國立交通大學

資訊工程學系

碩士論文

在感測網路中分散式保證覆蓋及電源節約協定

Distributed Coverage-Preserving and Power-Saving Protocols for Wireless Sensor Networks

研 究 生：林耿仲

指導教授：曾煜棋　教授

中華民國九十三年六月

在感測網路中分散式保證覆蓋及電源節約協定
Distributed Coverage-Preserving and Power-Saving Protocols for Wireless Sensor Networks

研 究 生：林耿仲　　Student：Keng-Chung Lin

指導教授：曾煜棋　　Advisor：Yu-Chee Tseng

國 立 交 通 大 學
資 訊 工 程 學 系
碩 士 論 文

A Thesis

Submitted to Department of Computer and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer and Information Engineering

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

在 感 測 網 路 中 分 散 式 保 證 覆 蓋 及 電 源 節 約 協 定

學生：林耿仲　　　　　　　　　　　　　　　　　　　　　指導教授:曾煜棋 老師

國立交通大學資訊工程學系（研究所）碩士班

摘　　　要

無線感測網路是一個新興的技術，藉由提供無所不在的偵測、計算與通訊的能力可以大大地便利我們的生活。然而，在設計許多實際的應用上要維持足夠的覆蓋並且延長系統的使用期是兩個互相矛盾的因素。在這篇論文中，我們提出一個分散式的機制可以讓多餘的感測器進入休眠模式以節省電源，而同時維持整個感測的區域在 k 層覆蓋上（k 是一個給定的值）。與 [7] 這篇作者提出的方法做比較，我們提出的方法有較低的複雜度。

關鍵字：感測網路，無線通訊，電源節省，覆蓋。

Distributed Coverage-Preserving and Power-Saving Protocols for Wireless Sensor Networks

Student: Keng-Chung Lin   Advisors: Prof. Yu-Chee Tseng

Institute of Computer Science and Information Engineering
National Chiao-Tung University

ABSTRACT

The wireless sensor network is an emerging technology that may greatly facilitate our life by providing ubiquitous sensing, computing, and communication capability. However, to maintain sufficient coverage and achieve long system lifetime are two contradicting factors in the design of many practical applications. In this paper, we propose a distributed scheme that allows us to put redundant sensor nodes to sleep mode to save energy while maintain the sensing field sufficiently k-level covered, where k is a given parameter. Compared to an existing scheme [7], the proposed scheme has a smaller computational complexity.

Keywords: sensor network, wireless communication, energy conservation, coverage.

Acknowledgments

My advisor, prof. Yu-Chee Tseng, is the first one I would like to express my gratitude to. With the wonderful research conditions he provided and his attentive instructions, I came to discover the pleasure of research. I am also grateful to Chi-Fu Huang. Without his help and suggestions, I would not be able to have this thesis done. Finally, I would like to thank all HSCC members for their generous advices. Discussing with them benefited me in many ways.

Keng-Chung Lin at CSIE, NCTU.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Against the backdrop of the war on terrorism, battlefield application, ecological detection, object tracking, intruder alarming, etc, more and more researches are aimed at on sensor networks. In such environment, to get the accurate measurement, it is unlikely to use the traditional wireless network devices owing to the high cost of large scale deployment. In order to replace the traditional wireless network devices with low-cost sensors, the deployment of sensors needs to have a highly accurate (with different application needs different coverage degrees) yet maintains the original capability - sensing, computing, and networking, i.e., it can detect or monitor the event, have the ability to do the localized computation, and transmit the report back. However, low cost also means less computing ability and limited resources. How to achieve the expected sensing job with low-cost sensors is the main challenge in the design of sensor networks.

Beside the cost, there are some other proprieties of sensors : Sensors are usually tiny, they are powered by battery, we usually deployed large numbers at a time, and after the deployment, they are unattended. Because sensors have these proprieties, there are some constraints go with them. 1. Energy constraint : Because sensors are powered by battery, their lifetime cannot hold very long. 2. Environment constraints : The sensors are usually deployed in battlefield, or some adverse circumstances, such as the neighborhood of volcano. It is unrealistic for sensors to be recharged again. 3. Scale constraints : Owing to large number of deployment, it is hard for administrators to individually and centrally control again. We, therefore, expect sensors to operate as long as possible. In this requirement, power-saving is becoming an important issue.

Although the power saving issue is not ignorable, coverage constraint cannot be neglected, either. Different applications require different level of coverage, so the sensor network need to be maintained at a desired coverage degree. Besides, owing to higher degrees of coverage can also tolerate higher rates of node failures, when the fault-tolerant purpose is mainly concerned. Many applications, such as triangulation-based positioning protocols [3][4], require at least three sensors at any moment to monitor a moving object. Moreover, some applications require even higher degrees of sensing coverage, such as distributed tracking and classification [2]. By these coverage requirement, we need to deal with the power-saving issue with the coverage-preserving issue together. For example, if we want to turn off some redundant nodes to reach longer system life time, we must avoid turning off these redundant nodes will incur some insufficiently covered area (i.e., affects the original sensing coverage). So, we need a safely procedure to turn off these nodes for the desired coverage to be completely guaranteed. Furthermore, nodes may be damaged unexpectedly while new nodes may also be spread again without notification in advance. In both cases, the network should have the ability to adapt itself to the new topology. Considering the above, in this paper we focus the problems on power and coverage issue of sensor networks. In other words, if turning off some nodes still provide the same *coverage* (i.e., the desired coverage is not affected), then it is not necessary for all nodes to be on duty simultaneously. We propose a coverage-preserving and power-saving protocol which can achieve energy conservation without loss of desired coverage.

Achieving longer network life time by scheduling sensor nodes to enter sleeping state has been studied widely by many papers. Most of them cannot satisfy the requirement of *k-level* covered without complicated computation complexity, or guarantee the turning off procedure without causing the loss of coverage, where *k-level* covered means for any position of a given sensor network, it is at least covered by k sensor nodes. For example, [7] requires at least O(n^3) computational complexity to decide degrees of coverage, and the rules proposed in [6][7], which schedule nodes whether to wake up or sleep can't completely guarantee the original sensing area is totally covered. According to the above is our motivation to propose a distributed algorithm for each node to make whether to off-duty decision itself with lower computational complexity, where the desired coverage is still maintained.

The fundamental concept of our algorithm is : for any sensor, if coverage degree of its neighbors will not be affected after its turning off, then the sensor can turn itself off to save the energy, where the desired coverage degree is still guaranteed. Compare to [7] our scheme can compute the level of coverage in low complexity computation. Furthermore, the proposed distributed schedule mechanism can adapt to topological changes. The organization of this paper is as follows. Chapter 2 describes the problem statement. Our proposed coverage-preserving and power-saving mechanism is presented in Chapter 3. The simulation results are given in Chapter 4. Chapter 5 draws our conclusions.

Chapter 2

Problem Statement

We are given a set of sensors, $\prod = \{ n_1, n_2, \ldots, n_n \}$, in a two-dimensional area A. Each sensor $n_i, i = 1..n$, is located at coordinate (x_i, y_i) inside A and has a sensing range of r_i, i.e., it can monitor any point that is within a distance of r_i from n_i.

Definition 1 A location in A is said to be *covered* by n_i if it is within n_i's sensing range. A location in A is said to be *k-covered* if it is within at least k sensors' sensing ranges.

We consider the coverage-preserving and power-saving problem as follow.

Definition 2 Given a natural number K, the *$\underline{K}$-$\underline{C}$overage-preserving and $\underline{P}$ower-saving (K-CP) Problem* is to schedule sensor nodes' on-duty time to prolong network lifetime while maintaining all points in A always K-covered. Our proposed protocol to solve this problem, for short, is call K-CP protocol.

In the beginning, we denote K as the desired level of coverage which is given in advance, and k as the current coverage degree provided by sensors. When considering our goal - to achieve the desired level of coverage, K, how to obtain the current k is the most difficult part of this problem. We exploit the method proposed in [1] to get the current k. Let each node determine its perimeter coverage by placing the intersectional points(left or right point) on the line segment $[0, 2\pi]$ and sort all these points in an ascending order into a list L. Whenever an left point is traversed, the level of perimeter-coverage should be increased by one. Whenever an right point is traversed, the level of perimeter coverage should be decreased by one. By visiting each point in this sorted list form the left to right, we can determine k.

After determining the k, we can compare it with K to know whether the coverage degree is too much. If k is bigger than K, it means there are too many on-duty sensors. Beside to turn off the redundant nodes, we need to deal with k-level covered, and any change of environmental conditions. Therefore, the K-CP problem can be formulated as follows. *For each node :* First, what rule should a sensor take to determine whether to turn off? Second, if a sensor is eligible, how to turn off? Third, after a sensor's turning off, when to turn on?

Chapter 3

The Coverage-Preserving and Power-Saving Protocol

In this section, we introduce our proposed protocol - the coverage-preserving and power-saving protocol. To guarantee the coverage degrees with different application and achieve the energy conservation, each node in the sensor network autonomously and periodically makes turning-off decision itself. To maintain the desired level of coverage, a node can enter the off-duty state once some nodes can take over its sensing responsibility. We divide the algorithm into three processes. In the first part, our study concentrates on determining whether a node is eligible to turn off. Our proposed rule is a self calculating method for each node to make the decision locally. In the second part, we discuss the *blind points* problem and how to prevent it. The *blind points* are insufficiently covered area caused by some nodes' turning off. Because several nodes' deciding to turn off simultaneously, the original coverage degree may below than the desired level. In order to prevent such situation, we propose a procedure for each node to safely turn off, which can guarantee the node's turning off will not affect the desired coverage degree. In the third part, if nodes are going to use up their energy or cannot maintain the coverage degree, we discuss when to wake up the sleeping nodes.

To facilitate illustration, we define the following notation : (Assume each node knows its sensing range r.)

- s_{ij}: For any two nodes, n_i and n_j have overlap of their sensing range, as illustrated Fig. 3.1(a). We denote the bold arc as s_{ij}, which is the segment of node j's perimeter intersected by node i.

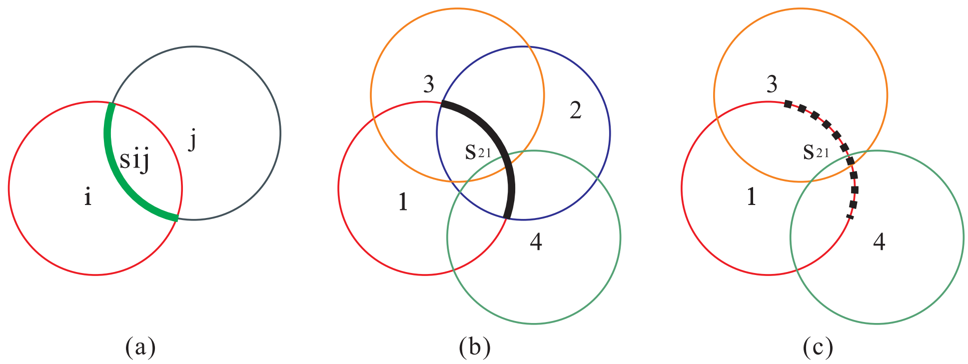


Figure 3.1: (a) s_{ij} the segment of n_j's perimeter intersected by n_i. (depicted as thick line) (b) If coverage degree of s_{21} is greater than K (K is a desired coverage degree, without loss of generality, assuming K = 1 in this case), then this segment covered by n_1 is also covered by some nodes, such as n_3 and n_4. (c) n_2's turning off won't let the coverage degree of s_{21} less than K

- Neighbor: The neighbors of node n_i is defined as $\mathrm{N}(n_i) = \{n_j \mid \mathrm{d}(n_i, n_j) \leq r_i + r_j, n_j \neq n_i, \forall\, n_j \in \prod \}$, where $\mathrm{d}(n_i, n_j)$ is the distance between node n_i and n_j.

3.0.1 Eligible Rule

The goal of our algorithm is to minimize the number of on-duty nodes, where the desired coverage degree is still maintained. To begin with, we review the off-duty eligible rules of some related studies and then compare ours with them.

Probing-Based Eligible Rule

The eligible rule proposed in [9], each node can be in one of the three modes: sleeping, wakeup, or working. A working node is responsible for sensing and data communication in the sensor network field. Other nodes will switch between sleeping and wakeup mode, to prepare for replacing a dying working node due to energy depletion or other hard failures. A sleeping node wakes up after sleeping for an exponentially distributed period of time specified by wakeup rate λ. After a sleeping node wakes up, it broadcasts a probing message PRB within a range with radius γ. When hearing a probing message, any working node within range γ will locally broadcast a reply message PRB_RPY over the wireless channel. If the wakeup node

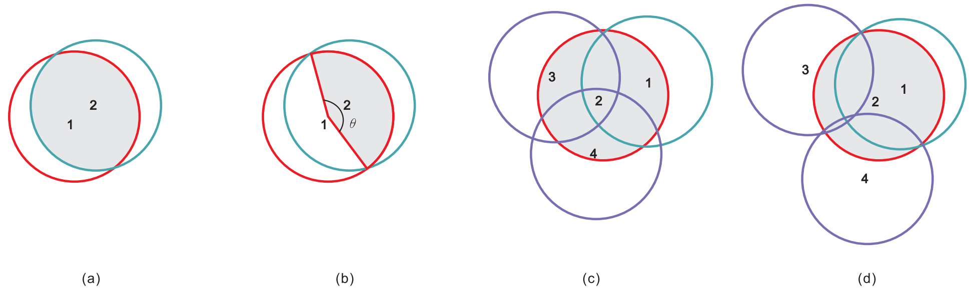


Figure 3.2: Coverage-Based Model.

hears a reply message, it knows that there is a working node within γ. The node will go back to sleeping mode. If the wakeup node does not hear a reply message within time T_w, it assumes that no working node is within its probing range. The node starts working continually. By carefully designing two parameters : the probing range γ and the wakeup rate λ of each node, it can maintain the working node density at a desired value ρ.

But, the probing-based eligible rule proposed in [9] can not guarantee the desired coverage degree of a sensor network. To achieve this objective, intuitively, whether a node can turn off has to make sure its turning off will not cause neighbors' coverage degree less than K. By this consideration, if a node, n_i, desires to turn off should broadcast a packet to all of its neighbors. When a neighboring sensor node, n_j, receives this packet, it calculates coverage degree, k, of n_i with its local information. If k is greater than the desired value, K, n_j acknowledges an approving message matching this packet. If each n_i's neighbors agree to its off duty, n_i can thus be an eligible node, which is an candidate node to turn off. However, there are many drawbacks for this request-and-acks mechanism: 1. Nodes are possibly to send such packet simultaneously. 2. A node who receives this packet needs to check if this neighbor can turn off, then returns the result. The first one involves too much collision, which means if the sender wants to send the request packet or the receiver wants to respond the ack packet needs more packet retransmission. The second one causes lots of unnecessary computational overload. For example, if one of n_i's segments lacks sufficient coverage degree, other nodes are not necessary to compute n_i's eligibility.

Coverage-Based Eligible Rule

To avoid aforementioned packet transmission, the eligible rule proposed in [6], [7],and ours is a coverage-based eligible rule, where each node calculate if itself can be off-duty and can prevent the blind points occurred. [6], [7] are briefly illustrated as follows : In [6], if a node want to turn off, it needs to check whether its is covered by neighbors. But, as illustrated in Fig. 3.2(a), although node 1 and 2 are overlap in the crescent-shape area, it is hard to calculate the crescent than the sector area. Therefore, for a node to decide the eligible rule is to check if the sum all sectors, which are covered by neighbors, can cover its total area. For example, as illustrated in Fig. 3.2(b), if node 1 is turned off, although node 2 can cover a shadow shape within node 1's sensing area, node 1 can only let the node 2 help it monitor a sector-shaped region. The protocol needs neighbors' sensing area can embrace its center, then check whether the union of sectors contains the current node's sensing area, which in turn, is equivalent to calculate whether the union of central angles can cover the whole 360 degree as illustrated in Fig. 3.2(c). According to the constraint of [6]'s eligible rule, although node 2 has been totally covered by its neighbors, yet node 2 cannot turn off. The constraint, which needs a node's center to be covered by neighbors, will cause some unnecessary on-duty nodes.

In [7], the authors transform the problem of determining the coverage degree of a region to the simpler problem of determining the coverage degrees of all the *intersection points* in the same region. The *intersection point* is defined as : (a) A point p ∈ A is called an intersection point between nodes u and v, i.e., p ∈ u ∧ v, if p is an intersection point of the sensing circles of u and v. (b) A point p on the boundary of the coverage region A is called an intersection point between node v and A, i.e., p ∈ u ∧ A. A sensor is eligible for turning off if all intersection points inside its sensing circle are at least K-covered. Considering the complexity of this algorithm, a node needs O(d^2) to know how many intersection points in its sensing range, where d is the number of nodes in a node's sensing neighbor set. Then, for each intersection point, a node needs O(d) to calculate its coverage degree. Therefore, the complexity of eligible rule, which is proposed in [7], is O(d^3).

Nevertheless, unlike [6], where each node check whether itself is fully covered by others, we check whether the neighbor's coverage is sufficient, i.e., if a node's turning off will not affect the neighbors' coverage. And unlike [7], we check the coverage degree of segments inside each node instead of intersection points. Before addressing the details of our eligible rule in algorithm 1, we describe why our eligible rule works.

Observation1: Given a set of nodes, $\prod$, for any node, n_k, (n_i, n_j, $n_k \in \prod$, $n_k \neq n_i$, n_j), if it can cover any segment of s_{ij}, we say n_k must be both n_i and n_j's neighbor. (the set of n_k is denoted by N(n_i) $\cap$ N(n_j), i.e., n_i and n_j's common neighbors)

Apparently, n_k can cover any segment of s_{ij} only when it has some overlap on n_i and n_j. Form the definition of Neighbor, it means n_k must be n_i and n_j's neighbor.

Observation2: Suppose that no two sensors are located in the same location. The whole network area A is k-covered iff each sensor's perimeter in the network is k-covered.

This lemma has been proved in [1].

Based on **Observation1**, if the k is sufficient ($\geq$ K), it means that there exist some neighbors (N(n_i) $\cap$ N(n_j)) can cover n_i's s_{ij}. The eligible rule for n_i can be regulated by calculating whether all s_{ij} of n_i is k-covered. In other words, to calculate the coverage of s_{ij} needs to find n_i and n_j's common neighbors first. **Observation2** guarantees if the any segment of n_i's neighbors, s_{ij}, whose coverage degree inside n_i's sensing range is greater than K, then removing n_i still maintains the desired coverage degree in current sensing range.

For example, if a node, n_i, wants to decide whether to turn off, we transform the problem form calculating coverage degree of n_i itself into checking whether the coverage degree (inside its sensing range) of each n_i's neighbor is sufficient. Take node 2 (Fig. 3.1(b)) as an example, if the arc s_{21}'s coverage degree is grater than K, it means that there exist some nodes let s_{21} be covered at least K level of coverage. In Fig. 3.1(c), without node 2's coverage, s_{21}'s coverage degree is still sufficient here. The rest may be deduced by analogy. If each s_{ij}'s k is greater than K, it means each perimeter of n_i's neighbor, which is inside n_i, can be covered by another neighbors, then n_i's turning off will not affect the coverage degree of n_i's original sensing region. The *Eligible Rule* algorithm is shown below.

for each node in this sensor network, do the following procedure :

Algorithm 1:

for each s_{ij} (j = 1, ..., N(n_i)) of n_i ($n_i \in \prod$)

1. find all arcs of node j covered by (N(n_i) $\cap$ N(n_j))

2. sort the left and right intersections of all these arcs in an ascending order

3. traverse the range of s_{ij}, then determine the k of s_{ij}

4. if (k $\leq$ K) terminate this procedure, else s_{ij} is eligible.

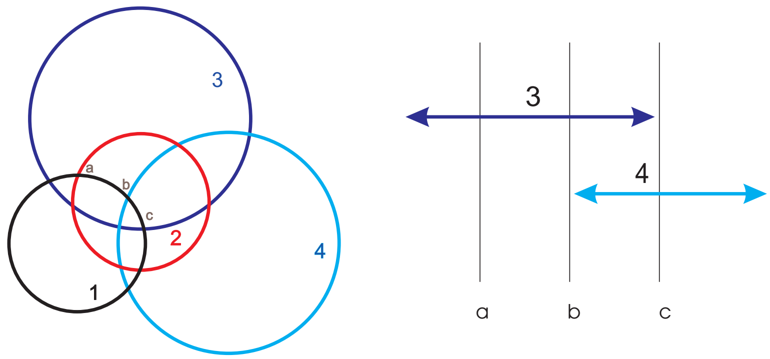
   If each s_{ij}'s coverage degree is greater than K, n_i is eligible to be off-duty; otherwise, n_i should keep on-duty.

It is very easily observed we can achieve lower number of on-duty nodes than [6] after performing the turning off eligible rule and also incur no blind points. Take Fig. 3.2(d) as an example, n_3 and n_4 can also help cover the sensing area of n_2, but unfortunately, they don't embrace n_2's center, so n_2 cannot take advantage of these nodes, but our protocol can also use these nodes, n_3 and n_4 to help the n_2.

Considering the complexity of this algorithm, let d be the maximum number of sensors that are neighboring to a sensor ($d \leq n$). The complexities of steps 1 and 2 are O(d) and O($d$log$d$), respectively. Since the list sorted by step 2 will divide the line segment into as many as $2d$+1 segments, the complexity of step 3 is O(d). Because the node should execute these steps for all on-duty neighbors, whose maximum number is d, the overall complexity is O(d^2log$d$). Compare with [7], the computational complexity of its eligibility algorithm is O(d^3), where d is the number of nodes in the sensing neighbor set. Our algorithm can determine the eligible node at lower cost.

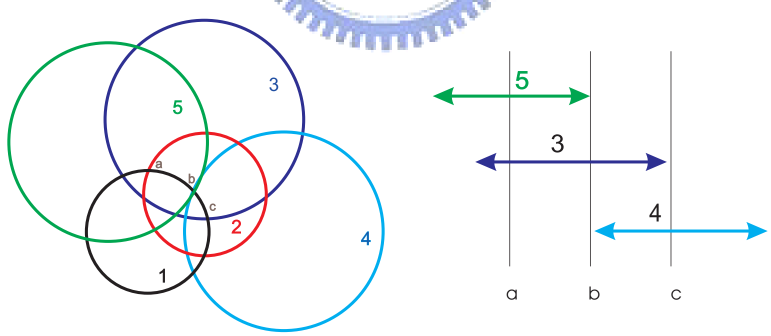
3.0.2 Safe Turning off Procedure

The above discusses which node is eligible to be off-duty. After some nodes are off-duty, for any place in the sensor network, if it can not be detected by any on-duty node, but can be detected by the original on-duty nodes, the place is called as a "*blind point*". The occurrence of blind points means that the corresponding scheduling procedure cannot preserve the original sensing coverage. In the following, we briefly discuss how to prevent the blind point when there are several nodes eligible to turn off.



(a) (b)

Figure 3.3: (a) if n_2 turns off, the coverage degree in the range of a, b can't be decreased again (b) in the range of a, b, check the potential candidate, such as n_3, whether to turn off.



(a) (b)

Figure 3.4: (a) if n_3 also turns off, the coverage degree in the range of a, b is still sufficient. (b) in the range of a, b, although n_3 has higher priority, n_2 need not to ask n_3.

A solution for a node to safely turn off without causing the blind point is to ask each neighbor of the node whether itself can turn off. For example, if a sensor is eligible to enter the sleeping state, it starts a random delay counter. When the counter expires, this node broadcasts an REQ packet to all of its neighbors to ask if it can turn off. After receiving all the confirmation messages(ACK) from its neighbors, this node can enter to the sleeping state. If it does not receive all ACKs in a pre-specified period, it must keep on-duty. If other eligible sensor receives an REQ before the delay counter expiration, it stops the counter, and replies an ACK.

However, asking all neighbors suffer from high cost. The other turning off procedures, such as proposed in [6][7], avoiding collisions by waiting a period of randomized time to check if there is any other node also wants to turn off. This mechanism is not necessary for an elidible node to receive all ACKs, but it can't guarantee to maintain the desired coverage. For example, if an active node, n_i, wants to enter the sleeping state, first of all, n_i checks whether itself is eligible. If so, n_i waits a randomized time to send a wanna-sleep packet and starts a timer to listen. Before the timer expires, if n_i receives the same packet form others, n_i cancels its turning off procedure. Otherwise, n_i enters the sleeping state. However, in lots of situation, if n_i can turn off, the reason is n_i's failing to receive the wanna-sleep packet of others, not because its every neighboring nodes permits its turning off. In this case, several nodes' turning off in the meantime will take the risk of losing packets, and losing packets also means losing some coverage, which may induce the blind points.

To avoid the blind points, we can not rely on waiting a random time to judge if some nodes also want to turn off. However, instead of asking all neighbors, our proposed turning off procedure can reduce some nodes which are not necessary to be asked. First, recall the definition of k-covered, which means : for any given segment, there exists k level of coverage on it. Therefore, if the coverage degree of a segment is K+n, at most n nodes' turning off still make the coverage degree of this segment sufficient ($\geq$ K). Take Fig. 3.3(a) as an example, if n_2 itself decides to turn off, then the coverage degree of a, b segment will become 1, i.e., no other neighbors can turn off anymore. Second, let the lower remaining power node has the higher priority to be off-duty (send the wanna-sleep packet). For any subsegment of s_{ij} in the n_i, we denote $P(s_{ij})$ as the neighbors of n_i and n_j ($\mathrm{N}(n_i) \cap \mathrm{N}(n_j)$), whose power are less or equal than n_i, and each has an overlap with sij. If one of these nodes wants to sleep, it will have higher priority to notify others than node n_i. So coupled with these, if n_i wants to turn off, it need not to ask every node, whose priority is higher than itself. All we need to do is to check if there exist some s_{ij}, $n_j \in \mathrm{N}(n_i)$, whose coverage degree minus $P(s_{ij})$ leaves below or equal K. If so, we call the nodes which cover these segments as *potential candidates*. Then, just check potential candidates whether to turn off. As Fig. 3.3(b) shown, if n_3 has higher priority than n_2, owing to coverage degree of segment a, b can't be decreased again, n_2 need to ask n_3 whether to turn off. However, in Fig. 3.4(a), assuming n_3 is also an eligible node and decides to turn off, although n_3's remaining power is lower than n_2, because n_2 and n_3 both turn off will not make the coverage degree of segment a, b lower than K, n_2 need not to check n_3 whether to turn off. More details are shown in Algorithm 2.

Algorithm 2:

1. for each s_{ij}, check if k of any sub-segment minus the number of $P(s_{ij})$ leaves below or equal K. If so, add the nodes of $P(s_{ij})$ into the potential candidate checking list.
2. alarm neighbors when to sleep by sending a wanna-sleep packet including a list of potential candidates, using its remaining power as the weight to send this packet. (The lower remaining power, the higher priority it can sleep.)
3. any neighbors receiving this packet must check whether itself is in the list. If so, it must send an ack back, and then keep on duty.
4. after sending the wanna-sleep packet, sender needs to wait a period of time to receive these acks. If all neighbors in the checking list send back, we can guarantee to turn off without incurring blind points.

Regarding the goal, the coverage guarantee is the goal of our purpose. As described above, our turning off procedure can entirely preserve the desired coverage without any unexpected packet loss and unnecessary transmission while check much fewer nodes than ask all neighbors.

3.0.3 Turning on Procedure

After describing the eligible rule for a node to determine whether to turn off and a safe procedure for a node to turn off without incurring blind points. There are several issues not yet addressed : 1. when should a node wake up after turning off?

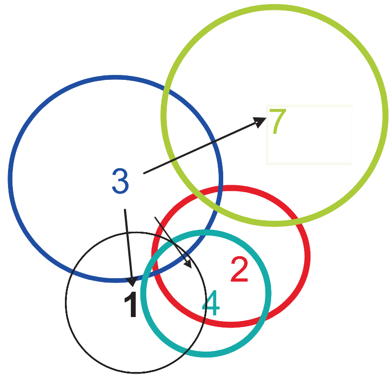
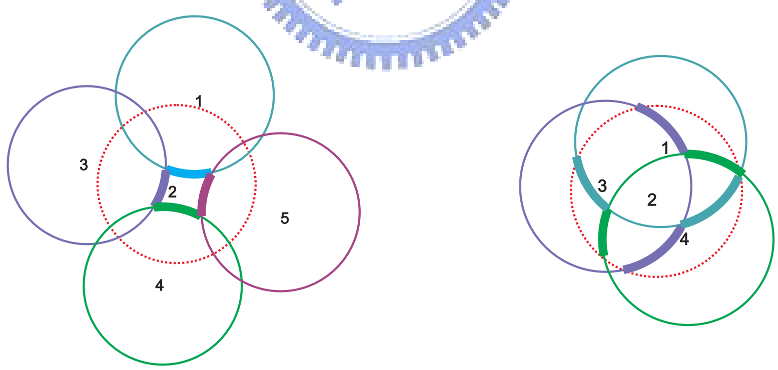


Figure 3.5: The influence of n_3's turning off to node 2.



(a) (b)

Figure 3.6: The bold lines show the insufficient segments when (a) K = 1 (b) K = 2

2. nodes may crash unexpectedly owing to some external factors. 3. new nodes may be spread again without notification in advance. In these situations, nodes should have the ability to adjust themselves to adapt the new environment.

In our proposed K-CP protocol, we set each sensor of the network as synchronized, and the operation is divided into cycles. Each node's cycle is divide into two phase. The first is the scheduling phase and the second is the sensing phase. (As shown in Fig. 3.7, the active window period is the scheduling phase, and the following part of a cycle is the sensing phase.) The scheduling phase has two use : 1. active : advertise its location and 2. passive : listen the advertisement of the others. They are described as follows.

1. Active : If a node is on-duty, it must send a beacon every cycle during the active window. On the contrary, if the node is able to turn off, it must declare the time when it expects to send the beacon. We assume the waking up period is calculated in inverse proportion to residual energy. By sending this beacon, a node let others know it is still alive and according to its waking up period, neighbors listen the beacons to judge whether it is still alive. Take n_2 (Fig. 3.5) for instance, owing to n_3's turning off will affect its level of coverage, n_2 exploits n_3's beacon to record the information including the node's ID, status (active, sleep, or exhausted), coordinates, and the remaining power in a scheduling table.

2. Passive : Every node should wake up periodically to receive beacons. With this step, a node can passively detect the coverage degree by received beacons. (As Fig. 3.7). From these beacons, it is aware that some nodes have not sent the beacon several times during their scheduling period, or the coverage degree is below than K. In the first case, it can delete the entity in its scheduling table by assuming the node is broken or crash. Combining with computing the current k, a node can check whether it needs to wake up some nodes. Once an on-duty node finds its remaining power is not sufficient enough or the coverage degree of some segments is less than K, it checks its recorded table, and then finds some nodes, which can cover its original sensing range if it is going to use up its power or help itself to enhance the coverage degree of some critical segments. We, then, discuss how to find these nodes from the recorded table.

Take Fig. 3.5 as an example, if n_2 is going to use up its power, it recognizes that its turning off will affect some neighbors' coverage, such as n_1, so n_2 checks its table whether it records some nodes whose remaining power is greater than itself and the union of these segment can cover the desired segment. In this case (assuming n_3, and n_4 are in sleep state, and n_1, n_2 are in active state), n_2 may find that n_3, and n_4 can wake up to cover n_2's s_{21}, so it will send a packet at each n_3, and n_4's expecting wakeup time to notify them. As Fig. 3.5 shown, from n_1's viewpoint, its perimeter covered by n_2 originally now can be covered by n_3, and n_4 instead, which maintains the same coverage degree after n_2's turning off.

However, waking up nodes based on individually segment is not a good solution. As above mentioned, for each s_{ij} segment, if a node wakes up some sleeping neighbors to cover one segment, it may cause redundancy for the other segments. Take Fig. 3.5 as an example, for n_2, if there exists one node can cover s_{27}, and also cover s_{21}, obviously n_2 need not wake up n_3 and n_4 to cover s_{21}. Therefore, the less wake up nodes is the better, rather than aiming at each segment individually. According to this reason, if a node needs to wake up some neighbors to cover its original sensing range, it need to take all segments into consideration, i.e., as Fig. 3.6 shown, the node, which we want to wake up, should reduce the most insufficient segments. Our proposed method is as follows :

for each node, who wants to wake up some sleeping neighbors to cover its original sensing range, do the following procedure :

1. do
   (a) for each s_{ij} of n_i ($n_i \in \prod$), where j = 1, ..., N(n_i), and n_j is on-duty node. S = 0; determine the k of s_{ij}
   (b) if (k > K) j++;
   (c) else sum all sub-segment's range of s_{ij} into S, where the sub-segment's k $\leq$ K.
2. for each sleeping state node, which is recorded in n_i's table, determine which one's waking up can cover most of these insufficient sub-segments (reduce S most). Then, wake up the node.
3. Repeat step1 and 2 until there are no insufficient segments.

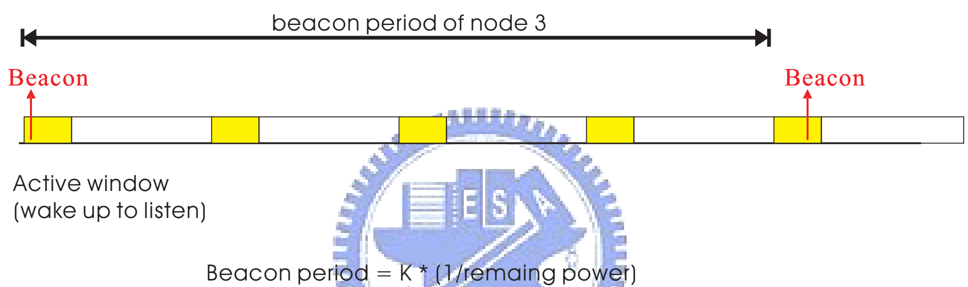


Figure 3.7: After node 3's turning off, the waking up schedule is as above.

Chapter 4

Simulation Results

We set our experiments in the same parameters as [6]. Deploying 100 sensor nodes in a square space (50m by 50m), whose coordinates are set randomly in this place. The sensing range of each node is 10 meters and each node knows neighbors' position. When a node turns off, all its neighbors can know this turning off decision. After all nodes have made decisions, the number of off-duty nodes and the current sensing coverage degree by on-duty nodes are compared with the original one where all nodes are on-duty. To calculate sensing coverage, we also use the same mechanism as [6], which divides the space into 1m by 1m unit cells. Assume an event occurs in each cell, with the event source located at the center of the cell. We investigate how many original nodes and how many on-duty nodes can detect every event, and also compute the average sensing degree before and after turning off nodes.

Table 4.1 shows the experimental results when we apply our protocol, [6] and probing-based [9] off-duty eligibility rules 100 rounds in 100 random topologies, respectively. As the results illustrated, by applying our proposed protocol, nodes can be turned off to eighty, which is better than [6]'s one half of original on-duty nodes. The sensing degree after turning off redundant nodes is reduced from 10 to 2, which also means more redundant nodes can be off-duty. The probing-based off-duty eligibility rule makes almost the same number of nodes be turned off as ours when the probing range is set as 7 meter, but blind points appear in 100 topologies in that case, i.e., blind points occur in every topology. It also shows the proposed coverage-based off-duty eligibility rule, such as [6] and ours can guarantee no blind points occurred.

Eligibility Rules	Probing range	Number of off-duty nodes	Original sensing degree	Obtained Sensing Degree	Number of topologies with blind points	Average number of blind points per topology
Proposed	N/A	79.95	10.30	2.03	0	N/A
[6]	N/A	50.57	10.32	5.62	0	N/A
[9]	3	38.05	10.30	6.32	13	0.63
[9]	4	54.41	10.30	4.6	43	4
[9]	5	66.79	10.33	3.32	62	9.29
[9]	6	75.16	10.30	2.46	95	34.41
[9]	7	80.75	10.30	1.89	100	104.06

Table 4.1: Comparison of three off-duty eligibility rules.

On-duty node number vs. Density

Another result of Table 4.1 shows that the larger probing range of [9] results in more nodes being turned off and more sensing coverage being reduced when the probing-based off-duty eligible rule is used. To investigate the relation between density and on-duty node number when the coverage-based off-duty eligible rule is used, we change node density by varying the node number and sensing range. Although increasing node number and sensing range will increase the on-duty nodes, the proposed mechanism as illustrated in Fig. 4.1 also works well with coverage-based eligible rule. As shown in the figure, when the number of nodes increase to three times, the on-duty nodes proposed in [6] increase about 30% while ours K-CP protocol only increase about 10%. When the nodes are nearby the boundary, in this experiment, according to both [6] and our proposed K-CP eligible rule, nodes have no chance to turn off owing to all the other nodes cannot help to cover their sensing range, which is outside the boundary. However, our eligible rule still gets better results in the boundary situation.

Coverage degree vs. Density

One characteristic of our proposed protocol is that we can decide "coverage degree" with different application. In this experiment, we, firstly, investigate the change of sensing degree over node density. As shown in Fig. 4.2, the experiment show

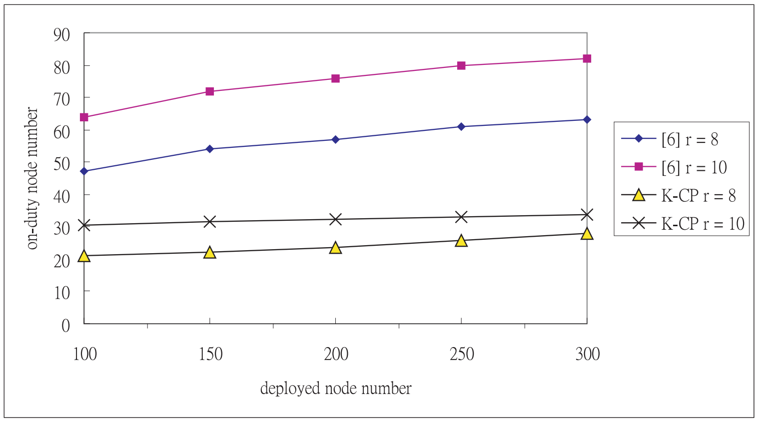


Figure 4.1: on-duty number vs. deployed number. *dtian means the protocol proposed in [6]*

that increasing the number of nodes will also leads to level of coverage. Let each node calculates its coverage degree of perimeter, and the total sum of each node's coverage degree divided by total nodes is the *average* coverage degree of this network. Compare the average coverage degree, we define the *actual* coverage as : after each node calculates its coverage degree, the lowest coverage degree of node in this sensor network is actual coverage degree. We then define obtained coverage degree as : after turning off some redundant nodes, the actual coverage degree is obtained coverage degree. Interestingly, although the average coverage degree increases almost three times as the increase of deployed node number, the actual network coverage doesn't have the same proportion. It means the inequality deployment of the sensors cause unbalanced level of coverage. Because our goal is to maintain the coverage degree of a given system above K, the gap between actual k and the obtained k is the space, where we can use some mechanism to prolong the surviving time.

Coverage degree vs. Surviving time

To investigate the survival time with the different coverage degree, we set the original power of each node ranges form 200 to 1000 unit, and deploy the sensor nodes in a square space (50m by 50m). To simplify, we assume each scheduling cycle consuming one unit power of on-duty node, and there is no other energy consumption. We define survival time as : in a given sensor network, how long can this network maintain the K before any point's coverage degree is below than K. Fig. 4.3 shows the survival time with different K. Compare with the Fig. 4.2, the survival time doesn't increase as the same ratio as the gap between actual k and the obtained k. Because there are some area, even after the nodes performing the eligible rule , only a few nodes can be turned off in order to maintain K (i.e., the k in that area may greater than K). These nodes are called critical nodes, which means although we can reduce original network's level of coverage degree to K, once the critical nodes use up their power, they can't find any neighbors to wake up to cover their sensing area. It also means there is no disjoint relationship between level of coverage, which will be another consideration for deploying the sensors. Fig. 4.4 shows the correlation between off-duty number and survival time. It also demonstrates that more off-duty nodes indeed increase more survival time. As we can tell from the increase of the slope, the more deployed node numbers can the system prolong more system survival time of K-CP than [6], because it increases the probability of disjoint situation of coverage degree.

The other characteristic of our proposed protocol is that even the network can't maintain K, it still provides more stable coverage degree. If the sensor network calculates its average coverage degree after each cycle, as Fig. 4.5 illustrated, the slope whether perform the algorithm shows great variations in intensity, though both coverage degree can not remain above K. When we deploy 200 sensor nodes in a square space (50m by 50m), the vertical line means the time when the system can not maintain the K coverage degree, and we can see the slope of average coverage degree descends much more than the eligible rule algorithm is performed. It means that although some area of the sensor network occur blind points, most area can still be sufficiently covered.

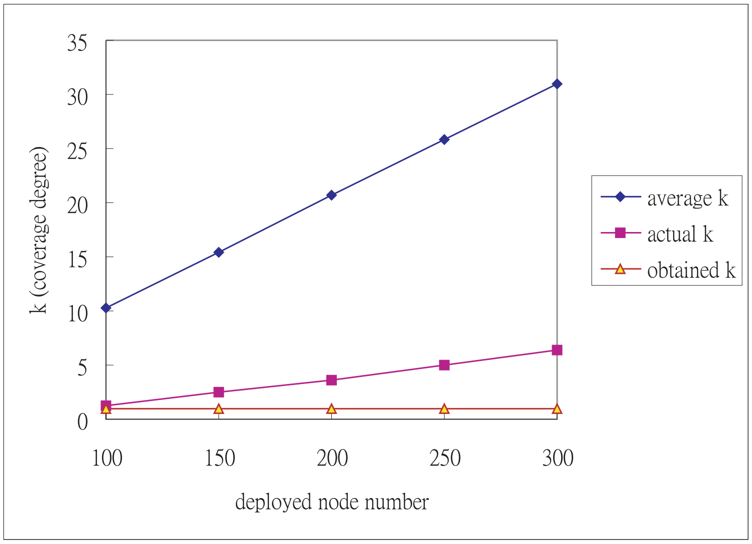


Figure 4.2: deployed number vs. coverage degree

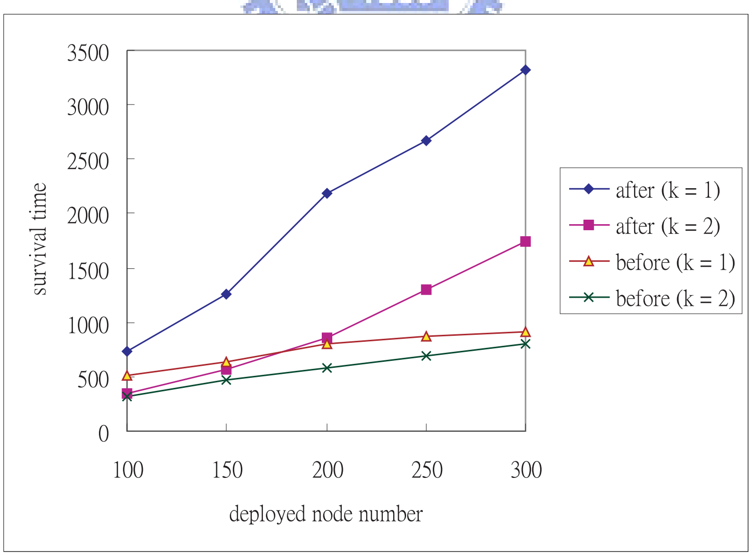


Figure 4.3: deployed number vs. survival time

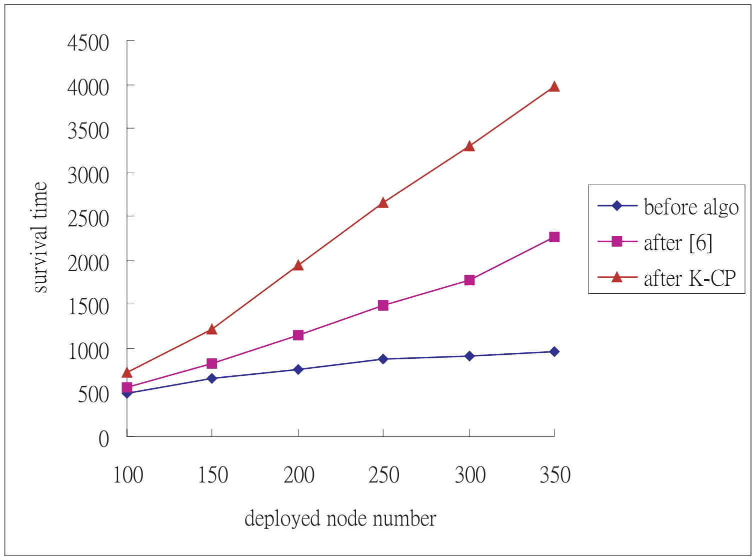


Figure 4.4: [6] and K-CP deployed number vs. survival time

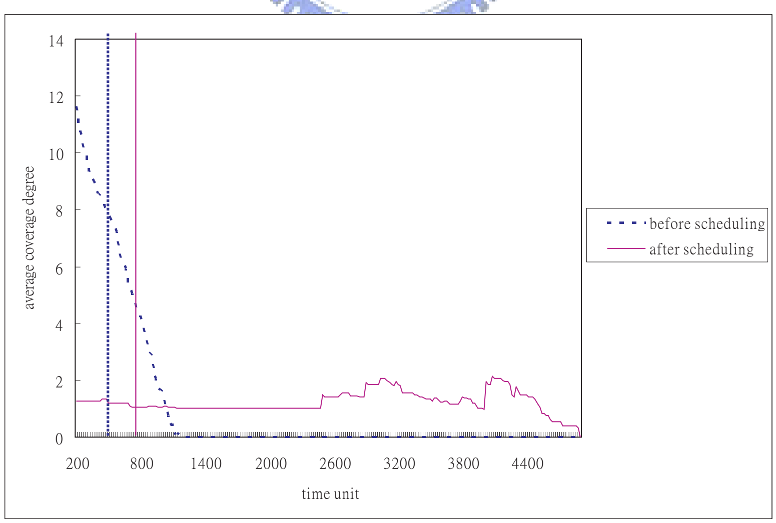


Figure 4.5: average coverage degree vs. time unit

Chapter 5

Conclusions

In this paper, we have proposed a mechanism, namely eligible rule for node's turning off calculation, safely turning off procedure to prevent blind points occurred, and scheduling rules for nodes to be on- or off-duty for power saving, which is based on distributed method to calculate coverage degree of each node. With the proposed techniques, we can guarantee the network is fully covered at least K level of coverage and also let maximum nodes to sleep. Our proposed solution can also work out with the ability to adapt the topological change. Applying the mechanism to asynchronous sensor network is currently our work.

Bibliography

[1] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. In *ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 115–121, 2003.

[2] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed. Detection, classification and tracking of targets in distributed sensor networks. *IEEE Signal Processing Mag.*, 19(2):17–29, Mar. 2002.

[3] D. Nicules and B. Nath. Ad-hoc positioning system (APS) using AoA. In *IEEE INFOCOM*, 2003.

[4] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *ACM Int'l Conf. on Mobile Computing and Networking (MobiCom)*, pages 166–179, 2001.

[5] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. In *IEEE Int'l Conf. on Communications (ICC)*, pages 472–476, 2001.

[6] D. Tian and N. D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *ACM Int'l Workshop on Wireless Sensor Networks and Applications (WSNA)*, 2002.

[7] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *ACM Int'l Conf. on Embedded Networked Sensor Systems (SenSys)*, pages 28–39, 2003.

[8] F. Ye, G. Zhong, S. Lu, and L. Zhang. PEAS: A robust energy conserving protocol for long-lived sensor networks. In *Int'l Conf. on Distributed Computing Systems (ICDCS)*, 2003.

[9] F. Ye, G. Zhong, S. Lu, and L. Zhang. Energy Efficient Robust Sensing Coverage in Large Sensor Networks. In *UCLA Technical Report*, 2002.

[10] R. Min, and A. Chandrakasan Energy-Efficient Communication for Ad-Hoc Wireless Sensor Networks. In *Asilomar Conference on Signals, Systems, and Computers, vol. 1, pp. 139-143.*, Nov. 2001.

Curriculum Vita

Keng-Chung Lin (kclin@csie.nctu.edu.tw) received his B.S. degree in Computer Science from the National Chiao-Tung University, Taiwan, in 2000. His research interests include wireless network, sensor network and computer security.