Wireless Personal Communications (2007) 41:449–464
DOI 10.1007/s11277-006-9152-4

A Distributed Query Protocol in Wireless Sensor Networks

JANG-PING SHEU, SHIN-CHIH TU and CHIA-HAO YU

Department of Computer Science and Information Engineering, National Central University, Jhongli, 32054 Taiwan, R.O.C.
E-mail: sheujp@csie.ncu.edu.tw

Abstract. In wireless sensor networks, query execution over a specific geographical region is an essential function for collecting sensed data. However, sensor nodes deployed in sensor networks have limited battery power. Hence, the minimum number of connected sensor nodes that covers the queried region in a sensor network must be determined. This paper proposes an efficient distributed protocol to find a subset of connected sensor nodes to cover the queried region. Each node determines whether to be a *sensing node* to sense the queried region according to its priority. The proposed protocol can efficiently construct a subset of connected *sensing node*s and respond the query request to the sink node. In addition, the proposed protocol is extended to solve the *k*-coverage request. Simulation results show that our protocol is more efficient and has a lower communication overhead than the existing protocol.

Keywords: Wireless sensor networks, coverage problem, query execution

1. Introduction

In wireless sensor networks, unusual events or general phenomena sensed by sensors are typically collected by a sink node through a query execution over a specific geographic region. However, sensor nodes in sensor networks have only limited battery power [1–4]. Allowing all nodes in a region of interest to answer an incoming query is very energy-inefficient and unnecessary. In fact, only a subset of the sensor nodes is required to sense a region during query execution, while other sensor nodes need not deal with the incoming query. Besides, sensor nodes in a specific region must efficiently report an urgent query from a remote sink node. Hence, sensor nodes, which are dynamically deployed in the region, should be able to determine whether to sense a region and confirm the coverage [5] and connectivity in a distributed way.

A greedy method for query execution has been proposed in [6], to find a *connected sensor coverage set*. That work presented a centralized version of an approximation algorithm and a distributed one. An arbitrary node in the centralized algorithm within a queried region is selected at the start of the algorithm. Then, a path of sensor nodes that connects an already selected sensor node to another sensor node (*candidate sensor node*), which is partially covered by already selected sensor nodes, is selected as a candidate path. Among the selected candidate paths in each step, the one that covers the most uncovered area in the queried region (*most beneficial candidate path*) is added to the already selected sensors. Such a process operates continuously until the selected set of sensor nodes cover the queried region completely, and the algorithm terminates. However, such a centralized algorithm is not easily adapted to a large-scale region. The distributed algorithm in [6] is converted from the centralized one.

Each addition of a path of sensor nodes is associated with a large communication overhead. Furthermore, although the greedy method operates in a distributed way, the overall operation is sequential. Therefore, the process requires a long time (that is proportional to the number of selected nodes) to cover the sensed region. Therefore, such an algorithm cannot respond efficiently to the query execution especially for a large queried region.

This work proposes an efficient two-phase distributed protocol. Here, sensor nodes with different sensing ranges and communication ranges are considered. In the first phase, upon receiving a sensing query request, each sensor node in the queried region concurrently determines whether to be a *sensing node,* from its priority value. The *remaining energy*, *sensing range* (sensing area) or *communication degree* (the number of neighbors of a sensor node) can represent the *priority*. If two nodes have the same *value*, then the node with the larger *id* has higher priority. Different settings of the priority value result in the selection of different sets of *sensing nodes*, with particular properties. The settings of priority will be determined by our simulation results.

In the second phase, each *sensing node* is aware of other neighboring *sensing nodes* and connects with each other using the route information of its *1-hop-cover* neighbors. Sensor nodes are *1-hop-cover* neighbors of each other if their sensing areas intersect with each other. *Sensing nodes* are neighboring *sensing nodes* if they are *1-hop-cover* neighbors each other. Consequently, a *connected sensor coverage set* can be efficiently formed using the proposed protocol. The entire operation is executed by all nodes concurrently, rather than sequentially [6], so a quick response time with low overhead, and high scalability are achieved. Moreover, the proposed protocol is extended to solve the *k*-coverage problem, which can find a set of *sensing nodes* satisfy the *k*-coverage request. Simulation results show that the proposed protocol is more time-efficient with a lower communication overhead than the greedy method presented in [6].

The rest of this paper is organized as follows. Section 2 introduces the environmental assumptions and challenges encountered in solving the query execution problem. Section 3 presents the proposed two-phase protocol. Section 4 presents the *k*-coverage sensing node discovery algorithm. Section 5 evaluates the performance of the proposed protocol by simulations. Section 6 concludes the paper.

2. Preliminaries

The sensor network environment considered in this paper is that all the sensor nodes are assumed to lie in a two-dimensional domain. Transmission ranges and sensing ranges differ between sensor nodes. The sensing range of a sensor node may differ from its transmission range. A sensor node can send packets to all nodes within its transmission range; otherwise it sends via intermediate nodes to relay the packets. Each wireless sensor node is static and is aware of its own location either through the Global Positioning System (GPS) or other positioning methods [5, 7, 8]. Additionally, all the sensor nodes in a sensor network are assumed to be connected and sufficient to cover the region of interest, so the arising neighboring *sensing nodes* can connect with each other to form a *connected sensor coverage set*.

In this paper, the problem to be solved is to determine a subset of connected sensor nodes, which covers the queried region [5,9,10] and can answer the query [11]. Note that, the selected sensor nodes to cover the queried region are named *sensing nodes* (working nodes) in this paper. For example, in Figure 1, the *nodes* with small solid circles, which are selected to sense the

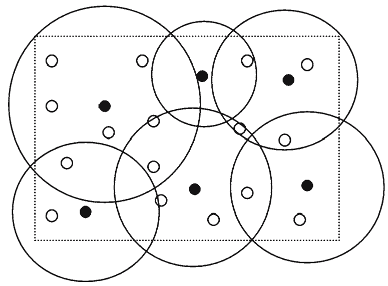


Figure 1. Example of queried region in a sensor network.

queried region, are *sensing nodes*; the others are non-sensing nodes. Here, three important characteristics of the solution to the query execution problem are presented. First, finding the minimum number of connected *sensing nodes* that cover an area is an *NP*-hard problem [6,12], even when a centralized algorithm is used. Second, wireless communication consumes most of the power during the lifetime of a sensor node [13]. Therefore, a large control overhead reduces the lifetime of the sensor network. Last, some emergent queries, such as danger detection, should be replied to in a limited time; otherwise, the replied information would become invalid. Hence a *connected sensor coverage set* should be organized efficiently. The proposed protocol considers these three challenges.

3. Coverage Set Determination Protocol

The proposed protocol consists of two phases – *self-pruning phase* and *sensing nodes discovery phase*. In the self-pruning phase, each sensor node determines whether to be a working node to sense the queried region. In the sensing nodes discovery phase, each *sensing node* determines which of its *1-hop-cover* neighbors are *sensing nodes* and then connects to them. A sensor node j, whose sensing area intersects with sensor node i is called the *1-hop-cover* neighbor of node i. The set formed by the *1-hop-cover* neighbors of sensor node i is defined as $NB(i) = \{j|$ where $SA_i \cap SA_j \neq \phi\}$. In the proposed protocol, an area covered by the sensing area of sensor node i is denoted by SA_i, where SA_i is considered to be a circular disk with radius r.

In the beginning of the presented protocol, each sensor node is assumed to have the information of its *1-hop-cover* neighbors. Each sensor node can collect its *1-hop-cover* neighbors once it is deployed in the sensor network. The *1-hop-cover* neighbors of each node can be collected by exchanging node information of each other. Node information includes a node's *id*, *sensing range*, *location* and *priority*. First, each node broadcasts its node information to neighbors. Then, each of the neighboring nodes will rebroadcast the received node information if its sensing range is intersecting with sensing range of the broadcasting node. Since the sensor nodes have different sensing ranges and communication ranges, a sensor node may require more than one hop to communicate with its *1-hop-cover* neighbors when its sensing range exceeds the communication range.

3.1. Self-pruning Phase

Each sensor node that has sensing area within the queried region executes the self-pruning phase when it receives a query from a sink node. Let $NPri(i) = \{j|pri(j) > pri(i)$ and

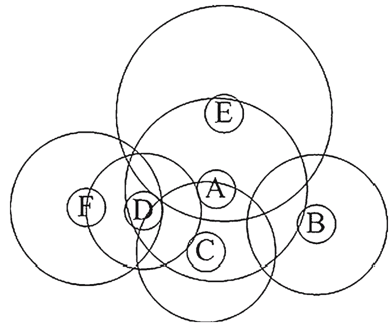


Figure 2. Example of rule 1.

$j \in NB(i)\}$ be a subset of $NB(i)$, whose priority is higher than that of node i, where $pri(k)$ denotes the priority of node k. Let $SA(NPri(i))$ be the sensing area covered by the nodes in $NsPri(i)$. In this phase, each node checks whether it will become a *sensing node* by applying the following rule.

Rule 1. A sensor node i becomes a *sensing node* if the sensing area SA_i is not completely covered by $SA(NPri(i))$.

Figure 2 shows an example of rule 1. In the following, the priority value of each node is related to the sensing radius of each node. A larger sensing radius represents a larger priority value. Although SA_A is completely covered by the union of SA_B, SA_C, SA_D, SA_E, and SA_F, node A is still a *sensing node* because its sensing area is not fully covered by $SA(NPri(A))$. Node D cannot be a *sensing node,* according to rule 1, since SA_D is completely covered by $SA(NPri(D))$.

Theorem 1. Suppose the deployed sensor nodes are sufficient to cover the queried region. The *sensing nodes* selected by *rule 1* can fully cover the queried region.

Proof: For a sensor node S, assume that SA_S is fully covered by $SA(NPri(S))$. Let $NPri(S) = \{S1, S2, \ldots, Sn\}$ and R_i be the intersection area of SA_s and SA_{si}. Accordingly, $SA_S = R_1 \cup R_2 \cup \ldots \cup R_n$. By rule 1, node S is not a *sensing node*. Assume there exists a node Sk that is not a *sensing node*, where $1 \leq k \leq n$. According to rule 1, SA_{sk} must be fully covered by $SA(NPri(Sk))$. Assume $NPri(Sk)$ is the union of SA_{h1}, SA_{h2}, …, and SA_{hm}. Thus, the sub-area R_k intersected by SA_s and SA_{sk} is still covered by SA_{h1}, $SA_{h2}, \ldots$, and SA_{hm} completely. Consequently, the queried region will be fully covered by the selected *sensing nodes* according to rule 1. □

An example in Figure 3 illustrates the results of the self-pruning phase. Here, gray nodes represent the connected *sensing nodes* and white nodes represent *non-sensing nodes*. The number inside each circle represents the *id* of each sensor node. The sensing range of each *sensing node* is stressed and denoted by circular disk in a bold line. Priority value of each sensor node is determined by its sensing radius. The sensor nodes have different communication ranges, so the lines between the nodes indicate that the nodes can directly communicate with each other. In the following, we will present how each node i can detect that its sensing area SA_i is fully covered by $SA(NPri(i))$ or not. If k sensor nodes cover all points in an area (or a sub-perimeter), the area (or the sub-perimeter) is said to be k-covered [5]. Notably, the

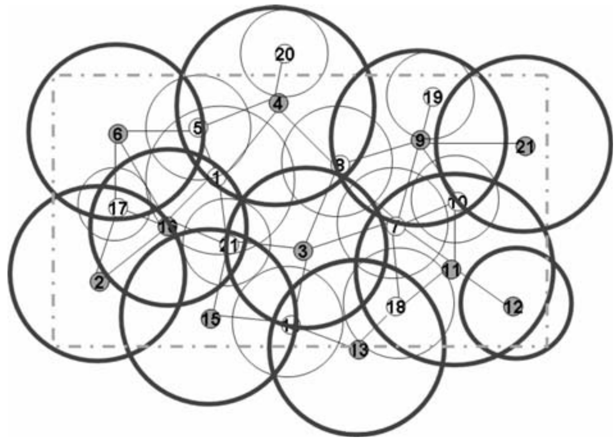


Figure 3. Result of *self-pruning phase*.

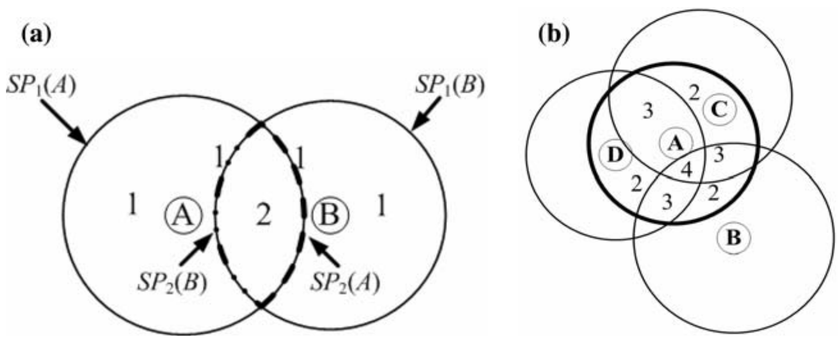


Figure 4. (a) Determining the coverage level of each sub-area. (b) Node A is fully covered by nodes B , C , and D.

perimeter of a node i is not covered by SA_i. Let $P(i)$ be the perimeter of SA_i. Suppose $P(i)$ is the union of n sub-perimeters $SP_j(i)$, $j = 1, 2, \ldots, n$. That is

$$P(i) = \bigcup_{1 \le j \le n} SP_j(i),$$

where $SP_j(i)$ is the jth sub-perimeter of the $P(i)$. If a sub-perimeter is k-covered, then the sub-area outside the sub-perimeter is k-covered and the sub-area inside the sub-perimeter is $(k+1)$-covered. For instance, in Fig. 4(a) since $SP_2(A)$ is 1-covered (covered by node B), sub-area outside $SP_2(A)$ is 1-covered (covered by node B) and sub-area inside $SP_2(A)$ is 2-covered (covered by nodes A and B). Assume that the sensing area SA_i is divided into several sub-areas by the perimeters of nodes in a subset $H \subseteq NB(i)$. If every sub-area in SA_i is at least 1-covered by H, then node i is fully covered by H [5].

Before node i can determine the coverage degree of each sub-area in SA_i, node i has to firstly obtain the coverage degree of each sub-perimeter associated with the sub-areas in SA_i. The sub-perimeters associated with each sub-area in SA_i include the sub-perimeters of node i and the sub-perimeters of nodes in H located in SA_i. The coverage degrees of these sub-perimeters can be obtained by using the locations and the sensing radii of nodes in H . For example, in Figure 4(b) assume nodes B, C , and D belong to $NPri(A)$. Each sub-perimeter of node A is at least 1-covered by the nodes in $NPri(A)$ and the sub-perimeters of nodes in $NPri(A)$ located in

SA_A are at least 2-covered (1-covered by node A and at least 1-covered by other nodes of higher priority than node A). Therefore, $NPri(A)$ fully covers node A. Note that, since the sensing range may exceed the communication range, a node cannot guarantee to collect all of its *1-hop-cover* neighbors. The uncollected node may become a *sensing node* even it is fully covered by its *1-hop-cover* neighbors. Although the insufficient collection of *1-hop-cover* neighbors will increase the number of *sensing node*s, it does not make sensing void in the query region.

3.2. Sensing Nodes Discovery Phase

In the sensor network, phenomena sensed by each *sensing node* have to be further forwarded to the remote sink node. To achieve this goal, each *sensing node* needs to be aware that which sensor nodes among its *1-hop-cover neighboring set* become the *sensing nodes* so as to connect to them. Here, the sensing nodes discovery phase is presented to identify the *sensing nodes* from its set of*1-hop-cover* neighbors. A *connected sensor coverage set* could be formed after the sensing nodes discovery phase is executed. The following describes a *property* that helps to find the *neighboring sensing nodes* of each *sensing node*. If a specific region R is fully covered by a set of *sensing nodes*, then the sensing area that is covered by the *sensing nodes* in R must intersect with each other. Otherwise, a sensing void exists in R unless R can be fully covered by simply one sensor node. Hence, for any *sensing node* i, there must exist at least one *sensing node* in $NB(i)$ if R cannot be fully covered by simply one sensor node i.

Rule 2. Any node i can recognize node j as a *sensing node* if there exists at least one sub-perimeter of $P(i)$ that is covered by node j, where node j has the highest priority value among all of the nodes that cover the same sub-perimeter. Notably, each node j in $NB(i)$ may cover more than one sub-perimeter of $P(i)$.

Figure 5 shows an example of the application for rule 2. Suppose four sensor nodes A, B, C and D have sensing ranges located in a queried region. Consider the sensor node A. Node A is not completely covered by $NPri(A)$, so A is aware of being a *sensing node* after the self-pruning phase. In Fig. 5, the perimeter of $P(A)$ is divided by the perimeters of its *1-hop-cover neighbors* into five sub-perimeters, $SP_1(A)$, $SP_2(A)$, $SP_3(A)$, $SP_4(A)$ and $SP_5(A)$. $SP_1(A)$ and $SP_4(A)$ are only covered by nodes B and D, respectively, so nodes B and D are recognized as *sensing nodes* by node A, according to rule 2. On the other hand, $SP_2(A)$ and $SP_3(A)$ are covered by nodes B and D, respectively, but $SP_2(A)$ and $SP_3(A)$ are also covered by node C, which has a higher priority value than both nodes B and D. Accordingly, node C would be recognized as a *sensing node* by node A too.

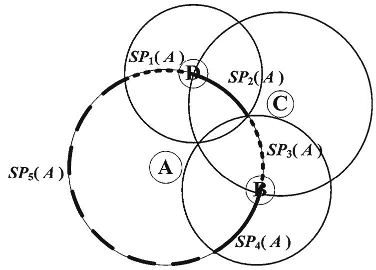


Figure 5. Example of rule 2.

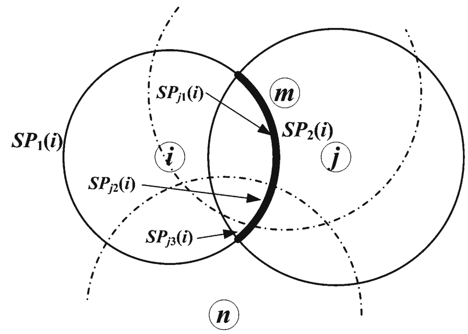


Figure 6. Sensor node i can recognize sensor node j as a *sensing node.*

Theorem 2. Any *sensing node* can find at least one *sensing node* of its neighboring *sensing nodes* according to rule 2.

Proof: Given any two nodes i and j, whose sensing areas lie in the same queried region and $j \in NB(i)$, of which situation an example is depicted in Figure. 6. Assume that perimeter $P(i)$ is divided by $P(j)$ into $SP1(i)$ and $SP2(i)$. $SP2(i)$, which is covered by node j, may be divided by other perimeters of the nodes in *NB*(i) into n sub-perimeters $SPj1(i)$, *SPj*2(i), . . ., and$SPjn(i)$. For example, $SP2(i)$ the thick curve in Figure 6 is divided into three sub-perimeters $SPj1(i)$, $SPj2(i)$, and $SPj3(i)$. If node j has the largest priority value in any sub-perimeter $SPjk(i)$, where $1 \leq k \leq n$, according to rule 1, node j must be a *sensing node* to cover the sub-perimeters $SPjk(i)$. Otherwise, there exists at least one node k has the largest priority covering one or more sub-perimeters $SPjk(i)$, where $1 \leq k \leq n$, and node i will recognize node k is a *sensing node*. □

Although a *sensing node* can recognize its neighboring *sensing nodes* by rule 2, the recognition is uncertain in some special cases. To illustrate this case, given two sensor nodes i and j. Node j is an uncertain case to node i if node i does not recognize j as a *sensing node* according to rule 2 but j is indeed a *sensing node* by rule 1. This happens when the sub-perimeters in $P(i)$ covered by node j are also covered by a subset of nodes $H \subset NB(i)$ and the nodes in H have higher priority values than node j. Therefore, node i would recognize some node(s) in H as *sensing node*(s) rather than node j. However, node j may be determined to be a *sensing node* by rule 1 when *SAj* is not fully covered by its neighboring *sensing nodes*. Such determination will result in the inability of node i to recognize some *sensing nodes* in $NB(i)$. Figure 5 shows an example with uncertainty. Here, sub-perimeters $SP_2(A)$ and $SP_3(A)$ covered by node C are also covered by nodes B and D, respectively. Suppose the priority values of nodes B and D exceed that of node C. Node A will recognize nodes B and D as *sensing nodes* instead of node C. However, our protocol is not affected by the uncertain case because the *sensing node C* will be detected by other *sensing nodes* eventually.

The following describes why the selected *sensing nodes* are connected and can cover the queried region. According to rule 2, each *sensing node i* can recognize a set of *sensing nodes* to fully cover its perimeter. Similarly, each *sensing node* recognized by node i can also find a set of *sensing nodes* that covers its perimeter in the queried region and so on. Therefore, each *sensing node* can construct a communication path to each recognized *sensing node*. Hence, all the recognized *sensing nodes* are guaranteed not only cover the queried region but also guaranteed to connect with each other. Each *sensing node* in the sensing nodes discovery phase recognizes the neighboring *sensing nodes* simply from the *1-hop-cover* neighbors that have already been collected. Thus, no additional communication overhead is associated with this phase.

3.3. Query Execution

A sink node that is interested in a specific region may send a sensing query to the center of the queried region through geographical routing [14, 15]. The node i that first receives the query request in the queried region will set its priority $= \infty$ and then flood the request to all the sensor nodes in the region. Each sensor node executes the two-phase protocol after it has received the query request. Node i has the highest priority, so node i will certainly become a *sensing node* in the *self-pruning phase*. After the *sensing nodes discovery phase*, node i becomes a root node and begins to construct a tree named Q-tree. The *sensing node* i unicasts a *construct* packet to each of its neighboring *sensing nodes*. After the *sensing nodes* receive the *construct* packet from node i, they further unicast to their neighboring *sensing nodes*. Each *sensing node* treats the up-stream *sensing node* that firstly sent the *construct* packet to it as a father node. Finally, the constructed Q-tree will cover all the *sensing nodes* in the queried region and the data sensed by the members of the Q-tree will be sent from the leaf *sensing nodes* to sink node through the root node i.

4. The *k*-coverage Sensing Nodes Discovery

In sensor networks, one important issue is the *k-coverage* problem. The k-coverage problem is to determine whether every point in a specific region is covered by at least k sensors, where k is a predetermined value. In the following, we extend our protocol to find a set of *sensing nodes* can satisfy the k-coverage request in a query execution. Assume that we can get a set of *sensing nodes* called *SN*1 according to the rule 1. If a non-sensing node is aware of its *neighboring nodes* in *SN*1, it can delete these *sensing nodes* from its *1-hop-cover* neighboring set and execute the rule 1 again to determine whether it can be a *sensing node*. After the second iteration, all the *non-sensing nodes* can determine their roles- *sensing nodes* or *non-sensing nodes*. We have another coverage set called *SN*2 to fully cover the queried region if the remaining sensor nodes can fully cover the queried region. It is obviously that the *sensing nodes* in *SN*1 and *SN*2 can satisfy the 2-covered request in a queried region. Applying the above procedures, we can solve the k-coverage request problem. In the following, we propose a rule to find the neighboring *sensing nodes* of a *non-sensing node*.

Rule 3. Let SA_i be the sensing area of a *non-sensing node* i. Then node i can recognize node j as a *sensing node* if there exists at least one sub-area of SA_i that is covered by node j and node j has the highest priority value among all of the nodes that cover the same sub-area.

Property 1. Assume that node i is a *non-sensing node* and can recognize a set of *sensing nodes* H according to rule 3. The sensing areas of *sensing nodes* in H can fully cover the sensing area of the node$i$.

Note that, it is possible that some *nodes* cannot be recognized as *sensing nodes* by their *1-hop-cover* nodes according to rule 3 but they are indeed *sensing nodes*. However, such case is benefit for our protocol. These unrecognized *sensing nodes* would not be deleted by their

1-hop-cover neighbors which are the non-sensing nodes. Thus, there exist more sensor nodes that can be used in the next iteration of self-pruning phase.

In Figure 7 the rectangle drawn from dotted lines represents a queried region and $pri(A) > pri(B) > pri(C) > pri(D)$. According to rule 1, nodes *A* and *B* are *sensing nodes* but nodes *C* and *D* are non-sensing ones. Let *SAc* be the sensing area of sensor node *C* . *SAc* is divided by the perimeters of nodes *A*, *B* , and *D* into six sub-areas in the queried region, SAc_1, SAc_2, SAc_3, SAc_4, SAc_5, and SAc_6. Since SAc_3 and SAc_2 are only covered by nodes *A* and *B*, respectively, nodes *A* and *B* can be recognized as *sensing nodes* by node *C*, according to rule 3. Accordingly, nodes *A* and *B* can also be recognized as *sensing nodes* by node *D*. In Figure 7, if all *non-sensing nodes C* and *D* delete the *sensing nodes A* and*B* from their *1-hop-cover* neighboring set, nodes *C* and *D* will become *sensing nodes* according to rule 1. Consequently, the queried region is 2-coverage by nodes *A, B, C* and *D*.

5. Simulation Results

A simulator is implemented in ANSI *C* to evaluate the performance of the proposed query execution protocol. Sensor nodes are randomly deployed in a region of size 100 m×100 m. The number of deployed sensor nodes varies from 1,000 nodes to 2,500 nodes with an interval of 500 nodes. The communication range of every sensor node is fixed at 4 m, 8 m or 12 m, but the sensing range of each node varies from 4 m to 12 m. The proposed protocol is compared with the distributed greedy method presented in [6]. In the distributed greedy method [6], the candidate sensor node must collect the communication paths whose sensing ranges intersect with those of the *sensing nodes* that have already been added. Here, 6-hop, 3-hop and 2-hop local flooding are used for each candidate sensor node, to collect the candidate paths in each round with communication ranges of 4 m, 8 m and 12 m, respectively. The metrics for comparing performance are as follows.

Number of selected sensing nodes: The number of *sensing nodes* selected for sensing the queried region. Other nodes that are not selected to be *sensing nodes* are normal nodes. The number of *sensing nodes* will affect the power consumption for the queried region and the transmission power of the replied packets.

Control packets overhead: The number of control packets is considered in constructing a *connected sensor coverage set* to sense the queried region.

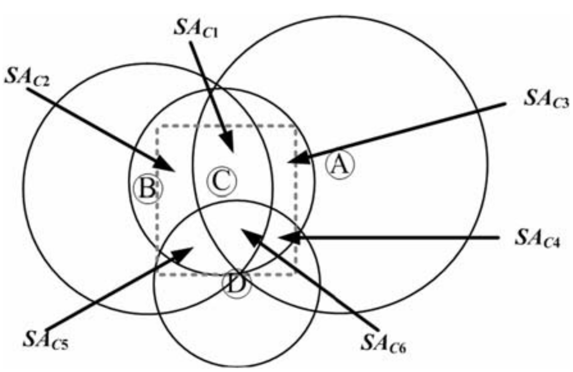


Figure 7. Example of rule 3.

Response time: The passage of time is taken to construct a *connected sensor coverage set* since the node in the queried region first receives the query request from the sink node.

In the proposed protocol, the priority of each sensor node is a main factor in determining whether a node becomes a *sensing node*. Thus, different priority selections will affect the performance of our proposed protocol. The simulation compares the number of *sensing nodes* with three priority selections, which are *remaining energy*, *sensing range*, and *communication degree* (the number of neighbors of a sensor node). If two nodes have the same priority value, then the node with the larger *id* has higher priority. Figure 8 shows the number of selected *sensing nodes* with communication range = 8 m. The simulation shows that using the sensing range as priority yields the fewest *sensing nodes* because the nodes with higher sensing ranges are more likely to be selected as *sensing nodes*. In contrast, using the degree of communication as priority yields the most *sensing nodes* because nodes in a dense area tend to become *sensing nodes*. Although using the remaining energy as priority has more number of *sensing nodes* than that of using sensing range, the network lifetime can be prolonged if remaining energy is used as priority. The sensing range is used as node priorities in the following simulations to compare the performance of the proposed protocol with that of the distributed greedy method.

Figure 9 compares the number of *sensing nodes* selected using the proposed protocol with the numbed selected using the distributed greedy method. The simulation demonstrates that the communication range affects the number of *sensing nodes* selected by the distributed greedy method but not that selected by the presented protocol. The distributed greedy method selects *sensing nodes* one by one, according to the current collected information. When the communication range is 4 m (as shown in Figure 9(a)), the proposed protocol has many fewer *sensing nodes* than the distributed greedy method because the sensor nodes in the proposed protocol can be efficiently divided into two kinds of roles, which are *sensing nodes* and relay nodes. The neighboring *sensing nodes* selected using the proposed protocol must not communicate with each other directly although their sensing ranges intersect with each other. The relay nodes need only transmit packets between the *sensing nodes*, which cannot directly communicate with each other. For example, in Figure 10, the proposed protocol will select nodes A and D as *sensing nodes,* even when they are three hops away. However, the distributed greedy method will select a path of *sensing nodes* from nodes A to D . Therefore, the proposed protocol in that case (*inefficient* case) performs better than the distributed greedy method.

As the communication range increases to 8 m (as shown in Figure 9(b)), the proposed protocol is close to that selected by the distributed greedy method. In the proposed protocol, the number of *sensing nodes* increases slightly when the nodes in the network become dense.

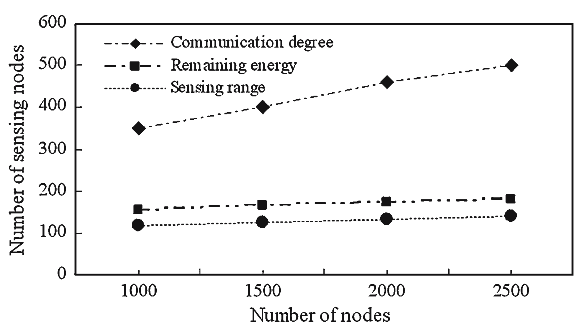
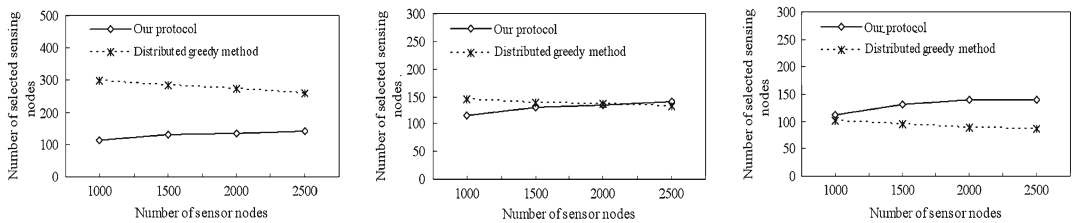


Figure 8. Number of *sensing nodes* selected with three priority selections.



(a) Communication range = 4 m **(b)** Communication range = 8 m **(c)** Communication range = 12 m

Figure 9. The number of *sensing nodes* selected from our protocol and distributed greedy method.

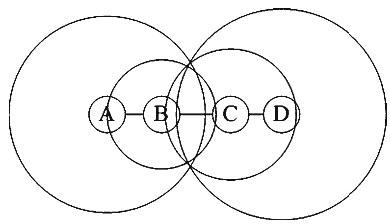


Figure 10. Example of an inefficient case.

However, the high-network density helps the distributed greedy method to select fewer *sensing nodes*. Therefore, the proposed protocol selects more *sensing nodes* than the distributed greedy method as the number of sensor nodes >2000. In Figure 9(c), the number of *sensing nodes* selected by the greedy method is near optimal because the *inefficient* case reduces greatly when the communication range is 12 m. Hence, the greedy method outperforms the proposed protocol when the communication range = 12 m.

Figure 11 compares the percentage of *sensing nodes* selected by the proposed protocol with that determined by the distributed greedy method for communication range = 8 m. The result shows that the proposed protocol and the distributed greedy method can greatly reduce the number of redundant *sensing nodes*. Figure 12 presents the simulated control packet overhead. The control packet overhead obtained using the distributed greedy method increases dramatically with the number of sensor nodes but that obtained using the proposed protocol increases only slightly. In the distributed greedy method, the candidate sensor node selected in each round searches for all of the paths through local flooding. In the proposed protocol, the control packet overhead in each query execution consists of only the overhead of flooding over the queried region and that of the constructing of a Q-tree of *sensing nodes* via unicasting. Figure 13 shows the simulated response times of the presented protocol and the distributed greedy method. The response time of the distributed greedy method is several times greater than that obtained by the proposed protocol, which is more time efficient than the distributed greedy method for various communication ranges.

Table 1 shows the percentages of maximum values of k under various network densities and communication ranges for the k-coverage query request. The simulation of k-coverage sensing nodes discovery is executed in 50 times for each case of network density and communication range. For example, in Table 1, there are 66% of k equal to 4, 32% of k equal to 5, and 2% of k equal to 6 for number of nodes = 2000 and communication range = 8 m. The simulation results show that the number of k is not affected by the communication range. However, the number of k increases as increasing the network density. Figure 14 shows the number of *sensing node*s obtained form the n th iteration in the k-coverage sensing nodes discovery algorithm with

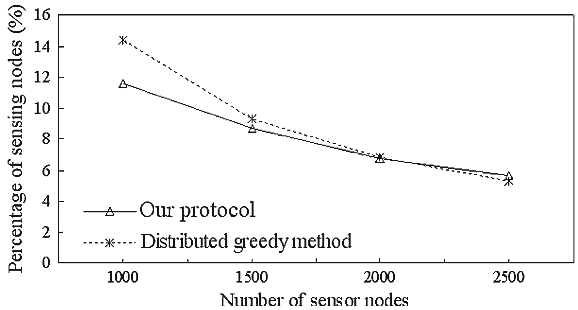
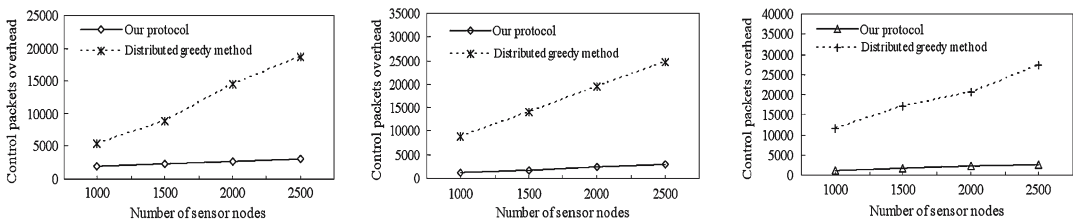
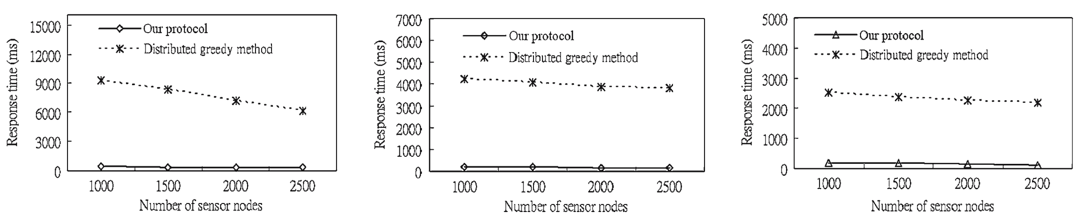


Figure 11. Percentage of *sensing nodes* selected using the proposed protocol and the distributed greedy method.



(a) Communication range = 4 m **(b)** Communication range = 8 m **(c)** Communication range = 12 m

Figure 12. Control overhead for constructing a connected *sensor coverage set.*



(a) Communication range = 4 m **(b)** Communication range = 8 m **(c)** Communication range = 12 m

Figure 13. Time to form a *connected sensor coverage set.*

Table 1. The percentage of maximum values of k under various network densities and communication ranges

	Number of sensor nodes							
	1000		1500		2000		2500	
Communication range	Max. k	%	Max. k	%	Max. k	%	Max. k	%
4	2	90%	2	2%	4	68%	5	24%
	3	10%	3	70%	5	32%	6	72%
			4	28%			7	4%
8	2	92%	3	70%	4	66%	5	20%
	3	8%	4	30%	5	32%	6	74%
					6	2%	7	6%
12	2	88%	2	2%	4	72%	5	22%
	3	12%	3	72%	5	28%	6	74%
			4	26%			7	4%

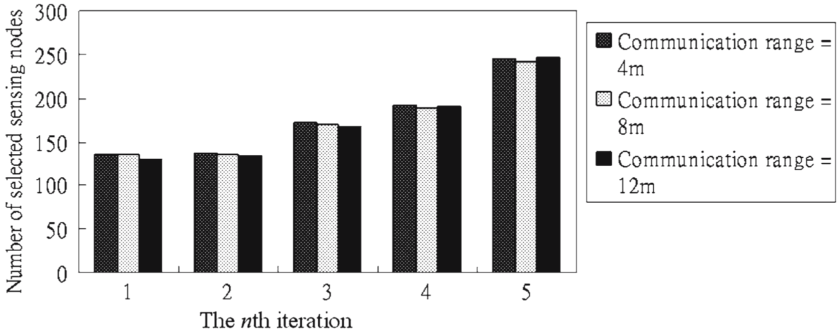


Figure 14. Number of *sensing nodes* selected from each iteration of the k-coverage sensing nodes discovery algorithm with network size = 2500.

network size = 2500, for $1 \leq n \leq 5$. The number of sensing nodes discovery in the $(i+1)$th iteration is larger than ith iteration, for $2 \leq i \leq 4$. In each iteration, the high priority nodes with large sensing ranges will be first selected as sensing nodes to cover the queried region. Thus, the number of sensing nodes to cover the queried region will increase if we use the lower priority nodes in the later iterations. Since there are enough high priority nodes to be used in the first two iterations, the number of sensing nodes selected in the first two iterations are comparable.

6. Conclusions

This work presented an efficient two-phase protocol for selecting the number of sensor nodes to cover the queried region, saving the power consumed by the redundant sensor nodes. Sensor nodes in the proposed protocol are efficiently divided into two groups, according to their roles-*sensing nodes* and relay nodes. The sensing range is used as priority in the simulations here. The simulation results demonstrate that the proposed protocol has fewer *sensing nodes* than the distributed greedy method when the communication range is smaller than the sensing range. Both the proposed protocol and distributed greedy method can effectively reduce the number of redundant sensor nodes. Furthermore, the simulation results show that the proposed protocol has a much lower control packet overhead and a shorter response time than the distributed greedy method, for various communication ranges. Finally, the proposed protocol is extended to solve the k-coverage problem, which can find a set of *sensing nodes* satisfy the k-coverage query request.

Acknowledgements

This work was supported by the National Science Council of the Republic of China under grant NSC 94-2213-E-008-023.

References

1. D. Estrin, R. Govindan, and J. Heidemann, "Embedding the Internet: Introduction", *Communications of the ACM*, Vol. 43, No. 5, pp. 38–41, 2000.
2. J.-P. Sheu, C.-S. Hsu, and Y.-J. Chang, "Efficient Broadcasting Protocols for Regular Wireless Sensor Networks", *Journal of Wireless Communications & Mobile Computing*, Vol. 6, No. 1, pp. 35–48, 2006.

3. S. Slijepcevic and M. Potkonjak, “Power Efficient Organization of Wireless Sensor Networks”, in *Proc. of IEEE Int. Conf. on Communications*, Helsinki, Finland, 2001, pp. 472–476.
4. W. Rabiner Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient Communication Protocols for Wireless Microsensor Networks”, in *Proc. of 33rd Annual Hawaiian Int. Conf. on Systems Sciences*, Hawaii, USA, 2000, pp. 3005–3014.
5. C.-F. Huang and Y.-C. Tseng, “The Coverage Problem in a Wireless Sensor Network”, *in Proc. of the 2nd ACM Int. Conf. on Wireless Sensor Networks and Applications*, San Diego, California, USA, 2003. pp. 115–121.
6. H. Gupta, S.R. Das, and Q. Gu, “Connected Sensor Cover: Self-organization of Sensor Networks for Efficient Query Execution”, in *Proc. of the 4th ACM Int. Symposium on Mobile Ad Hoc Networking & Computing*, Annapolis, Maryland, USA, 2003, pp. 189–200.
7. D. Nicules and B. Nath, “Ad-hoc Positioning System (APS) Using AOA”, in *Proc. of Twenty-Second Annual Joint Conf. of the IEEE Computer and Communications Societies*, San Francisco, California, USA, 2003, pp. 1734–1743.
8. Y.-C. Tseng, S.-P. Kuo, H.-W. Lee, and C.-F. Huang, “Location Tracking in a Wireless Sensor Network by Mobile Agents and its Data Fusion Strategies”, in *Int. Workshop on Information Proc. in Sensor Networks*, Palo Alto, California, USA, 2003, pp. 625–641.
9. D. Tian and N.D. Georganas, “A Coverage-Preserved Node Scheduling Scheme for Large Wireless Sensor Networks”, *in Proc. of First Int. Workshop on Wireless Sensor Networks and Applications*, Atlanta, Georgia. USA, 2002, pp. 32–41.
10. S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava, “Coverage Problems in Wireless Ad-hoc Sensor Networks”, in *Proc. of Twentieth Annual Joint Conf. of the IEEE Computer and Communications Societies*, Anchorage, Alaska, USA, 2001, pp. 1380–1387.
11. S. Shakkottai, R. Srikant, and N. Shroff, “Unreliable Sensor Grids: Coverage, Connectivity and Diameter”, in *Proc. of Twenty-Second Annual Joint Conf. of the IEEE Computer and Communications Societies*, San Francisco, California, USA, 2003, pp. 1073–1083.
12. V. S. A. Kumar, S. Arya, and H. Ramesh, “Hardness of Set Cover with Intersection 1”, *in Proc. of the 27th Int. Colloquium on Automata, Languages and Programming*, Springer-Verlag, 2000, pp. 624–635.
13. G. Pottie and W. Kaiser, “Wireless Integrated Network Sensors”, *Communications of the ACM*, Vol. 43, No. 5, pp. 51–58, 2000.
14. P. Bose, P. Morin, I. Stojmenovic, J. Urrutia, “Routing with Guaranteed Delivery in Ad Hoc Wireless Networks”, *Wireless Networks*, Vol. 7, No. 6, pp. 609–616, 2001.
15. R. Nelson and L. Kleinrock, “The Spatial Capacity of a Slotted Aloha Multihop Packet Radio Network with Capture”, *IEEE Trans.* on *Communications*, Vol. 32, No. 6, pp. 684–694, 1984.

Jang-Ping Sheu received the B.S. degree in computer science from Tamkang University, Taiwan, Republic of China, in 1981, and the M.S. and Ph.D. degrees in computer science from National Tsing Hua University, Taiwan, Republic of China, in 1983 and 1987, respectively. He joined the faculty of the Department of Electrical Engineering, National Central University, Taiwan, Republic of China, as an Associate Professor in 1987. He is currently a Professor of the Department of Computer Science and Information Engineering and Director of Computer Center, National Central University. He was a Chair of Department of Computer Science and Information Engineering, National Central University from August 1997 to July 1999. He was a visiting professor at the Department of Electrical and Computer Engineering, University of California, Irvine from July 1999 to April 2000. His current research interests include wireless communications, mobile computing, and parallel processing. He was an associate editor of Journal of the Chinese Institute of Electrical Engineering, from August 1996 to July 2000. He was an associate editor of Journal of Information Science and Engineering from August 1996 to July 2002. He is an associate editor of Journal of the Chinese Institute of Engineers. He is an associate editor of IEEE Transactions on Parallel and Distributed Systems. He was a Guest Editor of Special Issue for Wireless Communications and Mobil Computing Journal. He was a Program Chair of IEEE ICPADS'2002. He was a Vice-Program Chair of ICPP 2003. He received the Distinguished Research Awards of the National Science Council of the Republic of China in 1993–1994, 1995–1996, and 1997–1998. He was the Specially Granted Researchers, National Science Council, from 1999 to 2005. He received the Distinguished Engineering Professor Award of the Chinese Institute of Engineers in 2003. Dr. Sheu is a senior member of the IEEE, a member of the ACM, and Phi Tau Phi Society.

Shin-Chih Tu received the B.S. degree in information management from Aletheia University, in 1999, theM.S. degree in mathematic science from Aletheia University, Taiwan, in 2001. Since September 2001, he has been working toward the Ph.D. degree in the Department of Computer Science and Information Engineering, National Central University, Taiwan. His current research interests include mobile computing, bluetooth radio system, and ad hoc wireless networks.

Chia-Hao Yu received the B.S. degree in computer science and information engineering from Tamkang University, in 2002, the M.S. degree in computer science and information engineering from National Central University, Taiwan, in 2004.