

# 一种面向监测区域的链路质量和覆盖保证的节点调度算法

杨 浏

(国家无线电监测中心成都监测站 监测业务一室, 成都 610000)

**摘要:** 为了降低监测区域能耗总开销和减少网络传输时延, 保证监测区域网络链路质量、实现网络的全面覆盖和延长网络生命周期, 设计了一种基于扫描线和节点自适应调整苏醒时隙的节点调度方案; 首先, 定义了系统模型即网络假设和调度目标; 然后判断网络是否实现当完全覆盖, 当不能全面覆盖时, 通过调整部分节点的感知半径来实现网络的全面覆盖; 当链路质量过差导致传输延迟过大时, 通过设计从源节点到目标节点的增加节点苏醒时隙, 并根据节点的剩余能量和传输延迟阈值来减少数据传输次数以降低传输延迟; 在 NS2 环境下进行实验, 结果表明: 文中方法能有效地实现传感器网络监测区域的全面覆盖, 降低网络的传输延迟和提高网络的生命周期, 与其他节点调度相比, 具有很强的优越性和实用性。

**关键词:** 覆盖; 节点调度; 无线传感器网络; 路由

## Sensor Node Scheduling Algorithm for Monitoring Area Based on Coverage and Link Quality Guarantee

Yang Liu

(Monitoring Business Section I, Chengdu Monitoring Station of SRMC, Chengdu 610000, China)

**Abstract:** In order to reduce network energy consumption and decrease the network time delay consumption, guaranteeing the monitoring area network link quality and prolong the network life cycle, a node scheduling method based on scanning and node adaptive adjust the reviving time slot is proposed. Firstly, the network suppose and scheduling goal are defined, then the network is justified whether can realize the network converge, when the network is not coverage comprehensively, the radius of partial nodes can be adjusted to realize the network coverage. When the transporting time delay is too big, the node reviving time slot is added, and the transporting time delay can be reduced according to the rest energy of sensor nodes and transporting delay threshold. The experiment is operated in NS2 environment, the result shows: the method in this paper can realize the node scheduling in sensor monitoring area, network energy consumption and transporting delay are reduced and the network lifecycle is prolonged. Therefore, it has stronger application foresight and priority over the other node scheduling methods.

**Keywords:** coverage; node scheduling; wireless sensor network; route

## 0 引言

在无线传感器网络 (wireless sensor network, WSN) 中<sup>[1-3]</sup>, 传感器节点往往分布在野外环境恶劣或无人能及的区域<sup>[2-5]</sup>, 为了保证网络的高覆盖率, 通常为每立方米部署 20 个节点, 这造成大量冗余数据的存储和传输, 浪费了大量能源<sup>[6]</sup>。由于节点的部署缺乏先验知识, 因此, 仍然可能造成覆盖盲区<sup>[7]</sup>。

为了实现覆盖区域的最大程度地覆盖, 文献 [8] 以最大化分组数为目标, 通过对节点的冗余性进行判断和与邻居节点的信息交互, 实现工作节点的分组调度。文献 [9] 通过相对剩余能量水平优化覆盖冗余判断, 并通过增加两种状态来改进节点状态分布。文献 [10] 采用桥梁节点和填补空洞优化工作节点集, 达到网络的通信覆盖和全面感知。文献 [11] 设计了一种交错的周期倍增的同步休眠调度方案, 将不同层节点运行

不同频率的激活和休眠周期, 实现降低时延和节能。

上述文献都研究了 WSN 的传感器节点调度问题, 但仍然不能完全避免覆盖盲区问题, 同时数据延迟较大且没有考虑链路的可靠性, 因此, 文中提出了一种基于节能和链路质量保证的节点调度方案。

## 1 系统模型

### 1.1 网络假设

假设  $n$  个传感器节点构成的节点集合  $S$  组成 WSN, 传感器节点在初始时刻随机地部署在二维的监测区域中, 并假设下列条件成立:

1) 所有传感器同构, 具有相同的处理能力、处理能力和通信能力, 具有相同的初始能量。其监测范围和区域满足单位同圆性质;

2) 所有传感器节点的位置已知 (通过 GPS 定位装置或定位算法), 任意两个节点的相互距离可以精确进行计算;

3) 节点采用布尔感知模型, 节点感知半径  $R_s$  可以调整, 其半径值满足  $R_{smi} \leq R_s \leq R_{sma}$ , 感知半径  $R_s$  按单位长度进行增加或减少;

收稿日期: 2014-09-16; 修回日期: 2014-10-17。

作者简介: 杨 浏 (1982-), 男, 硕士研究生, 工程师, 主要从事通信与信息系统、计算机科学与技术方向的研究。

4) 节点通信半径  $R_c$  为感知半径  $R_s$  的 2 倍以保证网络连通性, 节点的状态可以分为工作状态和休眠状态, 节点处于休眠状态时不消耗任何能量。

5) 所有传感器节点均为时间同步, 采用 FTSP 来实现节点同步。

6) 节点之间的链路质量基本不变, 且随着时间变换, 链路质量变化的可能性可以低至忽略。

## 1.2 调度目标

假设网络区域  $L$ , 所有传感器节点组成的集合为  $S = \{s_1, s_2, \dots, s_n\}$ , 在满足 2.1 所示的网络假设的前提下, 通过选择能满足网络最大化覆盖的最小数量的节点集, 当节点不能实现网络区域的完全覆盖, 即仍然存在覆盖盲区时, 通过调整最小数量节点的感知半径实现覆盖区域的完全覆盖, 当节点通过其它节点的多跳路径向基站发送数据时, 通过对节点插入苏醒时隙减少节点到节点的端一端的数据传输延迟, 在节点插入苏醒时隙过程中加入剩余能量感知因素, 自适应地根据节点的剩余能量来决定是否加入苏醒时隙, 以在节约数据的能量和降低延迟之间取得平衡, 避免节点的过早死亡。

## 2 基于最大化覆盖的节点调度

### 2.1 基于扫描线的调度节点集选取

为了在最小化能耗目标的前提下实现最大化网络覆盖的节点调度, 需要对节点采用轮换方式进行轮流调度, 将未被调度的节点切换为低功耗的休眠状态, 以减少网络能耗和延长网络寿命。

采用扫描线算法选择最大化覆盖监测区域的节点集, 即对于将监测区域根据行划分为若干栅格, 从第一行开始, 对各个栅格从左到右进行扫描, 选择具有最大覆盖区域的节点进行调度, 其选择过程如下所示:

1) 将网络区域根据传感器节点的最小感知半径  $R_{mi}$  为宽度和高度, 划分为若干栅格;

2) 对网络区域中的每个栅格计算其所包含的各节点所能覆盖的目标区域, 选择具有最大覆盖面积的节点, 并计算其与调度节点集  $S$  中已选的节点的共同覆盖区域;

如果覆盖区域大于某预设阈值, 则重新选择次大覆盖面积的节点进行判断;

否则, 并将其加入调度节点集  $S$ ;

3) 如果当前栅格中没有符合要求的节点, 则当前栅格不选择节点, 转到下一个栅格继续选择;

4) 重复 2) ~4) 直到所有栅格均已遍历完毕, 则输出调度节点集  $S$  中的所有节点作为本轮调度节点集。

### 2.2 覆盖盲区区域

当调度节点集  $S$  中的所有节点不能有效地覆盖感知区域中的所有目标点时, 则存在着覆盖盲区, 覆盖盲区的计算方式是将覆盖区域减去各个传感器覆盖的区域总和: 如果剩余的覆盖区域不为空, 则存在覆盖盲区。如图 1 中的空白处所示。

在图 1 所示的覆盖区域中, 构成覆盖盲区的节点  $A$ 、 $D$ 、 $H$ 、 $I$ 、 $M$ 、 $C$  和  $B$  为盲区边缘节点, 因此, 覆盖盲区可以通过这些字母对于每个盲区边缘节点逆时针方向连接表示, 同时与其具有相同覆盖区域的节点互为邻居边缘节点。

### 2.3 自适应地调整节点感知半径

当调度节点集  $S$  不能全面覆盖监测区域的情况中, 即监测

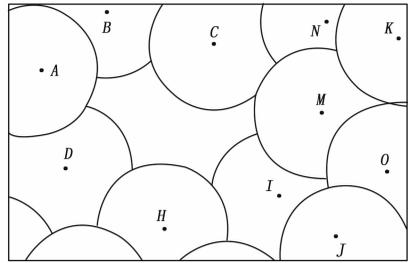


图 1 覆盖区域

区域中存在如图所示的覆盖盲区的情况, 可以通过调整节点的感知半径来实现监测区域的完全覆盖, 其调整感知半径实现覆盖的过程可以描述为:

- 1) 根据 3.2 获得网络监测区域中的所有覆盖盲区;
- 2) 计算每个覆盖盲区上的节点的填补权值:

$$\tau_{s_i} = \frac{|count_{s_i}' - count_{s_i}|}{R_{s_i}' - R_{s_i}} \quad (1)$$

在式 (1) 中,  $R_{s_i}'$  表示节点增加感知半径后的半径值,  $count_{s_i}'$  表示节点的感知半径变化为  $R_{s_i}'$  后, 节点的边缘交叉节点数目。

3) 选择具有最大填补权值的节点, 判断其剩余能量是否大于能量阈值;

如果大于能量阈值, 将当前节点作为半径调整候选节点;

否则, 选择填补权值仅次于最大的节点作为半径调整候选节点;

4) 增加半径调整候选节点的感知半径到  $R_{s_i}'$ , 同时向其邻居边缘节点广播“新感知半径”消息, 同时更新节点的边缘节点集, 构成新的边缘节点集, 构成新的覆盖盲区; 向其邻居边缘节点广播“新的覆盖盲区”消息;

5) 如果覆盖盲区不为空则转入 2);

否则, 算法结束, 当前经过自适应调整感知半径的节点集已能实现网络区域的全面覆盖。

采用本节所示算法流程对图 1 所示的覆盖区域进行盲区消除, 即将节点  $A$ 、 $H$  和  $C$  的感知半径增大, 使得节点实现了网络区域的全面覆盖, 如图 2 所示。

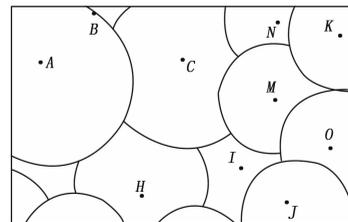


图 2 去除了盲区的覆盖区域

## 3 链路质量保证的节点传输

### 3.1 节点端到端的传输延迟

在获得了调度节点集, 并通过自适应地调整节点的感知半径实现对覆盖盲区的全面覆盖后, 节点就可以在其工作周期内采集数据并将数据发送给其邻居传感器节点。在链路质量非常完美的情况下, 每次数据发送都是成功的, 而数据传输延迟往往是毫米级别, 可以忽略其所耗时间, 但是现实情况下, 链路的状态往往是不完美的, 因此, 节点之间的数据传输也不是每

次都是成功的，因此，需要引入数据重传，直到数据成功发送到目标节点。

如对于下面一段链路，如图 3 所示。

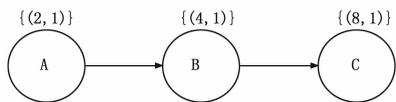


图 3 网络链路质量模型

从图 3 中可以看出，节点 A 通过节点 B 发送数据给节点 C，3 个节点的工作周期均为 8，即为  $8\tau$ ，节点 A 的工作周期表示为  $T_A = \{(2,1)\}$ ，节点 B 的工作周期表示为  $T_B = \{(4,1)\}$ ，节点 C 的工作周期表示为  $T_C = \{(8,1)\}$ ，在链路质量完美的情况下，假设  $\tau$  的值为 1，则节点 A 发送给节点 B 的休眠延迟可以表示为：

$$delay_{AB} = (4 - 1) \cdot \tau = 3s \quad (2)$$

节点 A 与节点 B 之间的休眠延迟可以表示为：

$$delay_{AB} = (4 - 1) \cdot \tau + (8 - 4)\tau = 7s \quad (3)$$

上述计算方式是在链路质量完美的情况下，当链路质量不完美时，当 A 节点与 B 节点之间的链路质量  $p_{AB} = 0.9$  时，当 B 节点与 C 节点之间的链路质量  $p_{BC} = 0.7$  时，则节点 A 与节点 C 构成的链路的休眠延迟可以计算为：

$$delay_{AB} = \sum_{count_{AB}} \sum_{count_{BC}} 0.9 \times (1 - 0.9)^{count_{AB}} \times 0.7 \times (1 - 0.7)^{count_{BC}} \times (2 + 80 \times count_{AB} + 4 + 80 \times count_{BC}) \quad (4)$$

当节点 A 向节点 B 传输数据时的重传次数大于 1 时，即  $count_{AB}$  大于 1 时，可以根据式 (4) 计算延迟，假设  $count_{AB} = count_{BC} = 2$  时，则休眠延迟的期望值可以根据式 (4) 计算为 58.4 s，约为链路质量完美时的 6 倍，而上述计算仅针对两跳路由，当节点数目更多时，网络的链路质量对节点端到端的传输延迟影响是巨大的。

### 3.2 增加节点休眠时隙

为了解决链路质量部不可靠时候节点一端传输延迟过大的问题，可以增加节点的苏醒时隙来减少延迟，如图 4 所示。

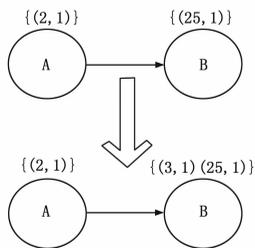


图 4 增加苏醒时隙的传输模型

在图 4 中，链路质量为 0.8，第一行表示在原始状态下，节点 A 发送数据给节点 B 时的休眠延迟为 31.06 s，而第二行中当对节点 B 增加时隙 (3, 1) 后，休眠延迟降低到 8.25 s。

当传输延迟阈值为  $Delay_{th}$  时，采用  $E[D_{ij}^{m,u}(t)]$  表示节点 i 发送数据到节点 j 时的休眠延迟期望值，m 为链路跳数，则自适应地实现节点苏醒时隙的过程可以描述为：

1) 计算在原始情况下的节点 i 发送数据到节点 j 时的休眠延迟期望值  $E[D_{ij}^{m,0}(t)]$ ，如果其小于传输延迟阈值为

$Delay_{th}$ ，则能满足实际要求，不需要额外增加苏醒次数；

2) 对于节点 i 到目的节点 j 之间的连接链路，对节点 j 从增加一次苏醒时隙开始，依次增加时隙至 u，直到  $E[D_{ij}^{m,u}(t)]$  小于  $Delay_{th}$ ；

3) 相邻节点的传输延迟期望值可以描述为：

$$E[D_{ij}^{m,u}(t)] = \begin{cases} E[d_{ij}(t)] & m = 1, h = 0 \\ E[d_{ij}^1(t)] & m = 1, h = 1 \end{cases} \quad (5)$$

4) 对于节点 i 到目的节点 j 之间的传输延迟可以在与相邻节点的传输的延迟的基础上，递归的表示为：

$$E[D_{ij}^{m,u}(t)] = \min \begin{cases} E[D_{ij}^{m,u}(t)] \\ E[d_{ij}(t)] \oplus E[D_{ij}^{m-1,u}(t)] \\ c = 0, m > 0, h > 0 \\ E[d_{ij}^1(t)] \oplus E[D_{ij}^{m-1}(t)] \\ c = 1, m > 0, h > 0 \end{cases} \quad (6)$$

在式 (6) 中，c 表示源节点 i 的邻居节点是否增加苏醒时隙，如果增加，则  $c=1$ ，如果不增加，则  $c=0$ 。

## 4 实验分析

### 4.1 实验场景和参数

在 NS2 环境下进行实验，仿真实验布置在一个  $100\text{ m} \times 100\text{ m}$  的监测区域中，节点的感知半径和通信半径分别为 10 和 22，仿真时间为 500 s，每轮 50 s，在初始时刻节点随机部署在网络区域中，节点的占空比为 1/80，每个节点的工作时隙持续时间为 1 s，链路质量的波动范围为 0.6~0.9，最大重传次数为 5，节点发送数据单位耗能为 1 J，接收数据单位耗能为 0.7 J，节点初始能量为 200 J，数据采集间隔为 12 s。

文中各部分的算法过程参数如下：覆盖区域阈值 95%，剩余能量阈值 20 J，传输延迟阈值为  $Delay_{th}$  为 25。

首先采用基于扫描线的方法确定调度节点集，然后计算其覆盖盲区是否存在，如果存在覆盖盲区，则改变某些节点的感知半径以实现网络区域的全面覆盖。

在确定了能实现全面覆盖的节点集合后，为了进一步减少节点耗能和延长网络生命周期，在链路质量不完美的状态下，通过增加链路中继节点的苏醒时隙实现数据的稳定传输。

为了验证文中方法的优越性，将文中方法与文献 [10] 和文献 [11] 在下面几个方面进行比较：

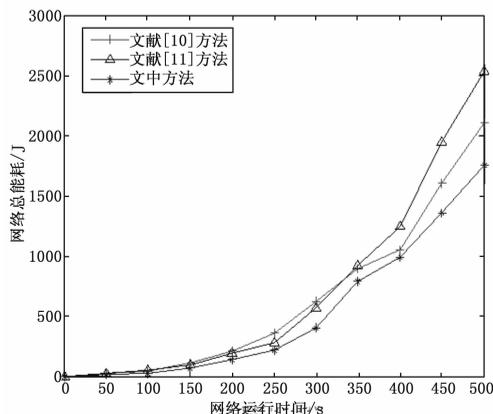


图 5 网络能耗对比

## 4.2 节点能耗

网络总能耗随时间变化仿真结果如图5所示,从图5可以看出,3种方法的网络能耗均随着网络运行时间的推进而逐步增加,但文中方法对应的曲线始终位于另外两种方法的下方,显然,文中方法具有更少的能量开销,文献[10]、文献[11]和文中方法对应的网络运行平均总能量开销分别为837.36 J、720.73 J和520.54 J,这是因为文中方法首先通过扫描线方法选择尽可能少的节点集进行调度,然后尽可能选择剩余能量大的节点来调整感知半径以覆盖盲区,并在节点传输节点,通过增加节点苏醒时隙进一步减少能量开销,因此,具有较小的网络能量开销,从而具有更长的网络生命周期。

## 4.3 休眠延迟期望

为了验证文中方法在不同链路质量下的的休眠延迟期望,对文中方法随着节点个数增加时,休眠延迟期望值变化的情况进行仿真,得到的结果如图6所示。

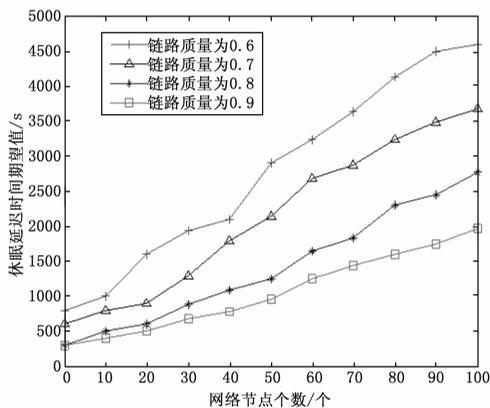


图6 不同链路质量的休眠延迟对比

从图6中可以看出,链路质量越高,休眠延迟时间越少,当链路质量为0.6时,随着节点数的增加,休眠延迟增加快速,而当链路质量为0.9时,休眠延迟增加缓慢,这说明了链路质量对休眠传输延迟的影响。

为了进一步验证文中方法的优越性,在链路质量为0.8的情况下,将3种方法的休眠延迟时间进行对比,结果如图7所示。

从图7中可以看出,3种方法的休眠延迟时间的比较,从图中可以看出,文中方法的传输休眠延迟远远小于另外两种方法,较其分别减少56.75%和48.92%,这是因为文中方法在节点调度时考虑了休眠延迟,并通过增加多跳路由中节点的苏醒时隙来最大化地减少节点的休眠传输延迟。

## 5 结论

为了实现目标区域中的全面覆盖和延长网络的生命周期,文中提出了一种适用于无线传感器网络的传感器节点调度方法。在对网络模型进行描述的基础上,采用扫描线方法来获取调度的节点集,当选择的调度节点不能实现网络区域的全面覆盖时,自适应地调整节点的感知半径以减少覆盖盲区,从而在实现网络全面覆盖的前提下,降低网络的总能耗开销,同时针对节点到基站的多跳路由中,当链路质量不完美时的休眠延迟

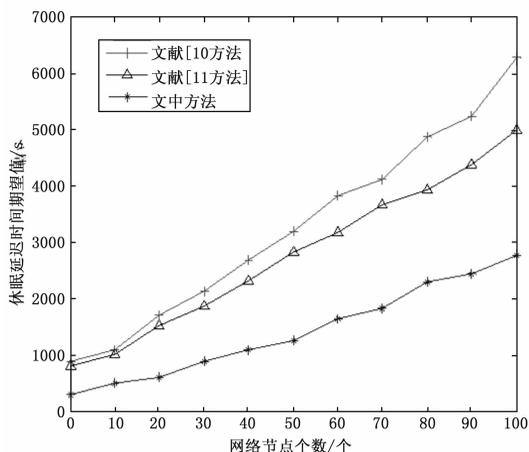


图7 不同方法的休眠延迟时间对比

过大的问题,设计了一种增加相邻节点的苏醒时隙以减少网络传输延迟的方法。通过仿真实验证明文中方法能保证区域的全面覆盖,使得文中方法有较少的网络能量开销和较短的休眠传输延迟,较其它方法具有较大的优越性。

## 参考文献:

- [1] Wang F, Liu J C. On reliable broadcast in low duty-cycle wireless sensor networks [J]. IEEE Trans. on Mobile Computing, 2012, 11 (5): 767-779.
- [2] 魏全瑞, 刘俊, 韩九强. 改进的无线传感器网络无偏距离估计与节点定位算法 [J]. 西安交通大学学报, 2014, 6 (48): 1-6.
- [3] Nath S, Ekambara V N, Kumar A, et al. Theory and algorithm for hop-count-based localization with random geometric graph models of dense sensor networks [J]. ACM transactions on sensor networks, 2012, 8 (4): 111-152.
- [4] Kumar S, Lobiyal D. An advanced DV-hop localization algorithm for wireless sensor networks [J]. Wireless personal communications, 2012, 71 (2): 1365-1385.
- [5] 任秀丽, 王伟勇. 基于空间分辨率的无线传感网节点调度策略 [J]. 计算机应用, 2013, 33 (8): 2108-2111
- [6] Gu Y, He T. Bounding communication delay in energy harvesting sensor networks [A]. In: Proc. of the ICDCS [C]. 2010. 837-847.
- [7] Li Z J, Li M, Liu J L, et al. Understanding the flooding in low-duty-cycle wireless sensor networks [A]. In: Proc. of the Parallel Processing (ICPP) [C]. IEEE Computer Society, 2011. 673-682.
- [8] 景伟娜. K覆盖保证的异构传感器网络节点调度设计 [J]. 计算机测量与控制, 2013, 21 (4): 1009-1101.
- [9] 卢旭, 程良伦. 物联网感知层高效覆盖优化节点调度算法 [J]. 计算机应用研究, 2013, 5 (30): 1458-1460.
- [10] 杨詠, 汪文勇, 唐勇. 基于通信与感知覆盖的WSNs节点调度算法 [J]. 计算机科学, 2013, 7 (40): 54-58.
- [11] 单洪, 马涛. 层次传感器网络的交错同步休眠调度方法研究 [J]. 传感技术学报, 2012, 7 (25): 980-984.