A Distributed Protocol for Query Execution in Sensor Networks

Jang-Ping Sheu, Chia-Hao Yu, and Shin-Chih Tu
Department of Computer Science and Information Engineering
National Central University, Jhongli, 32054, Taiwan, R.O.C.
E-mail: sheujp@csie.ncu.edu.tw (corresponding author)

***Abstract*- 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, which is represented by the remaining power or sensing area within the queried region. The proposed protocol can efficiently construct a subset of connected *sensing nodes* and respond the query request to the sink node. Simulation results show that the proposed protocol is more efficient and has a lower communication overhead than the existing protocol.**

Keywords*- *sensor network*; *coverage problem*; *query execution*; *connected sensor coverage set.

I. INTRODUCTION

Advances in wireless communication technology and microelectronic fabrication have led to the possibility of establishing a low-cost sensor network with thousands of sensor nodes. The cost effectiveness and ease of deployment make wireless sensor networks the most appropriate candidates for many applications, such as battlefield surveillance, inventory tracking, biomedicine, hazardous environment exploration, home security and smart space [1]. 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. 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 should be able to determine whether to sense a region and confirm the coverage [3] and connectivity in a distributed way.

A greedy method for query execution has been proposed in [5], to find a *connected sensor coverage set*. That work presented a centralized version of an approximation algorithm, and a distributed version converted from the centralized one. However, the centralized algorithm is not easily adapted to a large-scale region. The distributed algorithm is executed in a sequential approach, and the process requires a long n time (that is proportional to the number of selected *sensing 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* and *communication degree* (the number of neighbors of a sensor node) can represent the *priority*. If two nodes have the same *priority 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. In the second phase, each *sensing node* is aware of other neighboring *sensing nodes* and connects with 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, so a quick response with low overhead and high scalability is achieved. Simulation results demonstrate that the proposed protocol is more time-efficient with a lower communication overhead than the greedy method presented in [5].

The rest of this paper is organized as follows. Section 2 introduces the environmental assumptions. Section 3 presents the proposed two-phase protocol. Section 4 evaluates the performance of the proposed protocol by simulations. Section 5 concludes the paper.

II. 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 signals to all nodes within its transmission range. That is, a node *A* can directly send packets to a node *B* within its transmission radius, otherwise it sends via intermediate nodes to relay the packets. Each wireless sensor node is static and is aware of its own location [4][2] either through the GPS, triangulation-based positioning protocols or other positioning methods. Additionally, all the sensor nodes in a sensor network are assumed to be 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

This work was supported by the National Science Council of Republic of China under grants NSC93-2752-E-007-003-PAE and NSC92-2213-E-008-006.

sensor nodes, that covers the queried region [3][8] and can answer the query [9]. Note that, the selected sensor nodes to cover the queried region are named *sensing nodes* (working nodes) in this paper.

III. Coverage Set Determination Protocol

The proposed protocol has two phases - *self-pruning phase* and *sensing node 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 node 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 $NB(i) = \{ j \mid \text{where } SA_i \cap SA_j \neq \phi \}$. 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*. Each node will rebroadcast the received node information only if its sensing range is intersecting with the received sensing range. The sensor nodes have different sensing ranges and communication ranges, so 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.

A. 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 $NPri(i) = \{ j \mid pri(j) > pri(i) \text{ and } j \in NB(i)\}$ 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 $NPri(i)$. 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* iff the sensing area SA_i is not completely covered by $SA(NPri(i))$.

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 $NPri(S) = \{S_1, S_2, \ldots, S_n\}$ 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 S_k that is not a *sensing node*, where $1 \leq k \leq n$. According to rule 1, SA_{sk} must be fully covered by $SA(NPri(S_k))$. Assume $NPri(S_k)$ 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}, …, and SA_{hm} completely. Consequently, the queried region will be fully covered by the selected *sensing nodes* according to rule 1.

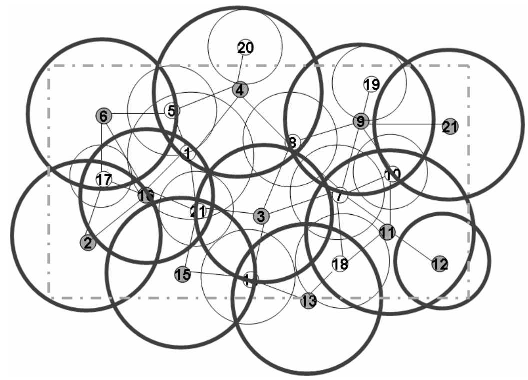


Figure 1. Result of *self-pruning phase*.

An example in Fig. 1 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. The sensor nodes have different communication ranges, so the lines between the nodes indicate that the nodes can directly communicate with each other. The queried region is denoted by the rectangle drawn in dotted line. Priority value of each sensor node is determined by its sensing radius. In Fig. 1, node 1 is not a *sensing node* because the sensing areas SA_3, SA_4, SA_6, SA_2, SA_{15} and SA_{16} completely cover the sensing area SA_1.

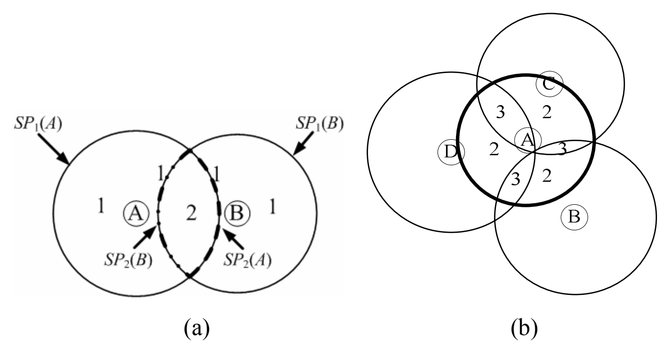


Figure 2. (a) Determining the coverage level of each sub-areas (b) Node *A* is fully covered by nodes *B*, *C* and *D*.

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 [3]. Notably, the 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

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

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. 2(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.

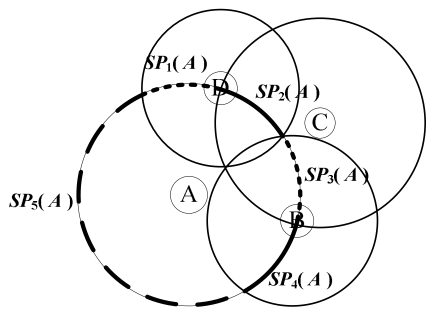


Figure 3. Example of rule 2.

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 Fig. 2(a) suppose $Pri(A) < Pri(B)$. The sub-perimeters of A are $SP_1(A)$ and $SP_2(A)$. The sub-perimeter of B located in SA_A is $SP_2(B)$. According to the location and sensing radius of node B, node A can determine that $SP_2(A)$ is 1-covered by node B; $SP_2(B)$ is 1-covered by node A, and $SP_1(A)$ is not covered by any node with a higher priority than node A. Node A can further determine the coverage degree of each sub-area in SA_A, according to the covering degree of each sub-perimeter. Similarly, the sub-perimeter $SP_2(B)$ is 1-covered, so node A can determine that the sub-areas inside and outside of $SP_2(B)$ are 2-covered and 1-covered, respectively. Thus, node A has the coverage degree of each sub-area in its sensing range and is aware that it is not fully covered by node B. Fig. 2(b) shows another example. 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.

B. Sensing Node Discovery Phase

After the *self-pruning phase*, the whole queried region is covered by a subset of *sensing nodes*. However, 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 neighbors set* become the *sensing nodes* so as to connect to them. Here, the *sensing node discovery phase* is presented to identify the *sensing nodes* from its set of *1-hop-cover* neighbors. 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 $NB(i)$.

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.

Figure 3 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. 3, 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.

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 SA_j 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 $NB(i)$. Our protocol is not affected by the uncertain case because any *sensing node* 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. Consider Fig. 1 as an example; *sensing node* 3 can find *sensing nodes* 4, 9, 11, 13, 15 and 16 that cover its perimeter *P*(3) by rule 2. *Sensing nodes* 4, 9, 11, 13, 15 and 16 will further find *sensing nodes* 2, 6, 12 and 21 to cover their perimeters. Finally, *sensing nodes* 2, 3, 4, 6, 9, 11, 13, 15, 16 and 21 fully cover the queried region. Each *sensing node* in the *sensing node 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.

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 [6][7][10]. The node *i* that first receives the query request in the queried region will set its priority = ∞ 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 node 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 upstream *sensing node* that initially 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 root node *i*.

IV. 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 x 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 [5]. In the distributed greedy method [5], 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 12m, 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.

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

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.

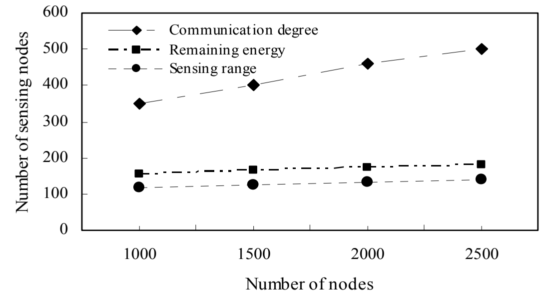


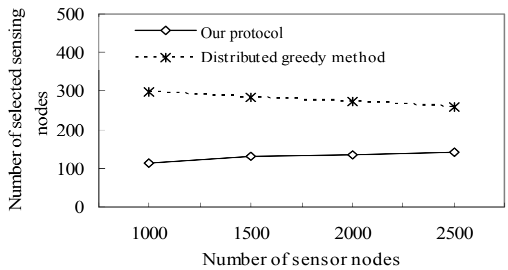
Figure 4. Number of *sensing nodes* selected with three priority selections.

In the proposed protocol, the priority of each sensor node is a main factor in determining whether a node becomes a *sensing node* or not. Figure 4 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.

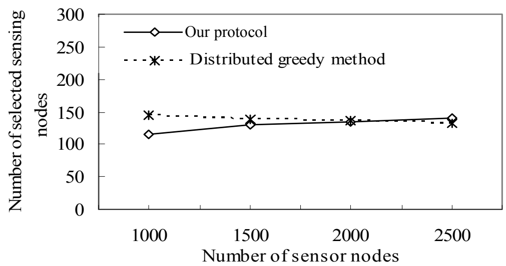
The simulation result in Fig. 5 demonstrates that the communication range affects the number of *sensing nodes* selected by the distributed greedy method but does not affect that selected by our protocol. When the communication range is 4 m (as shown in Fig. 5(a)), the proposed protocol has many fewer *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 relay nodes need only transmit packets between the *sensing nodes* which cannot directly communicate with each other.

For example, in Fig. 6, 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 performs better than the distributed greedy method. As the communication range increases to 8 m (as shown in Fig. 5(b)), the number of *sensing nodes* selected by the proposed protocol is close to that selected by the distributed greedy method. In Fig. 5(c), the number of *sensing nodes* selected by the greedy method is near optimal since the *inefficient* case reduces greatly when the communication range is 12 m.

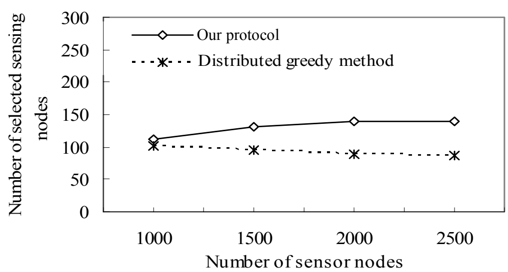
Figure 7 presents the simulated control packets overhead. The control packets overhead obtained using the distributed greedy method increases dramatically with the number of sensor nodes but that obtained using the proposed protocol increases only slowly.



(a) Communication range = 4 m



(b) Communication range = 8 m



(c) Communication range = 12 m

Figure 5. The number of *sensing nodes* selected from our protocol and distributed greedy method.

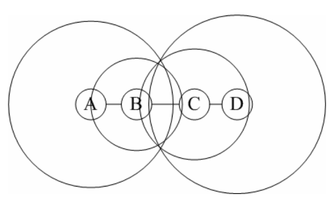
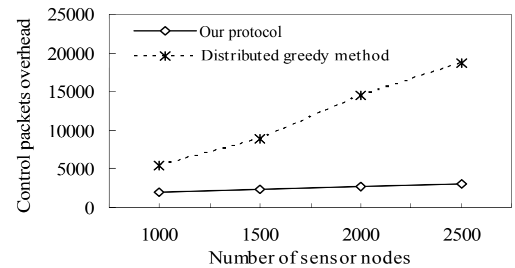
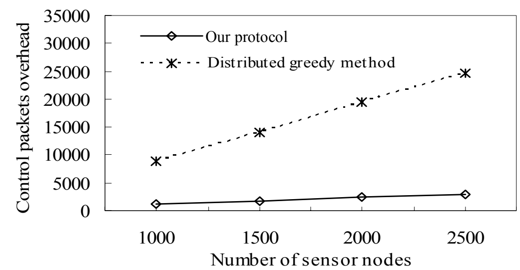


Figure 6. Example of an inefficient case.

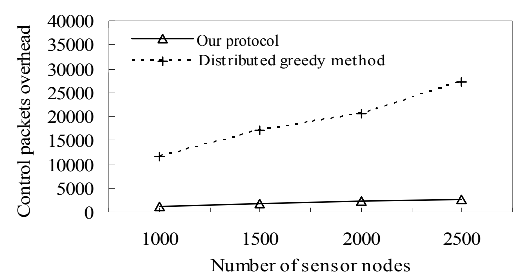
In the proposed protocol, the control packets overhead in each query execution includes only the overhead of flooding over the queried region and that of the construction of a *Q*-tree of *sensing nodes* via unicasting. The construction of a *Q*-tree is executed only by the *sensing nodes* and relay nodes if two *sensing nodes* cannot directly communicate with each other. Hence, the overhead of constructing a *Q*-tree depends on the number of *sensing nodes* in the queried region. Figure 7 shows that the proposed protocol has a lower control packets overhead than the distributed greedy method for various communication ranges. The control packets overhead obtained by the distributed greedy method will increase with the communication range. Although the distributed greedy method has fewer *sensing nodes* than the proposed protocol when the communication range = 12 m, the control packets overhead is considerable at the large communication range.



(a) Communication range = 4 m



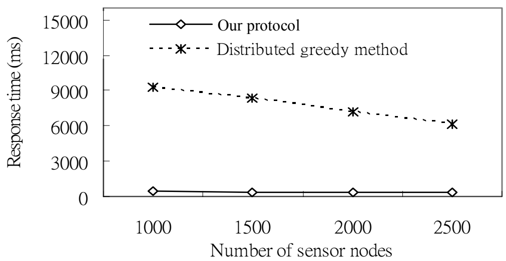
(b) Communication range = 8 m



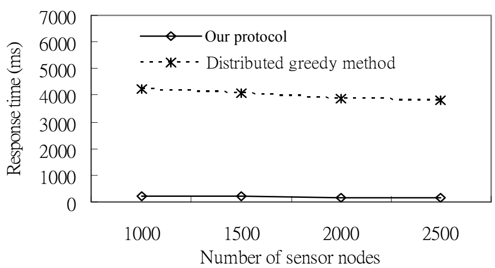
(c) Communication range = 12 m

Figure 7. The control packets overhead to construct a *connected sensor coverage set.*

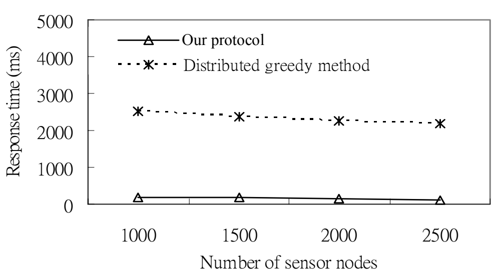
Figure 8 shows the simulated response time consumed by the proposed protocol and the distributed greedy method. Time to construct a *connected sensor coverage set* depends on the number of communication steps taken to complete the set. The number of communication steps in the proposed protocol depends on the maximum number of hops from the node that first receives the query request to any node in the queried region. The number of communication steps also depends on the maximum number of hops from the root node to any leaf node in the *Q*-tree, during the construction of the *Q*-tree. The size of the queried region and the communication range of each node are fixed, so only the density of nodes in the queried region affects the number of communication steps. Thus, fewer communication steps are required for any two nodes in the queried region as the density of nodes increases. The high-density environment also helps the distributed greedy method to find a communication path between a pair of nodes. Therefore, the distributed greedy method has a shorter response time as the density of nodes increases. However, 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, as shown in Fig. 8.



(a) Communication range = 4 m



(b) Communication range = 8 m



(c) Communication range = 12 m

Figure 8. Time to form a *connected sensor coverage set*.

V. 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. A *Q*-tree is constructed to connect all the *sensing nodes* to collect the sensing data and these data back to the sink node. 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 packets overhead and a shorter response time than the distributed greedy method, for various communication ranges.

REFERENCES

[1] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, J. Zhao, "Habitat Monitoring: Application Driver for Wireless Communications Technology," in *Workshop on Data Communication*, pp. 20-41, Latin America and the Caribbean, Costa Rica, April 2001.

[2] A. Savvides, C. C. Han, and M. B. Strivastava, "Dynamic Fine-grained Localization in Ad-Hoc Networks of Sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, (MobiCom), pp. 166-179, Rome, Italy, July 2001.

[3] C.-F. Huang and Y.-C. Tseng, "The Coverage Problem in a Wireless Sensor Network," in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications* (in conjunction with ACM MobiCom 2003), pp. 115-121, San Diego, California, USA, September 19, 2003.

[4] D. Nicules and B. Nath, "Ad-hoc Positioning System (APS) Using AOA," in *Proceedings of Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1734-1743, April 2003.

[5] H. Gupta, S. R. Das, and Q. Gu, "Connected Sensor Cover: Self-organization of Sensor Networks for Efficient Query Execution," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing* (ACM MobiHoc'03), pp. 189-200, Annapolis, Maryland, USA, June 1-3, 2003.

[6] 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, November 2001.

[7] R. Nelson and L. Kleinrock, "The Spatial Capacity of a Slotted Aloha Multihop Packet Radio Network with Capture," *IEEE Transactions*, *Communications*, Vol. 32, No. 6, pp. 684–694, Yorktown Heights, NY, USA, June 1984.

[8] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks," in *Proceedings of Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1380-1387, April 2001.

[9] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable Sensor Grids: Coverage, Connectivity and Diameter," in *Proceedings of Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1073-1083, 30 March-3 April 2003.

[10] T.-C. Hou and V. O.K. Li, "Transmission Range Control in Multihop Packet Radio Networks," *IEEE Transactions*, *Communications*, Vol. 34, No. 1, pp. 38–44, Holmdel, NJ, USA, January 1986.

 0-7803-8966-2/05/$20.00 © 2005 IEEE