

物联网感知层高效覆盖优化节点调度算法*

卢旭^{1,2}, 程良伦¹

(1. 广东工业大学 自动化学院, 广州 510006; 2. 广东技术师范学院 自动化学院, 广州 510006)

摘要: 为了更好地节约能量并有效解决节点不均等休眠问题,探讨了基于容忍覆盖区域的节点调度算法,并在此基础上进行了两方面改进:通过引入相对剩余能量水平参数 E_{remain} 优化覆盖冗余判断策略,并通过增加“预活动”和“回退”两种节点状态来改进节点状态分布情况,从而提出一种高效覆盖优化节点调度算法 ECO-NS。最后运用 MATLAB 对该算法进行了验证,结果表明,相对于同类算法,该算法有效提升了网络覆盖质量,延长了网络寿命。

关键词: 物联网; 感知层; 节点调度; 高效; 覆盖优化

中图分类号: TP393; TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2013)05-1458-03

doi:10.3969/j.issn.1001-3695.2013.05.044

Energy-efficient coverage optimized node scheduling algorithm for sensor layer in Internet of things

LU Xu^{1,2}, CHENG Liang-lun¹

(1. College of Automatization, Guangdong University of Technology, Guangzhou 510006, China; 2. College of Automatization, Guangdong Polytechnic Normal University, Guangzhou 510006, China)

Abstract: For saving energy and solving the problem of nodes non-uniform sleeping, this paper discussed the tolerance coverage area based node scheduling algorithm, and then proposed an energy-efficient coverage optimized node scheduling algorithm by improving this algorithm with two ways: improving the coverage redundancy judgment strategy by introducing a relative surplus energy level parameters E_{remain} , and optimizing node state distribution by increasing two node states pre-active and backoff. It used the network simulator MATLAB to demonstrate the performance of the algorithm. Experiment results show that this algorithm effectively improves the network coverage quality and prolongs the network lifetime in comparison with the similar algorithm.

Key words: Internet of things (IOT); sensor layer; node scheduling; energy-efficient; coverage optimizing

物联网 (IOT) 是把具有标志、感知和智能处理能力的物体,借助通信技术互连而成的网络^[1]。物联网的层次结构由下向上可分为感知层、接入网络层、中间件层和应用层四层。感知层处于最底层,它将各种设备上的传感器通过有线或无线的方式连接起来形成一个信息采集与控制的网络。无线传感器网络是物联网感知层的基本形态,在数据传输过程中,首先在最低层由传感器采集各类监控信号,经过感知层的汇聚节点或基站传输到上一层网络空间。

覆盖质量是物联网感知层对物理世界感知能力的体现,是常用来描述监测服务质量 (quality of service, QoS) 的标准^[2]。把网络中所有工作节点构成的监测区域面积占整个监控区域 M 面积的比例,称为网络覆盖质量^[3],即

$$\eta = \frac{\text{area}((U_{i \in \varphi} S_i) \cap M)}{\text{area}(M)} \quad (1)$$

其中: φ 表示所有工作节点的集合; $\text{area}(M)$ 表示整个监控区域的面积; S_i 表示节点 i 的监测区域; $\text{area}((U_{i \in \varphi} S_i) \cap M)$ 表示所有工作节点构成的监测区域与 M 重叠部分的面积。

节点调度是在网络密度较大或允许部分节点暂时休眠的

情况下,利用各种调度算法^[4-9]使节点在不同状态间进行转换,在满足网络覆盖质量要求的前提下实现轮流关闭部分冗余节点,从而达到降低节点能耗、延长网络寿命的目的。

1 相关工作

基本的节点调度算法是根据节点的地理位置信息计算覆盖信息,但由于地理位置信息的获得需依赖于外部基础设施 (如 GPS、有向天线等),这大大增加了节点的硬件成本和能量消耗,同时也会存在定位不精确的问题,从而造成覆盖控制算法很难准确地计算出节点间的覆盖关系。因此,无须地理位置信息的节点调度算法逐渐成为研究热点。

Kumar 等人^[5]提出了随机独立调度方法 (randomized independent scheduling, RIS),即在节点间实现时间同步的基础上周期性地对节点进行调度,在每个周期中每个节点独立地以概率 p 进入工作状态,以概率 $(1-p)$ 进入睡眠状态。该方法实现简单,但无法保证网络的完全覆盖和连通。Ye 等人^[6]提出的环境监测自适应休眠方法 (probing environment and adaptive sleeping, PEAS) 是一种通过观察邻居节点变化来调整节点状态的

收稿日期: 2012-09-12; **修回日期:** 2012-11-06 **基金项目:** 国家自然科学基金广东省联合基金重点项目 (U0935002); 广东省中国科学院全面战略合作项目 (2010B090301041); 省部产学研结合项目 (2010B090400211)

作者简介: 卢旭 (1986-), 男,江西修水人,讲师,博士研究生,主要研究方向为物联网、无线传感器网络 (bruda@126.com); 程良伦 (1965-), 男,教授,博导,主要研究方向为物联网、无线传感器网络、RFID 技术。

调度方法,每个睡眠节点定期检查其探测区域内是否有工作节点,若没有则进入工作状态,否则继续维持睡眠状态。Corpa 等人^[7]提出的自适应自配置拓扑控制方法(adaptive self-configuring sensor networks topologies, ASCENT)通过获取邻居工作节点数目和测量通信丢包率来确定是否参与数据传输。PEAS 和 ASCENT 中某些节点可能持续工作,导致其过早死亡,从而影响覆盖质量。基于节点均匀分布的假定,Wu 等人^[8]提出了一种基于部署特征的轻量级节点调度方法(lightweight deployment-aware scheduling, LDAS),该方法通过获知邻居节点的数量来实现概率覆盖,但该方法忽略了节点的两跳邻居对其感知区域的覆盖作用,导致其选取的工作节点中仍有大量冗余节点。

上述位置无关的节点调度算法中,由于监测区域中所有节点采用相同的覆盖冗余判别算法,处于监测区域边界上的节点由于邻居节点有限,较难达到睡眠条件,会长时间处于活动状态,从而导致节点能耗过大而过早死亡,进而引起节点对监测区域的覆盖面积向监测中心收缩的现象,即形成不均等休眠问题。在物联网感知层监测应用中,并不希望发生这种现象。针对这种情况,结合节点覆盖模型的特点,文献[9]提出节点容忍覆盖区域的概念,得出节点的覆盖区域与容忍覆盖区域的关系,进而提出基于容忍覆盖区域的节点调度(tolerable coverage area based node scheduling, TCA-NS)算法。

2 TCA-NS 算法

TCA-NS 算法采用分轮调度机制,每轮分为邻居发现、覆盖冗余判别和节点调度三个阶段。在邻居发现阶段,各节点在其通信范围内通过广播邻居发现消息获得其邻居节点信息(距离、邻居节点个数等信息)。在覆盖冗余判别阶段,定义以下容忍覆盖区域概念:半径为 r 的节点随机部署在半径 R 的圆形监测区域 A 内($r < R$),则节点在该部署区域内的容忍覆盖区域为

$$E_r[C] = \pi \frac{r^2}{R^2} (R-r)^2 + \frac{2}{R^2} \int_{R-r}^R C'(l) l dl \quad (2)$$

其中: $C'(l)$ 表示分布在监测区域的边界区域的节点覆盖面积。利用以上容忍覆盖区域来判别覆盖冗余,节点 i 通过邻居节点个数或距离等信息(采用如文献[8,10]的距离和节点个数计算方法获得),得到节点 i 被邻居节点覆盖的面积为 B_i 。TCA-NS 算法采用容忍覆盖区域对节点进行覆盖冗余判别的方式为

$$B_i/E_r[C] \geq \text{Threshold} \quad (3)$$

其中:Threshold 表示满足网络应用的节点覆盖冗余的门限值。若上式成立,则节点 i 满足覆盖冗余。

在节点调度阶段,TCA-NS 算法将节点的状态分成活动状态、预睡眠状态、睡眠状态三种。三种状态按图1形式相互转换。

文献[9]通过仿真实验从算法的有效性、部署节点个数与活动节点个数关系、网络覆盖比率与网络生存时间关系三个方面验证了 TCA-NS 算法的性能;证明其在不牺牲网络任何性能的前提下可有效缓解不均等休眠问题,减少边界因素对节点调度带来的影响,从而延长网络生存时间。但 TCA-NS 算法未考虑节点剩余能量及节点分布差异对算法的影响。如某节点的剩余能量过低,根据文献[11]中的能量模型,即节点在传输、空闲和休眠状态的能量消耗比率为 20 : 4 : 0.01,则应较多地让该节点处于休眠状态,以增长其存活时间。又如网络中节点密度大到一定程度,绝大多数节点都会满足冗余节点判断条件,此时所有满足条件的节点都会尝试进入睡眠状态,算法没

有很好的策略来避免互为邻居的多个节点同时进入睡眠状态的情况,会造成覆盖盲区,形成覆盖洞,从而降低覆盖质量。

3 高效覆盖优化节点调度算法

根据以上对 TCA-NS 算法的分析,从以下两个方面对其进行改进,提出一种高效覆盖优化节点调度(energy-efficient coverage optimized node scheduling, ECO-NS)算法。

a)在覆盖冗余判别阶段,增加节点剩余能量作为判别因素。将节点在整个网络当中的相对剩余能量水平 E_{remain} 作为参数($0 \leq E_{\text{remain}} \leq 1$), $E_{\text{remain}} = 0$ 时表示节点已死亡, $E_{\text{remain}} = 1$ 时表示该节点剩余能量是网络中所有节点剩余能量的最大值。在进行覆盖冗余判别时,采用下式进行判别:

$$E_{\text{remain}} \times B_i/E_r[C] \geq \text{Threshold} \quad (4)$$

相对于采用 TCA-NS 算法,在相同的网络环境下,将阈值 Threshold 适当减小,这样便可使处于活动状态的节点总数维持不变,从而维持总的网络覆盖率。

b)在节点调度阶段,所有冗余节点的功能模块被关闭以节省网络能量,剩余的活跃工作节点进行正常的监测和通信。为了避免上述由节点密度过大等节点分布差异带来的覆盖不均、能量浪费等问题,本文对 TCA-NS 算法中每个节点的状态分布进行改进,将其原本的三种运行状态值增加为五种,分别为活动状态(active)、预活动状态(pre-active)、回退状态(back-off)、预睡眠状态(pre-sleep)、睡眠状态(sleep)。五种状态之间的转换关系如图2所示。

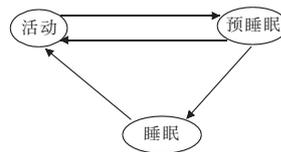


图1 TCA-NS 节点状态转换图

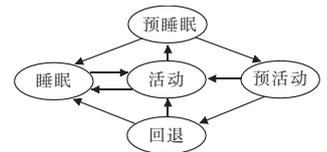


图2 ECO-NS 节点状态转换图

在节点密度很大时,ECO-NS 算法采用预先降低工作节点密度的方法来防止大量节点由于相互之间的判断干扰同时进入睡眠状态。在状态转换之前,先随机选取适量节点作为候选活动节点,后面的节点调度算法则在这些候选活动节点中运行,其余非候选活动节点则直接进入睡眠状态。另外,当处于预活动状态的节点在一定时长内满足冗余节点判断条件时,转换成回退状态,在回退状态下再在一定时长内满足冗余节点判断条件时转换成睡眠状态,否则转换成活动状态。这就有效提高了算法在复杂网络环境下的灵活性,能够在控制能耗的情况下合理优化网络覆盖质量。一轮调度结束后,所有睡眠节点转换成活动状态,重新开始新一轮调度。节点调度过程的状态转换算法如下:

```

while( state != active && state != sleep) {
switch( state) {
case pre-sleep: {
delay t1;
if( satisfy redundancy conditions) {
state = sleep;
broadcast state change message to neighbors;
}
else state = pre-active;
break; }
case pre-active: {
delay t2;
if( satisfy redundancy conditions) state = backoff;
else state = active;
break; }
case backoff: {

```

```

delay t3;
if(satisfy_redundancy_conditions) {
state = sleep;
broadcast state change message to neighbors;
else state = active;
default; } }

```

4 仿真与分析

为验证算法的正确性和有效性,采用 MATLAB 作为实验平台对该算法进行分析,考虑算法的改进特点,主要从网络覆盖质量和网络存活时间两个方面来验证算法的性能。所有节点随机分布在 100 m × 100 m 的正方形区域内,节点总数为 400,所有节点的感知半径为 10 m。节点的能耗模型采用文献 [11] 中的物理节点能耗模型,节点休眠的时间设为 5 s。采用与文献 [8] 中 LDAS 相同的覆盖分析模型对 ECO-NS 算法进行仿真,将在该分析模型上运行的 ECO-NS 算法称为 ECO-LDAS,并在网络性能上与同样采用此模型的 LDAS 和 TCAI-LDAS^[9] 两种算法进行比较分析。

4.1 网络覆盖质量

本文通过设置不同的覆盖冗余阈值,分别考察网络在活动节点数为 30、40、50、60、70、80 时的网络覆盖质量(由于活动节点数无法精确控制,实际数量存在少许误差,但不影响实验整体分析结果)。在网络的有效寿命期内,网络的覆盖质量情况如图 3 所示。每一个取样数据都是在相应实验参数下,随机选取 10 个有效数据求得的平均值。

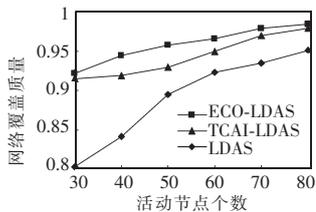


图3 网络覆盖质量比较

由图 3 可见,三种算法的网络覆盖质量都会随活动节点数的增加而提升,但相对于 LDAS 算法,ECO-LDAS 和 TCAI-LDAS 算法在活动节点数相同的情况下,整个网络的覆盖质量更高。这是由于 ECO-LDAS 和 TCAI-LDAS 算法通过执行基于容忍覆盖区域的覆盖判别后,使得靠近监测区域边缘的处于休眠状态的节点增加了,充分利用了每个节点的覆盖能力,通过更少节点就能够满足应用覆盖要求。而 ECO-LDAS 由于采用了节点状态转换优化算法,使得在高密度下节点的睡眠机制启动更加合理,并有效避免了同一监测区域所有节点同时睡眠造成的感知覆盖洞问题,从而使其相对于 TCAI-LDAS 具有更高的覆盖质量。图 4 给出了在覆盖质量要求分别为 80%、85%、90%、95% 及 97.5% 时,三种算法各自所需的节点数量情况。可以看出,ECO-LDAS 算法比另外两种算法只需更少的节点处于活动状态,即可保持相同的覆盖质量。

4.2 网络存活时间

图 5 给出了在覆盖质量要求为 90% 的条件下,三种算法的存活节点数随着网络运行时间的变化情况。由图可以看出,在相同的网络覆盖质量要求下,由于 TCAI-LDAS 算法充分利用了网络覆盖冗余特性,处于休眠状态节点的个数要多于 LDAS 算法,使得 TCAI-LDAS 的网络存活时间要高于 LDAS。而 ECO-LDAS 算法的网络存活时间又要高于 TCAI-LDAS,而且

从节点的死亡时间跨度来看 ECO-LDAS 也要明显小于 TCAI-LDAS。这是因为 ECO-LDAS 在覆盖冗余判别阶段,增加节点剩余能量作为判别因素,使得节点间能耗分布更加均匀,从而有效防止了某些节点因为经常使用过早死亡,也有效延长了网络寿命。

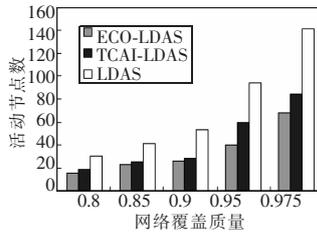


图4 活动节点数比较

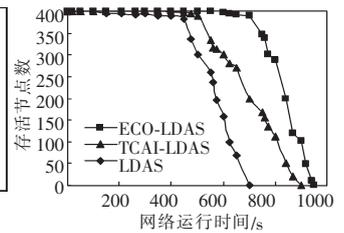


图5 覆盖质量要求90%时的网络节点存活情况

5 结束语

本文针对文献 [9] 提出的基于容忍覆盖区域的节点调度算法进行改进,提出一种新的高能效覆盖优化节点调度算法。该算法一方面在进行覆盖冗余判别时增加节点剩余能量作为判别因素,另一方面对节点的状态分布和转换进行了相关改进。仿真实验结果表明,相对于同类其他算法,该算法不仅解决了节点“不均等休眠”问题,而且在提升网络覆盖质量的同时,有效延长了网络寿命。下一步工作将对算法中覆盖冗余判别阈值的最优取值问题进行研究。

参考文献:

- [1] Commission of the European Communities. Internet of things: an action plan for Europe [M]. [S. l.]: Office for Official Publications of the European Communities, 2009: 1-12.
- [2] HE Xin, YIN Ke, GUI Xiao-lin. The area coverage algorithm to maintain connectivity for WSN [C] // Proc of the 9th IEEE International Conference on Computer and Information Technology. Washington DC: IEEE Computer Society, 2009: 81-86.
- [3] 王换招, 孟凡治, 李增智. 高效节能的无线传感器网络覆盖保持协议 [J]. 软件学报, 2010, 21 (12): 3124-3137.
- [4] XU Xiao-hu, LI Xiang-yang, WAN Peng-jun, et al. Efficient scheduling for periodic aggregation queries in multihop sensor networks [J]. IEEE/ACM Trans on Networking, 2012, 20 (3): 690-698.
- [5] KUMAR S, LAI T H, BALOGH J. On k-coverage in a mostly sleeping sensor network [C] // Proc of the 10th Annual Int'l Conference on Mobile Computing and Networking. New York: ACM Press, 2004: 244-258.
- [6] YE Fan, ZHONG G, CHENG J, et al. PEAS: a robust energy conserving protocol for long-lived sensor networks [C] // Proc of the 23rd Int'l Conference on Distributed Computing Systems. New York: IEEE Press, 2003: 28-37.
- [7] CORPA A, ESTRIN D. ASCENT: adaptive self-configuring sensor networks topologies [J]. IEEE Trans on Mobile Computing, 2004, 3 (3): 272-275.
- [8] WU Kui, GAO Yong, LI Fu-lu, et al. Lightweight deployment-aware scheduling for wireless sensor networks [J]. Mobile Networks and Applications, 2005, 10 (6): 837-852.
- [9] 凡高娟, 王汝传, 黄海平, 等. 基于容忍覆盖区域的无线传感器网络 [J]. 电子学报, 2011, 39 (1): 89-94.
- [10] TIAN D, GEORGANAS N D. Location and calculation-free node-scheduling schemes in large wireless sensor networks [J]. Ad hoc Networks, 2004, 2 (1): 65-85.
- [11] HILL J, SZEWCAYK R, WOO A, et al. System architecture directions for networked sensors [J]. Architectural Support for Programming Languages and Operating Systems, 2000, 35 (11): 93-104.