

基于时间同步的无线传感器网络覆盖控制优化算法^{*}

常光强, 樊晓平, 刘少强

(中南大学 信息科学与工程学院, 长沙 410001)

摘要: 为了实现网络覆盖控制的优化, 减少能耗, 提出了一种分布的、高效节能、与节点位置无关的传感器网络覆盖算法。在该算法中, 节点与邻居交换信息, 并通过能量大小竞选工作节点, 节点根据自身与工作节点的距离判断决定进入工作状态或休眠状态, 并采用在生成树中广播时间同步算法使工作节点网络达到时间同步。仿真结果比较表明, 该算法能够明显减少工作节点数, 从而减少能量消耗, 延长网络寿命。

关键词: 无线传感器网络; 覆盖问题; 与位置无关; 覆盖优化

中图分类号: TP393; TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2012)01-0035-03

doi:10.3969/j.issn.1001-3695.2012.01.009

Coverage control optimization algorithm in wireless sensor networks on time synchronization

CHANG Guang-qiang, FAN Xiao-ping, LIU Shao-qiang

(College of Information Science & Engineering, Central South University, Changsha 410001, China)

Abstract: In order to achieve optimal control of network coverage, reduce energy consumption, this paper presented a distributed, energy efficient, and the node position independent of the sensor network coverage algorithm. In the algorithm, nodes exchanged information with neighbors and worked through the energy of running nodes, other nodes according to their distance from the node with the work, made judgments, or to enter the work state, or into sleep state. Simulation results show that the algorithm can significantly reduce the number of nodes working, thus to reduce energy consumption and prolong network lifetime.

Key words: wireless sensor networks(WSN); coverage problem; location-unaware; coverage optimization

0 引言

无线传感器网络(wireless sensor network, WSN)的覆盖控制,就是在传感器网络节点能量、无线网络通信带宽、网络计算处理能力等资源普遍受限的情况下,通过网络传感器节点放置以及路由选择等手段,最终使无线传感器网络的各种资源得到优化分配。选择合适的无线传感器网络覆盖控制策略,有助于网络节点能量的有效控制、感知服务质量的提高和整体生存时间的延长^[1-3]。

作为一种分布式系统,WSN 时间同步是至关重要的。时间同步为建立许多 WSN 的应用提供了基础的服务。例如,传感器节点间的协调和协作需要时间同步。传感器节点需要在同一时刻休眠和唤醒,否则多个节点同时进入休眠状态会使网络出现盲区。

近年来已有一些学者开展了 WSN 优化覆盖控制方面的研究工作,并取得了一定的进展。Ye 等人^[4]提出了基于探测(probing)的密度控制算法 PEAS。PEAS 算法要求每个睡眠节点定期地在其探测范围内探测邻居节点的状态,若在其探测范围内没有工作节点,则进入工作状态;否则仍处于睡眠状态。显然,PEAS 算法中的某些节点可能持续工作,导致过早死亡,由于网络中的节点能耗不均匀将影响覆盖质量。Zhang 等人在文献[5]中讨论了网络覆盖和连通性,提出了一种分布式的

节点密度控制算法 OGDC,算法要求节点根据其邻居信息和自己的位置信息计算与邻居节点的覆盖关系。Xu 等人在文献[6]中提出了 GAF (geographical adaptive fidelity)算法。在 GAF 算法中,每个节点根据其位置信息将整个区域划分为若干个虚拟单元格(grid),其大小必须保证相邻格子中任何一对节点可以直接通信。但是 GAF 算法没有考虑到实际网络中节点之间距离的邻近并不能代表节点之间可以直接通信的问题,也不能保证节点能耗均匀。毛莺池等人在文献[7]中提出了 DELIC 协议,协议中的节点与邻居交换信息,并通过能量大小竞选工作节点,其他未竞选成功的节点关闭通信设备。

目前的大部分覆盖控制算法都需要节点的精确位置信息以计算节点间的覆盖关系,在选取活动过程中产生较大的计算和控制开销。本文主要研究了时间同步以及与位置无关的无线传感器网络优化覆盖控制策略,无须节点的位置信息,从而减少了整个网络额外的能量消耗。

1 网络模型的建立和问题陈述

1.1 网络模型的建立

假设 N 个传感器节点随机均匀地部署在一个 $L \times L$ 的二维正方形区域 M 内,节点密度要求足够大,当所有传感器节点都处于工作状态时,能够对整个区域完全覆盖且整个网络是连

通的,并假设该无线传感器网络具有以下性质:

a) 节点部署方式。节点是同构的,节点采用随机部署,且部署后节点不能移动。

b) 节点通信范围。相对于节点感知范围而言,监测区域 M 足够大 ($R_s \ll L$),边界因素可以忽略。

c) 节点感知模型。节点采用布尔感知模型,即每个节点的感知范围是以节点为圆心、传感半径 R_s 为半径的圆,在感知范围内的所有节点都可以被感知,否则不被感知。

d) 节点位置信息。每个节点无须装备 GPS,且不能通过测量或定位方法获得其具体的物理位置。

1.2 问题陈述

无线传感器网络覆盖控制优化的主要目的是调度节点的休眠间隔时间,满足覆盖要求下保持部分节点为工作状态,但网络仍能提供连续的检测服务。

服务质量期望值 q (desired QoS) 表示了网络覆盖率。为所有工作节点构成的监测区域面积占整个区域 M 面积的比例,即:

$$q = \frac{((\bigcup_{i=1}^k SR_i) \cap M)_{area}}{M_{area}}$$

假设在监测区域 M 中随机均匀部署 N 个节点,每个节点 $S_i (1 \leq i \leq N)$ 的监测范围为 SR_i ,需要从 N 个节点中最少选取多少个工作节点,使得这些工作节点能够满足应用期望的服务质量 (其中 K 是最少选取的工作节点数量),即:

$$\frac{((\bigcup_{i=1}^k SR_i) \cap M)_{area}}{M_{area}} \geq q, \text{ 以及如何选取 } K \text{ 个工作节点。}$$

工作节点的选取应考虑节点的能量大小。由于在每轮中节点的能量开销不一致,需要算法保证能量开销被均匀地分布到每个节点上,避免某些节点过早死亡。

如果网络中的节点时间不同步也会对覆盖强度造成影响,假设节点时钟与标准时钟差在区间 $(-T/2, T/2)$ 上均匀分布,即 $\Delta t \sim U(-T/2, T/2)$,当节点分布满足 Poisson 点过程时,网络覆盖强度^[8]为

$$C_n' = C_n - \left\{ \frac{2k}{\lambda \|M\|} \times [\exp(-\frac{\lambda \|M\|}{2k}) - \exp(-\frac{\lambda \|M\|}{k})] - \exp(-\frac{\lambda \|M\|}{k}) \right\}$$

其中: $\|M\|$ 表示区域 M 的面积, λ 表示 Poisson 点过程强度。

2 算法描述

2.1 工作节点选取机制

算法中传感器节点共有四种状态:准备状态 (ready)、工作状态 (active)、侦听状态 (listening)、休眠状态 (sleep)。节点状态转换如图 1 所示。

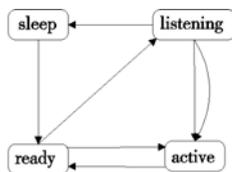


图1 节点状态转换

在算法每轮开始时,所有节点都处于准备状态 (ready),首先,从所有节点中随机选取一个根节点进入工作状态。首个节点的随机选择机制:计算自己此轮中被选中的阈值 $T(n)$,与文

献[9]选取簇头的机制类似,阈值大于事先设定值则选为工作节点。

$$T(n) = \frac{P_{work}}{1 - P_{work} \times [\text{Round} \times \text{mod}[\frac{1}{P_{work}}]]} \times \left[\frac{E_{curr}}{E_{max}} + [\text{Round}_s \times \text{div}[\frac{1}{P_{work}}]] \times [1 - \frac{E_{curr}}{E_{max}}] \right]$$

其中: P_{work} 为被选作工作节点的概率,即 $P_{work} = K/N$; Round 为算法运行的轮数; Round_s 表示节点连续没有被选为工作节点的轮数,当节点成为工作节点后, Round_s 重置为 0。在这个模型中,每一个节点通过由自身内部的晶体振荡器提供的本地计时器记录自己的时间。根节点是所有节点中选出来的可以作为时间基准的节点。这个节点可以访问外部时钟并能够与物理时钟达到同步,根节点层号为 0。同时工作节点向其邻居节点广播自身信息 (self-message, SM)。SM 信息包括节点 ID 信息、status 和层号。收到 SM 信息的节点检测与 active 节点之间的距离。

若 $0 < d \leq R_s$,为了使整个网络的能耗均衡,避免个别传感器节点提前死亡,传感器节点由 ready 状态以概率 $P_{active} = P_{init} E_{current} / E_{max}$ 转为侦听状态,成为侦听节点。其中, P_{init} 为工作节点的初始概率。

若 $R_s < d < 2R_s$,则传感器节点由 ready 状态转为 active 状态。为了避免在这一轮中有过多的邻居节点被选为工作节点,在此引入避让 (back-off) 机制,分别为每个开始节点设置一个随机时间 T_d 。若节点在 T_d 时间内没有收到其他开始节点的 SM 信息,则将自身状态激活为 active 状态,并向其邻居节点发送 SM 信息;若在 T_d 时间内又收到其他邻居节点的 SM 信息,则取消随机时间,并再次判断与此开始节点的距离。若 $d < R_s$ 则进入 listening 状态;若 $d > R_s$ 则进入 active 状态,并向其邻居节点发送 SM 信息。

每一个根节点临近的邻居节点收到 SM 信息后,它们就知道父节点的标志符和层号。接着,记录下标志符,给自己设置一个比父节点层号大 1 的层号,然后给根节点发送一个反馈信息,表明它们已经加入根节点。同理,其他节点根据上述条件决定自身的状态和层号,它们邻近的邻居将得到 SM 信息并加入生成树。这个过程一直进行,直到 WSN 中所有的节点都加入到生成树为止。然而一个已经加入到生成树的节点还有可能再一次接收到其他的 SM 信息,如果有这种情况,若处于 listening 状态下的节点收到 SM 信息后,邻居节点中工作节点数加 1,层号信息不变。图 2 展示了生成树的结果。当步骤 1 结束时,所有的节点处于 active 或 listening 状态。

Listening 状态节点检测其 $d \leq R_s$ 范围内有无工作节点,若没有则将自身激活为 active 状态;若 R_s 范围内有工作节点,则其工作状态不变,等待进一步的判断。直到所有传感器的感知范围内有一个工作节点或本身是一个工作节点,层号以及根节点信息不变。

为了满足网络的覆盖要求,需要进一步从 listening 节点中选取节点进入 active 状态。根据现有的节点冗余度数学模型^[10],对节点分布进行进一步选取,当邻居节点数量大于阈值 α 时,说明节点为冗余节点,可以关闭通信设备,进入 sleep 状态;否则节点成为工作节点,保证网络的覆盖率,阈值 α 的大小与应用期望的服务质量 q 有关。为避免多个节点同时进入工作状态,同理可引入避让 (back-off) 机制。

当整个传感器网络中的节点通过以上算法达到确定状态(进入休眠状态或工作状态)之后,将在 T_w 时间内保持整个状态不变,经过 T_w 后,所有节点被重新唤醒,所有传感器节点重新在下一轮的邻居节点集建立和工作节点的选取。

2.2 在生成树中的广播时间同步算法

在生成树建立起来后,整个 WSN 已经被分为几个可以看到的子树,如图 3 所示。

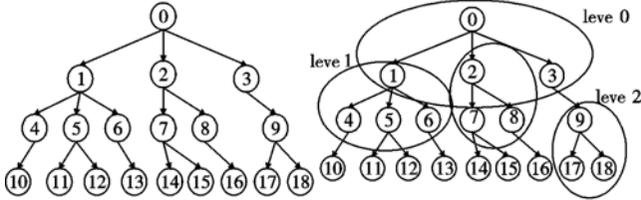


图2 生成树的结果

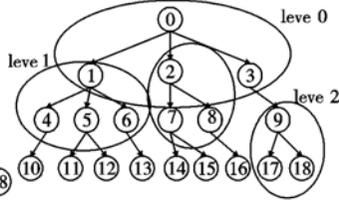


图3 整个网络中的子树

具有精确外部时钟的根节点将同步它的子树根节点 1、2 和 3。在 1、2、3 重置它们的时间为标准时间后,它们会同步其子树的节点。这种子树时钟同步过程会持续下去,直到所有节点设定与根节点一样的时钟为止。每次的子树同步过程如图 4 所示。

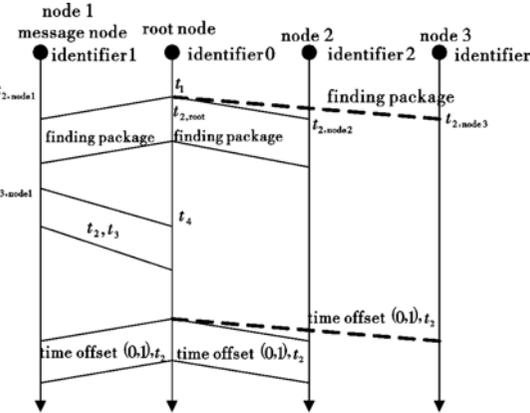


图4 基于生成树的广播同步

每个节点时钟都不相同,因此每两个节点间都会有时间偏差,而只有根节点具有标准的时间。起初根节点向它的子树节点 1、2、3 广播一个同步包,假如这个同步包包含节点 1 的标志符,这意味着根节点选择节点 1 来进行一对同步,因此称节点 1 为消息节点。假设根节点的子节点会在同一时刻接收到同步信息包。在节点 1 接收到同步包后,它知道了包中的标志符与它自己的一样,节点 1 记录接收时间 $t_{2,node1}$,然后开始处理这个包。在很短的时间后,在节点 1 给根节点发送返回包之前,记录瞬时时间 $t_{3,node1}$,并且把 $t_{2,node1}$ 和 $t_{3,node1}$ 的信息嵌入到返回包中。当根节点接收到这个信息时也记录时间 t_4 。于是根节点可以计算出与节点 1 的时间偏差如下:

$$O_{R,node1} = [(t_{2,node1} - t_1) - (t_4 - t_{3,node1})] / 2 \quad (1)$$

其中: $O_{R,node1}$ 表示了时间偏差。最后根节点再次广播一个包含与节点 1 的偏差和节点 1 的时间信息 $t_{2,node1}$ 的包。当然节点 1 基于偏差 $O_{R,node1}$ 就能够调整自己的时钟了。

另外的节点利用这个数据包信息也可以知道它们与根节点的偏差。例如,对于节点 2,有

$$t_{2,root} + O_{R,node1} = t_{2,node1} + O_{node1,node2} \quad (2)$$

其中: $O_{node1,node2}$ 表示节点 1 与 2 的时间偏差。于是可以得到

$$O_{R,node2} = O_{R,node1} - (t_{2,node1} - t_{2,node2}) \quad (3)$$

因为节点 2 会接收包括 $t_{2,node1}$ 和 $O_{R,node2}$ 的广播包,同时,节

点 2 也记录了时间信息 $t_{2,node2}$,于是 $O_{R,node2}$ 可以通过式(3)计算得到。在节点 2 知道了 $O_{R,node2}$ 后,它将调整自己的时钟与根节点达到同步。余下的节点如节点 3 可以通过同样的方式与根节点达到同步^[11]。当子树中的所有子节点都与父节点同步之后,子树的同步就实现了。很明显,与根节点同步的节点 1、2 和 3 也可以被看做它们子树的父节点。因此,类似的子树时间同步过程一直进行,直到整个 WSN 都同步到根节点。

3 仿真结果与评价

为了验证该算法选出的工作节点分布是否良好,在 $50\text{ m} \times 50\text{ m}$ 区域中,随机部署 100 ~ 1000 个节点,应用服务期望 $q = 99\%$, $R_s = 10\text{ m}$, $R_c = 20\text{ m}$ 。节点分布情况如图 5、6 中圆点所示,图中圆盘代表工作节点的覆盖范围。实验结果表明,运行该算法后,每轮生成的工作节点在网络中分布良好,所选择的工作节点完全能够满足应用服务期望,而且工作节点数相对较少,同时达到了整个网络工作节点的时间同步,增强了信息的可靠性。图 7 为部署 500 个节点时对应的连通效果,可以清晰地看出,整个工作状态的节点组成的网络是连通的,算法保证了在 $R_c \geq 2R_s$ 时整个网络的连通性。通过在生成树中广播时间同步算法,能够使工作节点网络同步到微秒级,在退避时间 T_d 的允许范围,因而也消除了网络中时间异步对覆盖强度的影响。图 8 为在 $50\text{ m} \times 50\text{ m}$ 监测区域下与 OGDC、PEAS、GAF、DELIC 几种算法所选择的工作节点数的比较。图中工作节点数为每种算法连续仿真 100 次后取平均值所得。

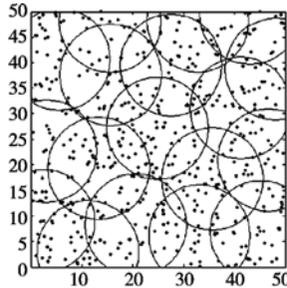


图5 部署500个节点时工作节点的分布效果

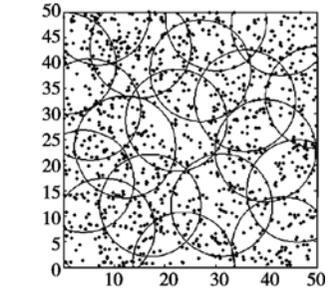


图6 部署1000个节点时工作节点的分布效果

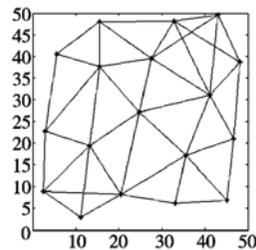


图7 部署500个节点时对应的连通效果

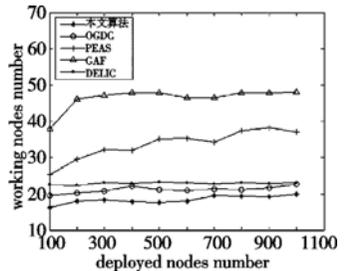


图8 部署节点数与工作节点数量的关系

由于本文首先选择覆盖冗余较少的节点作为整个网络的支撑节点,然后又引入节点冗余度数学模型选择节点,使整个网络满足覆盖要求,所以工作节点相对较少。从图中也可以看出,在大多数情况下 DELIC 需要 23 个工作节点才能达到 99% 以上的覆盖率,对于 GAF 协议,需要的工作节点数量几乎是该算法的 2 倍多;PEAS 可以提供与该算法差不多的覆盖质量,但是选出的工作节点数量比该算法多 50% 左右;OGDC 节点数最少,覆盖率相对较低,而且 OGDC 需要利用节点的位置信息来控制工作节点密度。本文算法无须预先得知节点的位置信息,选择的工作节点数目也相对较少,而且随着部署节点的增加,工作节点数基本不变。

了三种模型。运用模糊模拟与遗传算法相结合,设计了一种混合智能算法求解所提出的三个模型。最后,通过一个数值实验验证了混合智能算法的有效性。

表 4 不同参数设定下的求得的 α -权值

种群规模	模糊模拟次数	P_c	P_m	b	p	α -权值	相对误差 /%
100	3 000	0.5	0.1	6	0.8	263.823	0
100	3 000	0.7	0.2	6	0.8	263.927	0.039
100	3 000	0.5	0.1	10	0.5	264.111	0.109
100	3 000	0.8	0.1	12	0.5	264.084	0.099
100	5 000	0.8	0.3	12	0.8	264.678	0.324
100	5 000	0.7	0.2	10	0.3	264.794	0.368
100	5 000	0.5	0.4	6	0.5	264.691	0.329
50	3 000	0.8	0.4	12	0.3	264.221	0.151
50	3 000	0.7	0.3	8	0.8	264.244	0.160

参考文献:

[1] BAR-YEHUDA R, EVEN S. A local-ratio theorem for approximating the weighted vertex cover problem[J]. *Annals of Discrete Mathematics*, 1985, 109(25): 27-45.

[2] CHVATAL V. A greedy-heuristic for the set cover problem[J]. *Mathematics of Operations Research*, 1979, 4(3): 233-235.

[3] DINUR I, SAFRA S. The importance of being biased[C]//Proc of the 34th Annual ACM Symposium on Theory of Computing. New York: ACM Press, 2002: 33-42.

[4] MONIEN B, SPECKENMEYER E. Ramsey numbers and an approximation algorithm for the vertex cover problem[J]. *Acta Informatica*, 1985, 22(1): 115-123.

[5] CHLEBIKA M, CHLEBIKOVA J. Crown reductions for the minimum weighted vertex cover problem[J]. *Discrete Applied Mathematics*,

2008, 156(3): 292-312.

[6] KHOTA S, REGEV O. Vertex cover might be hard to approximate to within 2-epsilon[J]. *Journal of Computer and System Sciences*, 2008, 74(3): 335-349.

[7] KARAKOSTAS G. A better approximation ratio for the vertex cover problem[J]. *ACM Trans on Algorithms*, 2009, 5(4): 1-8.

[8] NI Yao-dong. Stochastic minimum weight vertex cover problem[C]//Proc of the 5th International Conference on Information and Management Sciences. 2006: 358-364.

[9] ZADEH L A. Fuzzy sets[J]. *Information and Control*, 1965, 8(3): 338-353.

[10] KAUFMANN A. Introduction to the theory of fuzzy subsets[M]. New York: Academic Press, 1975.

[11] NAHMIAS S. Fuzzy variables[J]. *Fuzzy Sets and Systems*, 1978, 1(2): 97-110.

[12] LIU Bao-ding. Uncertainty theory: an introduction to its axiomatic foundations[M]. Berlin: Springer-Verlag, 2004.

[13] LIU Bao-ding, LIU Y K. Expected value of fuzzy variable and fuzzy expected value models[J]. *IEEE Trans on Fuzzy Systems*, 2002, 10(4): 445-450.

[14] LIU Bao-ding, IWAMURA K. Chance constrained programming with fuzzy parameters[J]. *Fuzzy Sets and Systems*, 1998, 94(2): 227-237.

[15] HOLLAND J. Adaptation in natural and artificial system[M]. Ann Arbor, MI: University of Michigan Press, 1975.

[16] KARP R M. Reducibility among combinatorial problems[C]//Proc of Complexity of Computer Computation. New York: Plenum Press, 1972:85-103.

(上接第 37 页)

4 结束语

本文立足于 WSN 的覆盖控制问题提出了一个分布的、高效节能、与节点位置无关、基于时间同步的传感器网络覆盖算法。在本文算法的工作节点选取机制中,节点通过邻居节点的工作状态决定自身工作状态,工作节点网络可达到时间同步,根据能量的大小来选择工作节点。模拟实验结果表明,在无须节点位置信息的前提下,该算法能够大大地减少网络中的工作节点数,从而在减少整体能量消耗、均衡节点能耗方面具有良好的性能。

参考文献:

[1] 石为人,袁久银,雷璐宁. 无线传感器网络覆盖控制算法研究[J]. *自动化学报*, 2009, 35(5): 540-545.

[2] REN Yan, ZHANG Si-dong, ZHANG Hong-ke. Theories and algorithms of coverage control for wireless sensor networks[J]. *Journal of Software*, 2006, 17(3): 422-433.

[3] WU Yong-an, LI Min, CAI Zhi-ping, et al. A distributed algorithm to approximate node-weighted minimum a-connected (θ, k) -coverage in dense sensor networks[C]//Proc of International Frontiers of Algorithmic Workshop. 2008: 221-232.

[4] YE Fan, ZHONG G, LU Song-wu, et al. PEAS: a robust energy conserving protocol for long-lived sensor networks[C]//Proc of the 23rd

International Conference on Distributed Computing Systems. 2003: 28-37.

[5] ZHANG Hong-hai, HOU J C. Maintaining scheme coverage and connectivity in large sensor networks[C]//Proc of NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless, and Peer-to-Peer Networks. 2004.

[6] XU Ya, HEIDEMANN J, EST RIN D. Geography-informed energy conservation for Ad hoc routing[C]// Proc of the 7th Annual International Conference on Mobile Computing and Networking. 2001: 70-84.

[7] 毛莺池, 刘明, 陈力军, 等. DELIC: 一种高效节能的与节点位置无关的传感器网络覆盖协议[J]. *计算机研究与发展*, 2006, 43(2): 187-195.

[8] LIU Chong, WU Kui, KING V. Randomized coverage-preserving scheduling schemes for wireless sensor networks[C]//Proc of IFIP Networking. 2005: 956-967.

[9] HEINZELMAN W B. An application-specific protocol architecture for wireless sensor networks[D]. Boston: Massachusetts Institute of Technology, 2000.

[10] HUANG Chi-fu, TSENG Y G. The coverage problem in a wireless sensor networks[C]//Proc of the 2nd ACM International Conference on Wireless Sensor Networks and Applications. New York: ACM Press, 2003.

[11] LI Ming-he. Time synchronization for wireless sensor network[C]//Proc of the 10th ACIS International Conference on Software Engineering. 2009: 438-442.