D. Connectivity Maintenance and Coverage Preservation in Wireless Sensor Networks[C]// Proceedings of CCECE 2004. IEEE Canada, Sheraton Fallsview, Canada, 2004

[9] Huang C F, Tseng Y C. A Survey of Solutions to the Coverage Problems in Wireless Sensor Networks[J]. Journal of Intemet Technology, Special Issue on Wireless Ad Hoe and Sensor Networks, 2004, 12(3):2356-2359

[10] Huang C F, Tseng Y C. The coverage problem in a wireless sensor network[C]// Sivalingam KM, Raghavendra CS, eds. Proc. of the ACM Intl Workshop on Wireless Sensor Networks and Applications (WSNA). New York: ACM Press, 2003:155-121

[11] Kershner R. The Number of Circles Covering a Set[J]. Am. J. Math., 1939, 61:665-671

[12] Heinzelman W, Chandrakasan A, Balakrishnam H. An application specific protocol architecture for wireless microsensor networks[J]. IEEE Transactions on Wireless Communications, 2002, 1(4):660-670

[13] 王雪. 无线传感器网络测量系统[M]. 北京：机械工业出版社，2007