

一种连通性覆盖的无线传感器网络节点调度算法

朱孟贝, 张曦煌

(江南大学 物联网工程学院, 江苏 无锡 214122)

摘要: 研究传感器节点随机部署于监测区域内, 无节点地理位置信息情况下, 如何能量有效地保证网络的通信连通与感知覆盖; 节点采用基于概率的联合感知模型。提出 CDS-based SSCA 算法, 其为一种基于连通支配集构造树的节点调度机制, 每个节点根据剩余能量和与父节点的距离来设置等待时间及成为候选节点优先级。模拟实验结果显示, 本算法能够能量有效地满足感知覆盖和连通覆盖要求; 与 ASW 算法相比较, 工作节点个数较少, 网络生命周期明显延长, 降低了网络整体耗能。

关键词: 无线传感器网络; 基于概率的联合感知模型; 连通性覆盖; 节点调度算法

中图分类号: TP393 **文献标志码:** A **文章编号:** 1001-3695(2012)08-3091-05

doi: 10.3969/j.issn.1001-3695.2012.08.076

Sensor scheduling algorithm for connected coverage in wireless sensor networks

ZHU Meng-bei, ZHANG Xi-huang

(School of IOT Engineering, Jiangnan University, Wuxi Jiangsu 214122, China)

Abstract: This paper researched on how to energy-efficiently schedule sensor nodes and met both constraints of connectivity and coverage, in condition of random deployment policy without accurate location information. In addition, it employed sensors in probabilistic cooperative sensing model. This paper proposed a sensor scheduling algorithm called CDS-based SSCA based on connected dominating set. Sensor node had its own scheduling priority and waiting time on the basis of its residue energy and the distance between them and those parent node. Simulation results show that this algorithm can effectively satisfy connectivity and coverage of network. Compared with ASW (adaptive sensor scheduling) algorithm, CDS-based SSCA has less activity sensor nodes, thus, it can effectively prolong the network's lifetime and save the network's energy.

Key words: wireless sensor networks; probabilistic cooperative sensing model; connected coverage; sensor scheduling algorithm

0 引言

能量有效及连通性覆盖问题是衡量无线传感器网络中质量的重要标准。无线传感器网络部署之后, 传感器节点采集感知范围内的原始数据, 通过多跳方式将该数据传递给称为基站的特殊节点, 基站具有较强的处理能力和较大的初始能量, 它对所采集的原始数据进行一定处理后再发送给控制中心。连通性覆盖是无线传感器网络中的一个重要问题, 其所解决的是如何保证监测区域中所有节点形成的监测范围可以满足应用需求; 同时, 任何一个节点都可以将其感知的数据转发到基站而不产生网络分隔。如果无法保证一定的通信覆盖性和监控覆盖性, 那么就无法达到监测的目的。目前, 无线传感器网络连通性覆盖研究的工作大多是在节点地理位置信息上进行研究的。文献[1]通过实验验证表明, 在传感器节点中嵌入 GPS 或有向天线等定位装置, 不仅提高了网络成本, 而且需要消耗大量能量。因此, 解决在没有节点位置信息的情况下, 达到连通性覆盖要求可有效实现能量有效问题。此外, 目前大多无地理位置信息的连通性覆盖节点调度控制算法是基于布尔感知

模型的, 即在节点感知范围内, 感知覆盖概率为 1, 否则为 0。这种模型有失真性, 且目标点是否被感知仅由一个节点决定。而现实应用中, 目标点的监测是由周围多个工作节点进行感知, 并对数据进行融合处理得到监测数据。因此, 基于概率的联合感知模型^[2]能更真实、准确地描述网络感知覆盖性能。

1 相关工作

Tarik 等人^[3]提出了针对无线传感网部分覆盖所需要的分布式节点睡眠调度算法。该算法无须知道节点的地理位置信息, 节点通过计算与 sink 节点间的跳数及反馈给 sink 节点自身的剩余能量来进行节点调度, 但其要求时钟同步, 势必会影响整个网络的能量有效性。Tian 等人^[4]提出了基于免职合格规则的分布式覆盖调度算法, 根据节点的地理位置信息或接收信号到达角度来计算节点间的感知覆盖面积关系。基于探测的密度控制算法 PEAS^[5]无须知道节点的地理位置信息, 每个休眠状态的节点要定期地在其覆盖范围内监测其邻居节点的运行状态, 若在其覆盖范围内没有节点处于工作状态, 则其立刻进入工作状态; 否则, 仍处于休眠状态。上面的调度控制方法均是针对

收稿日期: 2011-11-18; **修回日期:** 2011-12-20 **基金项目:** 轻工过程先进控制教育部重点实验室开放课题 (APCLH1004); 高等学校学科创新引智计划资助项目 (B12018)

作者简介: 朱孟贝 (1987-), 女, 江苏徐州人, 硕士研究生, 主要研究方向为嵌入式系统、计算机网络 (zmb098@163.com); 张曦煌 (1962-), 男, 教授, 博士, 主要研究方向为嵌入式系统、计算机网络。

网络覆盖问题,没有考虑网络连通性问题。

近年来,伴随无线传感器网络快速发展,其连通性覆盖问题讨论也日益热烈。针对连通性与覆盖性关系,Zhang 等人^[6]证明了当通信半径是感知半径的两倍以上时,监测区域 k 覆盖也势必 k 连通,并以此结论为依据,提出了优化的地理密度控制算法(optimal geographical density control, OGDC)。蒋杰等人^[7]针对连通性覆盖集构造问题,设计了一种基于目标区域 Voronoi 划分的集中式近似算法(centralized Voronoi tessellation, CVT),针对通信半径与感知半径未达到两倍关系情况提出了一种基于最小生成树的连通算法来保证网络的连通性。文献[8]中通过选择节点连通路经来获得最大网络感知覆盖率,当基站向网络发送一个监测区域查询消息时,分别提出了集中式与分布式两种贪婪算法,选择最小的连通工作节点集并充分感知覆盖监测区域。文献[9,10]通过在节点集中选取骨干节点构造连通支配集来保证网络连通性实现数据顺利传达到基站,但均未充分考虑网络的感知覆盖要求。文献[11]基于连通支配集,提出自适应能量有效节点调度算法(adaptive sensor scheduling, ASW),该算法通过调度部分节点处于工作状态,在保证网络连通覆盖和感知覆盖前提下,与文献[12]算法进行仿真实验比较,有效地延长了网络生命周期。本文通过仿真实验,对 CDS-based SSCA 和 ASW 两节点调度算法进行比较。

2 预备知识及问题描述

2.1 感知模型及相关定义

不同于布尔感知模型,基于概率的感知模型考虑节点感知过程中的不确定性。假设探测概率是基于距离连续衰减的,当监测区域内事件与传感器节点距离大于布尔感知模型中最大感知距离时,该模型就可以更真实地描述事件的感知情况。定义 $\lambda_i(j)$ 为监测区域内 j 点被节点 i 感知概率,计算公式为

$$\lambda_i(j) = \begin{cases} 1 & d_{ij} \leq r_i \\ e^{-k(d_{ij}-r_i)^m} & r_i < d_{ij} \leq r_s \\ 0 & d_{ij} > r_s \end{cases} \quad (1)$$

其中: d_{ij} 为节点 i 与监测区域内 j 点的欧式距离; r_i 相对于布尔感知模型中的感知半径, r_s 为节点的最大感知半径,当 d_{ij} 在 $(r_i, r_s]$ 区间时,感知概率成指数级衰减; k 和 m 是衰减系数, k, m, r_i, r_s 与节点感知设备参数和环境因素相关。

j 点的感知概率^[13]表示为

$$\lambda(j) = 1 - \prod_{i \in S} (1 - \lambda_i(j)) \quad (2)$$

其中: S 为当前所有工作节点的集合。如果感知概率超过阈值 ε ,笔者认为目标点 j 被感知覆盖;反之,不被感知覆盖。

定义 1 点被覆盖。IsCov(j) 表示目标点 j 被感知覆盖情况,表示为

$$\text{IsCov}(j) = \begin{cases} 1 & \lambda(j) \geq \varepsilon \\ 0 & \lambda(j) < \varepsilon \end{cases} \quad (3)$$

定义 2 点覆盖区域面积。节点 i 感知覆盖范围是以节点位置为中心,以 r_{is} 为半径的圆,节点 i 覆盖区域面积计算为

$$A_i = \pi \cdot r_{is}^2 \quad (4)$$

其中: r_{is} 是节点 i 感知范围内目标点被感知覆盖的最大感知半径。

定义 3 网络感知覆盖质量。监测区域 A 中被感知覆盖的面积与整个监测区域面积的比例,称为网络感知覆盖质量,即

$$\varphi = \|\text{IsCov}(j) d_A / \|A\| \quad (5)$$

其中: $\|A\|$ 为检测区域总面积;检测区域分成多个小闭区域 d_A , j 为区域 d_A 内任意一点,IsCov(j) 表示 j 点被感知覆盖情况并由式(3)计算;分子部分的二重积分即代表区域 A 内被感知覆盖的总面积。网络感知覆盖质量是衡量网络质量的一个重要标准。

2.2 网络模型

所有传感器节点采用随机部署策略部署在边长为 l 的正方形监测区域 A 内,并且传感器节点在区域 A 内密度足够大,即如果所有节点均处于工作状态,可以感知覆盖整个区域 A 。现假设该无线传感器网络具有如下性质:

- 相对于节点的感知覆盖范围,监测区域面积足够大,可以忽略边界效应;
- 所有传感器节点部署后为静止状态;
- 节点采用基于概率的联合感知模型;
- 所有节点同构,节点的最大通信半径(r_c)和最大感知半径(r_s)无特定关系,感知半径固定;
- 除 sink 节点外所有节点具有相似的处理能力和初始能量,且地位相等;
- 网络不要求时钟同步;
- 节点不知道其邻居节点的位置及角度信息,但节点可根据从邻居节点接收信号强度的大小大致确定两者之间的距离。

2.3 CDS

无线传感器网络可以抽象为一个无向图 $G = (V, E)$ 。其中: V 是传感器节点集合,也就是图中的顶点; E 是节点间通信线路的集合,也就是图中的无向边。

定义 4 支配集(dominating set, DS)。对于无向图 $G = (V, E)$,若顶点集 $C \subseteq V$,并且对任意顶点 $u \in V - D$,存在顶点 $v \in D$,使得 $(u, v) \in E$,则称顶点集 D 是图 G 的支配集。若 D 的任何两个顶点之间是相连通的,则称 D 是连通支配集(connected dominating set, CDS)。连通支配集可用于构造无线网络的虚拟骨干网来进行数据通信。如图 1 所示,图中三个黑色实心圆点组成集合即为该图的连通支配集。

2.4 问题描述

在监测区域 A 中随机均匀地部署大量传感器节点(包含少量 sink 节点),节点采用上述基于概率的联合感知模型,并给定节点感知模块相关设备参数,目标点的感知强度阈值 ε ,感知半径 r_i ,最大感知半径 r_s 。采用 2.2 节所述网络模型,传感器节点个数为 N 。通过基于 CDS 的节点调度控制算法,获得工作节点集合,同时达到能量有效及连通覆盖和感知覆盖要求。

3 基于 CDS 的节点调度控制算法

基于 CDS 的节点调度控制算法(CDS-based sensor scheduling control algorithm, CDS-based SSCA)不要求时钟同步,这样可减少大量通信产生的能量消耗,有效实现网络的能量有效性。同时本算法为分布式,避免了集中式算法无法针对网络改

变及时调整调度的弊端。CDS-based SSCA 运行过程包含四个阶段,具体描述如下:

1) 邻居节点发现阶段

节点部署好后,通过预先选定的节点(通常是 sink 节点)开始创建 CDS 树。Sink 节点的初始状态是工作状态(active state),sink 节点首先广播一个 hello 消息。如果一个节点没有接收到其他 hello 消息,它就把发送消息节点认做父节点(parent node),把自己认做子节点(child node),并反馈一个父节点确认消息(parent recognition message, PRM)。如果一个节点已经收到 hello 消息,它便忽略这个消息。父节点确认消息包括该子节点的剩余能量及该子节点接收此 hello 消息时的信号强度,利用式(6)权衡计算其值 W_{ij} ,并存储在表 list 中。如果一个父节点没有从它的邻居节点获得父节点确认消息,这说明它在网络中没有与其他节点通信连通,这时设置定时时间 $timeout_1$ 。如某子节点被父节点感知覆盖,则向父节点发送感知覆盖消息(sensing covered messages, SCM)。在 $timeout$ 时间内,父节点的所有邻居节点仍然都被各自的父节点感知覆盖,而没有给它发送感知覆盖消息,那么它将超时并进入休眠状态(sleeping state)。

$$W_{ij} = \alpha \times \frac{RE_i}{IE_{\max}} + \beta \times \frac{SS_j}{SS_{\min}} \quad (6)$$

W_{ij} 值在(0,1)区间,其值越大优先级则越高。其中: i 是候选节点, j 是它的父节点; α 是节点剩余能量的权重; RE_i 是节点 i 的剩余能量; $IE_{\max}$ 是最大初始能量; β 是与父节点距离的权重; SS_j 是从父节点 j 处接收到 hello 消息的信号强度; $SS_{\min}$ 是保证连通的最小信号强度。

2) 子节点选择阶段

父节点设置定时 $timeout_1$ 来等待子节点反馈消息,一旦接收到父节点确认消息,便存储子节点于候选列表(candidate list)中,并进入工作状态。定时时间结束,父节点对候选列表进行降序排列,并向子节点发送子节点确认消息(children recognition message, CRM),该消息包括该子节点位于候选列表中的位置和它的最大感知半径(r_s)。通过子节点确认消息,子节点可以确定是否被父节点感知覆盖,如没有被父节点感知覆盖,则需多等待一段时间 $timeout_1/5$ 。当某一子节点接收到父节点发送来的子节点确认消息时,它根据其位于候选列表中 W_{ij} 大小和其感知覆盖面积设置一个定时时间 $timeout_i$,该时间长度计算如式(7),并设置自身状态为候选节点(candidate node),如果在该定时时间内,它从兄弟节点处获得兄弟阻塞消息(brother blockage message, BBM),它便关闭一定时间 $timeout_i$ 并进入休眠候选状态(sleeping candidate),也就是说它的兄弟节点比它更适合作树中节点;反之,定时结束时仍然没有收到兄弟阻塞消息的节点进入活动候选状态(active candidate),并开始发现其未覆盖的节点。位于 list 中的最优节点首先发送兄弟阻塞消息。

$$timeout_i = \left(1 - \gamma \times W_{ij} - \theta \times \frac{A_i}{A}\right) \times T_{\max} \quad (7)$$

其中: $timeout_i$ 为传感节点 i 需设置的等待时间; γ 和 θ 分别是节点能量权重和感知覆盖区域权重; RE_i 为传感节点 i 的剩余能量; A_i 是传感节点 i 感知覆盖区域的面积; A 是整个监测区域面积; $T_{\max}$ 是设置的最大等待时间。

3) 机会阶段

为了避免一些节点进入休眠状态导致部分网络阻塞而影响网络的连通性能,当一个节点接收一个兄弟阻塞消息时开始设置定时,定时到期便调整到活动候选状态并发送一个 hello 消息来发现自己的邻居节点。这样可以保证网络中节点的连通性。

4) 感知选择阶段

如上所述,一些节点在第一阶段没有被选进 CDS 树,并且没有通过接收子节点确认消息被父节点感知覆盖,它们需要再待命一段时间 $timeout_2$ 。如果一个没被选入初始树的节点在超时前收到感知覆盖消息,也就是说两者之间距离足够小,那么它被感知覆盖并立即进入休眠状态。如果两者之间距离大于最大感知距离,这说明它虽然被通信覆盖但是没有被感知覆盖,所以它仍要被选择进 CDS 树的覆盖活动节点(active node for coverage)而进入工作状态,这样可以有效地扩大整个网络的感知覆盖面积。同时该节点发送一个感知覆盖消息和一个兄弟阻塞消息给它的邻居节点,以便减少工作邻居节点个数,实现网络能量有效性。

最后一个节点完成自己的创建进程后,整个过程结束。每个节点只需知道自己的进程,并不需要了解整个进程。实际上,一旦一个节点被选为 CDS 树的一部分,它便可开始其应用工作,一旦算法开始运行,每个节点不是工作状态就是休眠状态。

本算法的计算复杂度为 $O(n \log n)$ 。

CDS based-SSCA 的伪代码描述如下:

```

if (node is sink)
    node.setMode( active)
else   node.setMode( sleep)
end if
parent.add( sink)
sink.send( hello message)
if (sink.receive( PRM)) {
    sortList(  $W_{ij}$ )
    send( CRM, list)
}
else {
    while(  $timeout_1 > 0$ ) {
        if (listenall( SCM))
            sink.setMode( close)
        else   sink.setMode( sleep)
    }
}
end if (sink.receive( PRM))

if (node.receive( hello) && isFirst) {
    child.add( node)
    node.send( PRM,  $W_{ij}$ )
}
end if
if (node.receive( CRM)) {
    if (node.SensingCoved( parent) == false)
        node.waitTime.add(  $timeout_1/5$ )
    end if
    Candidate.add( node)
}
end if
if (node.inList == 1)
    node.send( BBM)
else {

```

```

while( timeout1 > 0 ) {
    if( node.receive( BBM ) ) {
        close.add( node )
        SleepingCandidate.add( node )
    }
}
end if
{
    ActiveCandidate.add( node )
}
end if
if( child( node ) == false && parent( node ) == false )
    node.waitTime( timeout2 )
end if
while( timeout2 > 0 ) {
    if( node.receive( SCM ) ) {
        if( distance( node, originofSCM ) < Rs )
            node.setMode( sleep )
        else node.setMode( active )
        end if
    }
}
end if
if( node.mode == active )
    node.send( SCM and BBM )
end if
}

```

4 性能评价

4.1 模拟实验环境及参数设置

本文使用 OPNET 仿真平台对本算法进行实验和分析,并与基于 CDS 的自适应节点调度算法 ASW 进行比较。

ASW 算法通过选择部分节点处于工作状态达到延长网络生命周期的目的,同时又保证网络的覆盖及连通要求。ASW 算法主要由 CDS 骨干节点选择和面向服务请求的 CDS 更新机制两部分组成。每个节点 i 关联一系列参数域:时间域 $t(i)$ 、优先级域 $p(i)$ 、状态域 $status(i)$ 、CDS 域 $CDS(i)$ 、角色标签域 $r(i)$ 。其中:时间域是指节点 i 保持工作状态的时间,该值基于节点 i 是否为 CDS 组成节点和其当前是否要求服务决定;优先级域由节点 i 剩余能量和节点 i 标志符决定,剩余能量越多优先级越大,如果剩余能量相同则根据节点标志符决定优先级大小,CDS 每时间 T 更新一次,CDS 构造前,节点更新其优先级,并在下个时间 T 内 $p(i)$ 保持不变;状态域为工作或休眠两种,由该节点的工作状态所决定;CDS 域为 true 或 false,由该节点当前是否为 CDS 骨干网节点所决定;如某节点当前位于 CDS 内并且它所服务的区域为目标区域时,角色域记录该节点当前角色。

本实验与文献[14]采用同样的能量模型和参数,其中发送数据和接收数据的无线通信模型计算公式分别如式(8)和(9)所示。

$$E_{\text{send}}(\text{len}, d) = \begin{cases} \text{len} \times E_{\text{elec}} + \text{len} \times \xi_{\text{fs}} \times d^2 & d < D \\ \text{len} \times E_{\text{elec}} + \text{len} \times \xi_{\text{mp}} \times d^4 & d \geq D \end{cases} \quad (8)$$

$$E_{\text{receive}}(\text{len}) = \text{len} \times E_{\text{elec}} \quad (9)$$

其中: len 表示数据包的长度,即比特位数; E_{elec} 表示无线收发电路所消耗的能量; ξ_{fs} 表示自由空间模型放大器所消耗的能量; ξ_{mp} 表示多路衰减模型放大器所消耗的能量; d 表示发射距离; D 为距离阈值,另外还有上述其他实验参数,列表如表 1 所示。

表 1 模拟实验参数设置列表

parameter	value	parameter	value
A/m^2	100×100	ε	0.6
N	400 ~ 800	$\text{timeout}_1/\text{s}$	30
proportion of sink/%	2	$\text{timeout}_2/\text{s}$	60
r_t/m	3	T_{max}	15
r_s/m	10	$T(\text{ASW update time, s})$	1800
r_c/m	18	$E_{\text{elec}}/\text{nJ/bit}$	50
D/m	10	$\xi_{\text{fs}}/\text{pJ/bit/m}^2$	10
k	0.3	$\xi_{\text{mp}}/\text{pJ/bit/m}^4$	0.0013
m	1.5	data packet/bit	200
α	0.6	broadcast packet/bit	20
β	0.4	packet header	5
γ	0.65	$\text{IE}_{\text{max}}/\text{J/battery}$	2.5
θ	0.35	$\text{SS}_{\text{min}}/\text{Hz}$	20

4.2 实验结果

图 2 为初始节点总数为 400 时,要达到对应感知覆盖率要求($\text{QoC} = 80\%、85\%、90\%、95\%、99.5\%$),两调度算法所对应的工作节点个数。从图中可看出,在相同的网络覆盖率要求下,CDS-based SSCA 比 ASW 算法要求的工作节点个数略少。

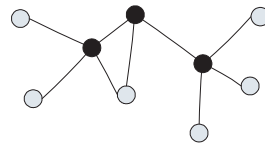


图1 连通支配集示例

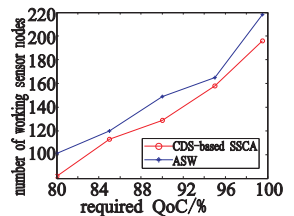


图2 不同覆盖率要求对应工作节点个数($N=400$)

在初始节点个数 400 ~ 800 间,本文分别计算了两调度算法达到感知覆盖率要求分别为 80%、90%、99.5% 时所要求的工作节点个数,如图 3 所示。

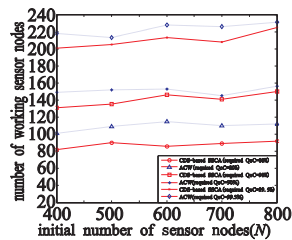


图3 不同节点个数及覆盖率要求下,两算法要求工作节点个数

从图 2、3 中可看出,在相同的网络覆盖率要求下,CDS-based SSCA 比 ASW 算法要求的工作节点个数略少。工作节点个数直接影响整个网络的生命周期,工作节点越少,网络的生命周期便越长。

另外,由于 ASW 算法中节点优先级只考虑了节点剩余能量,并在周期时间 T 内恒定不变;其次,ASW 算法每时间 T 重新构造 CDS 节点集合,该方法未能及时跟进网络拓扑改变,导致部分调度不够优化,并在一定程度上加速了网络能量的消耗,即缩短了网络生命周期。图 4 是初始部署节点总数为 800 时,三种节点调度策略所获得的感知覆盖质量与时间的关系。

从图 4 可以很直观地看出,相对于无节点调度机制的简单传感器网络,采用节点调度算法可有效延长网络生命。同时,与 ACW 算法相比较,将本文提出的节点调度算法应用到网络中可更有效实现网络负载均衡,延长网络生命周期。

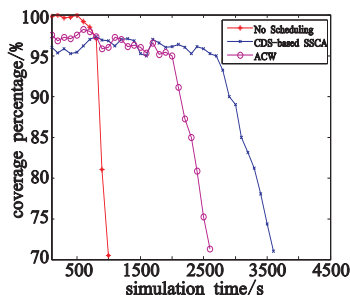


图4 覆盖率为95%下对应时间关系

5 结束语

随着无线传感器网络的广泛应用,能量有效地保证网络连通性覆盖问题得到关注。本文在节点随机部署机制下,节点采用基于概率联合感知模型,该感知模型相对于布尔感知模型可以更精确地描述网络覆盖能力;并在此基础上设计了一种基于连通支配集的节点调度控制算法(CDS-based SSCA)。该算法综合考虑节点剩余能量及距离因素等,在达到网络感知覆盖率及网络连通要求下,启动尽量少的节点处于工作状态。仿真实验将CDS-based SSCA与ASW算法进行对比,虽然两节点调度算法均通过选择部分节点处于工作状态,有效地实现了无线传感器网络连通覆盖和感知覆盖要求,但CDS-based SSCA更有效地延长了网络的生命周期,较好地达到了网络负载均衡。

参考文献:

- [1] STOJMENOVIC I. Position based routing in Ad hoc networks[J]. *IEEE Communications Magazine*, 2002, 40(7): 128-134.
- [2] ZOU Y, CHAKRABARTY K. Sensor deployment and target localization in distributed sensor networks[J]. *ACM Trans on Embedded Computing Systems*, 2004, 3(1): 61-91.
- [3] YARDIBI T, KARASAN E. A distributed activity scheduling algorithm for wireless sensor networks with partial coverage[J]. *Wireless Networks*, 2010, 16(1): 213-225.
- [4] TIAN D, GEORGANAS N D. A node scheduling scheme for energy conservation in large wireless sensor networks[J]. *Wireless Communications and Mobile Computing*, 2003, 3(2): 271-290.
- [5] YE Fan, ZHONG G, CHENG J, *et al.* PEAS: a robust energy conserving protocol for long-lived sensor networks[C]//Proc of the 23rd International Conference on Distributed Computing Systems. Washington DC: IEEE Computer Society, 2003: 28-37.
- [6] ZHANG H, HOU J C. Maintaining sensing coverage and connectivity in large sensor networks[J]. *Wireless Ad hoc and Sensor Networks*, 2005, 1(1-2): 89-123.
- [7] 蒋杰,方力,张鹤颖,等. 无线传感器网络最小连通覆盖集问题求解算法[J]. *软件学报*, 2006, 17(2): 175-184.
- [8] GUPTA H, DAS S R, GU Q. Connected sensor cover: self-organization of sensor networks for efficient query execution[C]//Proc of ACM International Symposium on Mobile Ad hoc Networking and Computing. New York: ACM Press, 2003: 189-200.
- [9] CHEN Y, LIESTMAN A. Approximating minimum size weakly-connected dominating sets for clustering mobile Ad hoc networks[C]//Proc of ACM International Symposium on Mobile Ad hoc Networking and Computing. New York: ACM Press, 2002: 165-172.
- [10] WAN P J, ALZOUBI K M, FRIEDER O. Distributed construction of connected dominating set in wireless Ad hoc networks[C]//Proc of IEEE INFOCOM. [S. l.]: IEEE Press, 2002: 1597-1604.
- [11] YANG Yin-ying, CARDEI M. Adaptive energy efficient sensor scheduling for wireless sensor networks[J]. *Optimization Letters*, 2010, 4(3): 359-369.
- [12] INTANAGONEWIWAT C, GOVINDAN R, ESTRIN D. Directed diffusion: a scalable and robust communication paradigm for sensor networks[C]//Proc of the 6th Annual International Conference on Mobile Computing and Networking. New York: ACM Press, 2000: 56-67.
- [13] ALTINEL I, ARAS N, GNEY E, *et al.* Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks[J]. *Computer Network*, 2008, 52(12): 2419-2431.
- [14] HEINAELEMAN W, CHANDRAKASAN A P, BALAKRISHNAN H. An application-specific protocol architecture for wireless microsensor networks[J]. *IEEE Trans on Wireless Communications*, 2002, 1(4): 660-670.