SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2012 Vol. 55 No. 4: 755–764
doi: 10.1007/s11432-011-4336-2

Cluster based node scheduling method for wireless sensor networks

WANG Lei[1,2*], WEI RuiZhong[2] & TIAN ZiHong[2,3]

[1]*College of Software, Hunan University, Changsha* 410082*, China;*
[2]*Department of Computer Science, Lakehead University, Thunder Bay* P7B 5E1*, Canada;*
[3]*Mathematics and Information Science College, Hebei Normal University, Shijiazhuang* 050016*, China*

Received March 23, 2010; accepted August 10, 2010; published online September 9, 2011

Abstract By researching on the node scheduling problem of m-covered and connected sensor networks, a new concept of two-hops-cluster is proposed in this paper, and based on it, a new distributed node scheduling algorithm THCNS for allocating all nodes in the sensor network into $k(k \leqslant m)$ different groups $\{0, 1, \ldots, k-1\}$ is designed, without requiring location information. Our algorithm guarantees that each group to be connected and maintains the coverage ratio with high possibility. Theoretical analysis and simulation results show that it has better performance than previous randomized scheduling scheme, and can prolong the lifetime of the sensor network effectively.

Keywords node scheduling, lifetime, wireless sensor network, two-hops-cluster

Citation Wang L, Wei R Z, Tian Z H. Cluster based node scheduling method for wireless sensor networks. Sci China Inf Sci, 2012, 55: 755–764, doi: 10.1007/s11432-011-4336-2

1 Introduction

Due to their extremely small dimension, sensor nodes have very limited energy supply, and it is usually hard to recharge the battery after deployment, either because the number of sensor nodes is too large, or because the deployment area is hostile. So one challenge in the research of sensor networks is how to obtain a prolonged system lifetime by using the limited energy sources economically. Because of the densely distribution of sensor nodes, node scheduling plays a critical role for energy efficiency in wireless sensor networks (WSNs) [1–4].

Currently, there are many node scheduling methods proposed, which can be classified into the following two major categories: round-based node scheduling and group-based node scheduling. In a round-based node scheduling method, the sensor nodes will execute the scheduling algorithm during the initialization of each round, And according to some kind of competition scheme, some nodes will keep active in the current round, and other nodes will keep sleep instead [5]. This kind of methods requires each sensor node to execute the scheduling algorithm for more than once during its lifecycle. But in a group-based node scheduling method, each node will execute the scheduling algorithm only once after its deployment. After execution of the scheduling algorithm, all sensor nodes will be allocated into some groups. Thereafter, in each of the following time slots, each group of nodes will keep active in turn [6].

*Corresponding author (email: wanglei@hnu.cn)

To design a node scheduling method, there are two major problems to be considered: the maintenance of coverage ratio and connectivity for each group. Since a sensor's sensing range is totally independent of its radio transmission range, it is generally hard to combine and solve the two problems together [6]. Most researchers focus on either the scheduling based on sensing coverage [7–9] or scheduling based on network connectivity [5,10,11] but not both at the same time.

But very few works have been devoted to solving the joint problem. Since it is usually costly to obtain and maintain the location information of each sensor node, the joint scheduling problem is even more difficult if the location of each sensor node is unknown [6,12,13].

If each point in a sensory field is monitored by at least m sensors, then we call the field is m-covered. Sensor nodes are usually densely deployed, i.e. we can assume that the sensory filed is m-covered [14,15]. In this paper, we deal with the following challenging task: Assuming that the sensor network is m-covered and connected, then, without location information, how can we schedule the sensor nodes into $k(k \leqslant m)$ different groups $\{0, 1, \ldots, k-1\}$, such that each group will be still connected and can at the same time maintain the coverage ratio as high as possible?

In order to solve the above problem, we define a new structure named two-hops-cluster in an m-covered and connected sensor network. Using the two-hops-clusters, we present a new distributed group ID assignment algorithm for the nodes in the sensor network. And finally, we give a connectivity maintenance algorithm to guarantee the connectivity for each group.

The rest of the paper is organized as follows: In the next section, we will introduce the randomized node scheduling Scheme proposed in [6]. In Section 3, we will introduce some models and definitions. In Section 4, we will introduce our two-hops-cluster based node scheduling scheme. In Section 5, we will give the performance evaluation of our algorithm. And finally, we conclude the paper in Section 6.

2 Randomized node scheduling scheme

In order to solve the two major problems mentioned in the above section, which are needed in a node scheduling method, Liu et al proposed a scheduling scheme named RSGC (randomized scheduling scheme with guaranteed connectivity) for WSNs in [6], which takes into consideration the maintenance of both coverage ratio and connectivity for each group. Their scheme works under the following conditions: (1) No location information is provided; (2) the radio transmission range and sensing range are independent; (3) there are no extra routing protocols for WSNs; (4) all computations are distributed; (5) there is only a roughly synchronized clock. Based on these above assumptions, the algorithm RSGC proceeds in the following steps.

Step 1: Select a subset randomly. Initially, each sensor node generates a random number i between 0 to $k-1$ and assigns itself to subset i.

Step 2: Propagate minimum hop count. This step starts from the sink node. First, the sink node broadcasts a HOP advertisement message to its neighboring nodes, which contains its minimum hop count to the Sink, its node ID and its subset ID. Then, after receiving a HOP advertisement message, each node will put the message in its buffer first, and then deferring the transmission of the HOP message for a back-off time, finally, it will re-broadcast the HOP message with the minimum hop count to the sink only, and of course, the hop count value in the HOP message will be increased by 1 before re-broadcasting. With this method, the broadcasts of HOP message with a non-minimal hop count to the sink will be suppressed if the HOP message with the actual minimal hop count arrives before the back-off time expires. And obviously, the number of broadcasts for each node depends on the length of back-off time.

Step 3: Exchange information with local neighbors. Each sensor node broadcasts its minimum hop count to the sink, its node ID, its subset ID, the node IDs of its upstream nodes and their subset IDs to its neighboring nodes. Here, the upstream nodes are defined as the nodes from which the current node receives its minimum hop count to the Sink. Each node records and maintains all the information it receives from its neighbors.

Step 4: Enforce the extra-on rule. Based on an extra-on rule and information obtained from step 3, each node decides the extra time slots it has to remain active to ensure global connectivity for each group, and updates its working schedule accordingly. Then the updated working schedule will be broadcasted to its neighboring nodes. Here, the extra-on rule is defined as: If a sensor node A has a downstream node B, which is active in time slot i, and if none of B's upstream nodes is active in that time slot, then A shall work in time slot i also. So A is required to work in some extra time slots, e.g., time slot i in this case.

Step 5: Work according to the new working schedule.

The main purpose of this paper is to give a new scheduling scheme under the same conditions of reference [6], but the performance of the new scheme will be improved significantly.

3 Models and definitions

In this paper, we consider stationary sensor networks in a two-dimensional field and assume that sensor nodes are randomly and independently deployed in the field.

Next, we assume that a sensor's sensing range, denoted by r_s, is defined as the range beyond which the sensor's sensing ability can be neglected. Furthermore, without losing generality, we assume that sensors' communication range r_c is larger than or equal to r_s, which is usually true in practice [16].

We do not assume accurate global time synchronization, which is an extremely hard task for large-scale sensor networks. Instead, our scheduling algorithm permits slight time asynchrony without performance degradation.

Finally, we assume all nodes are encoded from $\underbrace{00\cdots001}_{n}$ to $\underbrace{11\cdots111}_{n}$ according to the coding method of n-dimensional Hyper-cubes [17] before deployment, and the sink is encoded as $\underbrace{00\cdots000}_{n}$ especially. So, after deployment, each node will have a unique ID randomly chosen from $\{\underbrace{00\cdots001}_{n},\ldots,\underbrace{11\cdots111}_{n}\}$ beforehand, and the Sink will have an ID as $\underbrace{00\cdots000}_{n}$.

Next, we give some definitions that will be used in this paper as follows

Definition 1. For any two sensor nodes p_1 and p_2, if the distance between them is less than or equals to the communication range r_c, then the two nodes are called neighboring nodes.

Definition 2. For any sensor node p_1 in an m-covered and connected sensor network, supposing that p_2, ..., p_t are all its neighboring nodes, and for any node p_i in $\{p_1, p_2, \ldots, p_t\}$, supposing that all its neighboring nodes are $p_{i1}, p_{i2}, \ldots, p_{it_i}$, and all the different nodes in $\bigcup_{i=1}^{t}\{p_{i1}, p_{i2}, \ldots, p_{it_i}\}$ are $\{p_1, p_2, \ldots, p_t, p_{t+1}, \ldots, p_s\}$, where $s \geqslant t$, then the node set $\{p_1, p_2, \ldots, p_s\}$ is called the two-hops-cluster of node p_1.

Definition 3. For any node p_1, supposing that its two-hops-cluster is C_1, if a sub two-hops-cluster $C \subseteq C_1$ has t elements, then the sub two-hops-cluster C is called a t-cluster.

Definition 4. For any sensor node p_i, let p_j be a neighboring node of p_i. Supposing that the minimum hops to the Sink from p_i, p_j are H_i and H_j respectively. If $H_i = H_j + 1$, then p_j is called an upstream node of p_i and p_i is called a downstream node of p_j. If $H_i = H_j$, then p_j is called a brother node of p_i. If $p_j > p_i$, then p_j is called an older brother node of p_i. Otherwise, p_j is called a younger brother node of p_i.

4 Two-hops-cluster based node scheduling algorithm

4.1 GID assignment for nodes in two-hops-clusters

From the assumption that $r_c \geqslant r_s$, it's easy to have the following.

Theorem 1 (Two-hops-cluster)**.** Suppose that a given point p inside a monitored field is covered by t sensor nodes $p_1, p_2, \ldots, p_t$. Then for any node p_i in $\{p_1, p_2, \ldots, p_t\}$, the node set $\{p_1, p_2, \ldots, p_t\}$ is in its two-hops-cluster.

Based on the above theorem, we give a distributed GIDs (Group IDs) assignment algorithm for the sensor nodes in a two-hops-cluster of the m-covered and connected sensor network as follows, where $t \geqslant m$.

Algorithm 1 (GANC: GID assignment for nodes in a two-hops-cluster)**.** Suppose that the sensor network is m-covered and connected. For any two-hops-cluster consisting of t ($t \geqslant m$) sensor nodes $p_1, p_2, \ldots, p_t$ with $P_1, P_2, \ldots, P_t$ as their node IDs. Then these nodes can be allocated into $k(k \leqslant m)$ different groups $\{0, 1, \ldots, k-1\}$ according to the following steps:

Step 1. Without loss of generality, assume that $P_1 = \min\{P_1, P_2, \ldots, P_t\}$. Then p_1 declares itself as the cluster head in the two-hops-cluster $\{p_1, p_2, \ldots, p_t\}$. Therefore, p_1 will collect the GIDs of all its cluster members in the two-hops-cluster $\{p_1, p_2, \ldots, p_t\}$. If all nodes have been assigned GIDs already, then the algorithm is terminated. Otherwise, go to Step 2.

Step 2. Suppose that the first $a(a < t)$ nodes $\{p_1, p_2, \ldots, p_a\}$ in $\{p_1, p_2, \ldots, p_t\}$ have been already assigned GIDs $\{g'_1, g'_2, \ldots, g'_a\}$ respectively, and $\{g_1, g_2, \ldots, g_b\}$ are all the different GIDs in $\{g'_1, g'_2, \ldots, g'_a\}$, where $b \leqslant a$.

Case 1 If $b = k$, then for each sensor node $p_j \in \{p_{a+1}, p_{a+2}, \ldots, p_t\}$, p_1 selects a GID $j \in \{0, 1, \ldots, k-1\}$ randomly, and assigns GID j to node p_j.

Case 2 If $b < k$, then let $U = \{0, 1, \ldots, k-1\} \setminus \{g_1, g_2, \ldots, g_b\} = \{u_0, u_1, \ldots, u_{k-b-1}\}$.

Subcase A If $t - a \geqslant k - b$, then p_1 selects $t - a - k + b$ different GIDs $V = \{v_0, v_1, \ldots, v_{t-a-k+b-1}\}$ from $\{0, 1, \ldots, k-1\}$ randomly, and distributes $U \bigcup V$ to nodes $\{p_{a+1}, p_{a+2}, \ldots, p_t\}$ randomly.

Subcase B If $t - a < k - b$, then p_1 selects $t - a$ different GIDs $\{v_0, v_1, \ldots, v_{t-a-1}\}$ from U randomly, and distributes these GIDs to $\{p_{a+1}, p_{a+2}, \ldots, p_t\}$ randomly.

4.2 Distributed GIDs assignment scheme for m-covered sensor network

On the basis of Algorithm 1, we present our new distributed GIDs assignment algorithm for nodes in the whole sensor network as follows:

Algorithm 2 (DGAS: Distributed GIDs assignment scheme)**.** Suppose that an m-covered and connected sensor network has n different sensor nodes, and each node maintains an information table including IDNN(ID of its neighboring nodes), NNLN (neighboring nodes list of its neighbors), and IDNN∩NNLN. Then these n sensor nodes can be allocated into $k(k \leqslant m)$ different groups $\{0, 1, \ldots, k-1\}$ in a distributed way according to the following steps:

Step 1. For each node p_1, let $\{p_2, p_3, \ldots, p_s\}$ be all its active neighbors, $\{P_1, P_2, \ldots, P_s\}$, and let $\{N_1, N_2, \ldots, N_s\}$ denote the IDs and NNLN of $\{p_1, p_2, \ldots, p_s\}$ respectively. Then, according to Definition 2, node p_1 computes out its two-hops-cluster on the basis of $\{P_1, P_2, \ldots, P_s\}$ and $\{N_1, N_2, \ldots, N_s\}$.

Step 2. Node p_1 computes out all the sub t-clusters from its two-hops-cluster, where $t \geqslant m$ and each t-cluster shall include the node p_1 as one of its elements.

Step 3. For each t-cluster computed out by the node p_1, node p_1 assigns GIDs to all the nodes in the t-cluster according to Algorithm 1.

According to Algorithm 2, it's easy to prove.

Theorem 2. Supposing that the whole sensor network is m-covered and connected. After executing Algorithm 2, each node in the sensor network will be assigned a single GID, which belongs to $\{0, 1, \ldots, k-1\}$.

4.3 Distributed maintenance of connectivity for each group

In previous subsections we have given algorithms to evenly allocate the nodes in an m-covered and connected sensor network into $k(k \leqslant m)$ different groups in a distributed way. Next, we will consider the problem of connectivity maintenance for each group. Based on Definition 4 given in Section 3, we can present our connectivity maintenance scheme as follows:

Algorithm 3 (CMEG: connectivity maintenance for each group)**.** Suppose that a sensor network is connected, and each node maintains a list of all its upstream nodes and brother nodes. Then after the process of GIDs assignment, each node in the sensor network will update its GID List to maintain the connectivity for each group in the network according to the following steps, where the initial GID List just contains one GID obtained from Algorithm 2.

Step 1. For a node p_i, let L_i be its GID list. For any $g \in L_i$:

Step 1.1. If there is neither an upstream node nor a brother node having g in its GID List, then p_i selects one of its upstream nodes, p_j, which has the shortest GID List (Of course, here we can also consider the surplus energy level of these nodes simultaneously), and sends an AGAC (appended GID application for connectivity) message to p_j.

Step 1.2. If there are no upstream nodes but there are brother nodes, which have g in their GID lists, then p_i selects one of its brother nodes, p_j, which has the shortest GID List (Of course, here we can also consider the surplus energy level of these nodes simultaneously), and sends an AGAC (appended GID application for connectivity) message to p_j.

Step 2. After the GIDs assignment, each node in the sensor network maintains a back-off timer.

Step 2.1. If the node p_j received AGAC messages from one of its downstream nodes or younger brother nodes p_i before time out, then it will update its GID List according to the AGAC messages.

Step 2.2. If the node p_j has received AGAC messages from one of its older brother nodes p_i before time out, and p_j finds out that it has no upstream nodes having g in their GID Lists, then it will select one of its upstream nodes, p_x, which has the shortest GID list (Of course, here we can also consider the surplus energy level of these nodes simultaneously), and sends an AGAC (appended GID application for connectivity) message to p_x.

Step 2.3. After time out, it will broadcast its updated GID List to all its neighboring nodes.

It's easy to prove that Algorithm 3 has the following useful result:

Theorem 3. Suppose that the whole sensor network is connected. Then after executing Algorithm 3, each node in the whole sensor network will have a path with the minimum hops to the Sink and there is at least one same GID in all the nodes in the path at the same time.

4.4 Connectivity and coverage maintenance scheduling algorithm

Now we are in a position to give our new connectivity and coverage maintenance scheduling algorithm.

Algorithm 4 (THCNS: two-hops-cluster based node scheduling)**.** Suppose that a sensor network is m-covered and connected. Then all nodes in the sensor network can be scheduled into k different groups $\{0, 1, \ldots, k-1\}, k \leqslant m$, according to the following steps:

Phase (1) GIDs assignment phase.

Step 1. Each sensor node p_i maintains an information table IT_i which includes information IDNN (ID of its neighboring nodes), NNLN (neighboring nodes List of its neighbors), IUN (identification of upstream node), TSHS (the shortest hops to the sink), and the parameter k. Initially, IT_i is an empty table.

The sink broadcasts a Hello message with the parameter k. Each node that received this message will record k and set TSHS at 1.

Step 2. Each node with non-empty TSHS broadcasts a Hello message including its ID, k and TSHS.

For a node p_i, if it receives a message including the TSHS_j and parameter k from node p_j, then it compares the TSHS_j with the TSHS_i recorded in its IT_i. If the TSHS_i is empty, then it records the ID of p_j (in IUN), the TSHS ($=\text{TSHS}_j+1$) and parameter k into IT_i. If the TSHS_i is not empty, then it checks whether there is $\text{TSHS}_i = \text{TSHS}_j + 1$, and the node ID of p_j will be recorded in IUN if the equation holds true (Here, we assume that the TSHS included in the message that arrived at p_i first will not be bigger than that arrived later), and of course, p_i will also updates IDNN in IT_i when it receives these messages.

Step 3. Each node generates and broadcasts a message including ID and IDNN. Through information exchanges among neighboring nodes, each node confirms its active neighboring nodes, and forms its NNLN in the information table.

Step 4. Run Algorithm 2 to assign GIDs.

Phase (2) Connectivity maintenance phase.

Step 5. Run Algorithm 3 to maintain connectivity for all groups.

Phase (3) Group working phase.

Step 6. All groups work in turn at their given time slots. At the time slot t, if $t \equiv g \bmod k$, then all nodes in group g will keep active, while other nodes will hibernate.

5 Performance analysis and evaluation

5.1 Theoretical analysis

Lemma 1. Suppose k different colours are used to fill $t(k \leqslant t)$ spots randomly. Then the probability that these spots have $j(j \leqslant k)$ different colours is

$$\frac{1}{k^t}\sum_{i=0}^{j-1}(-1)^i \left(\begin{array}{c} k \\ j \end{array} \right) \left(\begin{array}{c} j \\ i \end{array} \right) (j-i)^t.$$

Proof. There are in total k^t different ways to use k colours to fill t spots. On the other hand, it is well-known that the number of onto functions from t elements to j elements is

$$\sum_{i=0}^{j-1}(-1)^i \left(\begin{array}{c} j \\ i \end{array} \right) (j-i)^t$$

and there are $\binom{k}{j}$ different ways to choose j colours from k colours. So the conclusion follows.

Define

$$P(t,k,j) = \frac{1}{k^t}\sum_{i=0}^{j-1}(-1)^i \left(\begin{array}{c} k \\ j \end{array} \right) \left(\begin{array}{c} j \\ i \end{array} \right) (j-i)^t. \tag{1}$$

Then it's easy to have the following

Lemma 2.

$$\sum_{j=1}^{\min\{k,t\}} P(t,k,j) = 1.$$

Using the second kind Stirling number:

$$S(n,j) = \frac{1}{j!}\sum_{i=0}^{j-1}(-1)^i \left(\begin{array}{c} j \\ i \end{array} \right) (j-i)^n,$$

we can write

$$P(t,k,j)=\frac{j!}{k^t}\left(\begin{array}{c}k\\ j\end{array}\right)S(t,j).$$

Theorem 4. Suppose that the sensor network is m-covered and connected, and Algorithm GANC has been used to allocate the nodes into k ($k\leqslant m$) different groups. If a given point p in the sensory field is covered by $t(t\geqslant m)$ sensor nodes, then the probability that there are at least k different nodes having different GIDs in the t sensor nodes can be estimated as

$$P_{\rm THCNS}=\begin{cases}1, & \text{If } t_2\geqslant k;\\ \dfrac{1}{k^{t_1}}\displaystyle\sum_{j=(k-t_2)}^{\min\{k,t_1\}}\sum_{i=0}^{j-1}(-1)^i\left(\begin{array}{c}k\\ j\end{array}\right)\left(\begin{array}{c}j\\ i\end{array}\right)(j-i)^{t_1}, & \text{If } t_2<k.\end{cases}$$

Here $t_1+t_2=t$, and t_1 is the number of sensor nodes that have been assigned GIDs before applying Algorithm GANC in the t-cluster consisting of these t sensor nodes.

Proof Since the point p is covered by t sensor nodes, these t nodes form a t-cluster according to Theorem 1. When Algorithm 1 is applied to this t-cluster, the cluster head will assign group IDs to these nodes. Let t_1 denote the number of sensor nodes that have been assigned GIDs before. If $t_2\geqslant k$, then according to Algorithm 1, the cluster head will select k different sensor nodes randomly from these t_2 sensor nodes that have not been assigned GIDs before, and assign $\{0,1,\ldots,k-1\}$ randomly to these k nodes. So, the possibility that there exist at least k sensor nodes with different GIDs among these t sensor nodes will be 1 in this case.

When $t_2<k$, by Lemma 2, the probability that these t_1 sensor nodes having been assigned GIDs before will have j different GIDs can be computed as

$$\frac{1}{k^{t_1}}\cdot\sum_{i=0}^{j-1}(-1)^i\left(\begin{array}{c}k\\ j\end{array}\right)\left(\begin{array}{c}j\\ i\end{array}\right)(j-i)^{t_1}.$$

Let

$$P(t_1,k,j)=\frac{1}{k^{t_1}}\sum_{i=0}^{j-1}(-1)^i\left(\begin{array}{c}k\\ j\end{array}\right)\left(\begin{array}{c}j\\ i\end{array}\right)(j-i)^{t_1},$$

Then the probability that there exist k sensor nodes with different GIDs among these t sensor nodes can be estimated as $\sum_{j=(k-t_2)}^{\min\{k,t_1\}}P(t_1,k,j)$, since all these t_2 nodes that have not been assigned GIDs before will be assigned different GIDs according to step 2 of Algorithm 1.

In the algorithm RSGC proposed by [6], each sensor node will choose a random number from $\{0,1,\ldots,k-1\}$ as its GID. For any t-cluster ($t\geqslant k$) in the m-covered and connected sensor network, let $P_{\rm RSGC}$ denote the probability that there are k different GIDs in this t-cluster after executing the algorithm RSGC. Then by using the following formula about Stirling number:

$$S(n+1,j)=S(n,j-1)+jS(n,j),\tag{2}$$

we have

Theorem 5. Suppose that the sensor network is m-covered and connected, and all the nodes need to be allocated into $k(k\leqslant m)$ different groups $\{0,1,\ldots,k-1\}$, Then the new node scheduling algorithm THCNS has better performance of coverage maintenance for each group of sensor nodes than the RSGC scheme. Moreover, we have

$$P_{\rm THCNS}-P_{\rm RSGC}=\begin{cases}\displaystyle\sum_{j=1}^{k-1}P(t,k,j), & \text{if } t_2\geqslant k;\\ \displaystyle\sum_{i=0}^{t_2-1}\frac{1}{k^{t-i}}\left((k-i-1)(k-i-1)!\left(\begin{array}{c}k\\ k-i-1\end{array}\right)S(t-i-1,k-i-1)\right), & \text{otherwise.}\end{cases}$$

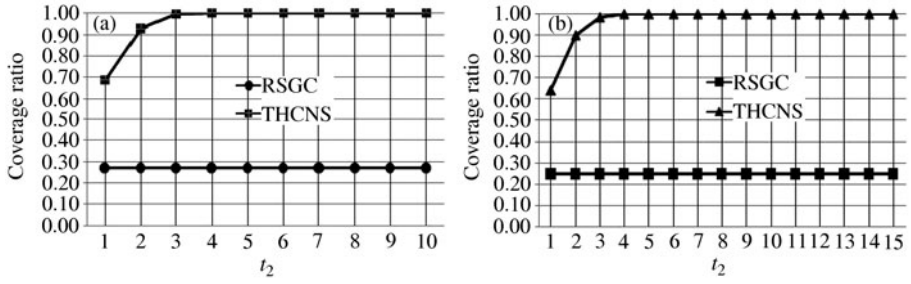

Figure 1 Comparison between THCNS and the randomized scheduling scheme RSGC. (a) $t = 10$ and $k = 6$; (b) $t = 15$ and $k = 8$.

In order to illustrate the results of Theorem 5, Figure 1 shows the actual degree of the improvement of THCNS, where we suppose that a given point p is covered by t different sensor nodes and all the nodes in the whole sensor network need to be allocated into k different groups.

As is evident in Figure 1, the performance of coverage maintenance of THCNS is much better than that of the RSGC, and furthermore, the coverage ratio of the THCNS can reach 100 percent in some cases, but that of the RSGC is very low (about 20 to 30 percent) and cannot reach 100 percent whatsoever.

Next, we will furthermore evaluate the system communication overhead of our scheduling scheme THCNS in terms of the average number of broadcasts needed by sensor nodes.

Step 1. No broadcast is needed for sensor nodes except sink in this step.

Step 2. All sensor nodes need 1 round of information exchange with all its neighbors confirming its active neighboring nodes in this step.

Step 3. If a node p_i receives a new message including the TSHS and parameter k from its neighbors for the first time, then, it needs to exchange 1 round of information with all its active neighboring nodes.

Step 4. By Algorithm 2 and Algorithm 1, we can know that, each node needs only to communicate with its cluster head. So, 1 round of broadcast is needed in this step.

Step 5. By Algorithm-3, we can know that, each node need to know the GIDs list of its upstream nodes. So, 1 round of broadcast of GIDs List is needed in this step.

Step 6. No broadcast is needed in this step.

Therefore, there are totally 4 rounds of broadcast needed in our node scheduling algorithm THCNS, which are distributed in the step 2, step 3, step 4, and step 5 respectively.

And additionally, we will also evaluate the system storage overhead of our scheduling algorithm THCNS as follows:

(1) Firstly, we can know that each node needs to maintain an information table IT consisting of following information: IDNN, NNLN, IUNN, TSHS, and the parameter k. For any given node in the whole sensory field, let $N_{\max}$ denote the maximum number of neighboring nodes, $H_{\max}$ denote the maximum hops to the sink, and N denote the number of sensor nodes in the whole sensor network. Since we encode the nodes according to the n-dimensional Hypercubes, then it is obvious that the length of ID of each node is $\log N$. So, the storage overhead of the information table IT will be no more than $[N_{\max} \cdot \log N + N_{\max}^2 \cdot \log N + \log H_{\max} + \log k]$.

(2) Secondly, each node needs to store the GID List after the GIDs assignment. It is easy to know that the storage overhead of GID List will be no more than $[k \cdot \log k]$. Since there are $k \leqslant m$, and $m \leqslant N_{\max} < N$, then, we can estimate the storage overhead of THCNS to be $O(N_{\max}^2 \cdot \log N)$.

5.2 Experimental evaluation

In order to evaluate the practical performance of the algorithm THCNS, we designed a simulation platform implemented in Java, and compared THCNS with RSGC using the following indexes:

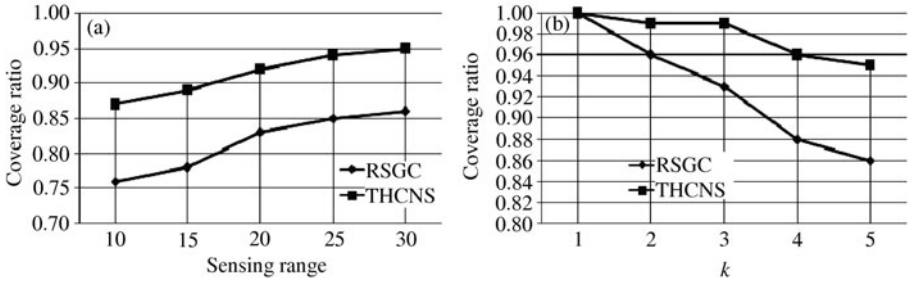

Figure 2 Coverage ratio between THCNS and the randomized scheduling scheme RSGC. (a) Network density vs coverage ratio; (b) k vs coverage ratio.

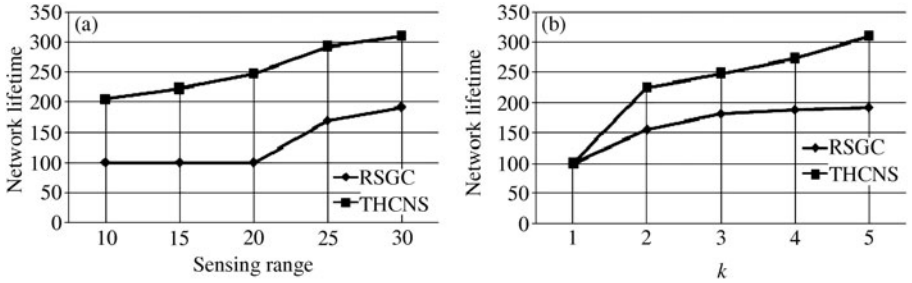

Figure 3 Network lifetime between THCNS and the randomized scheduling scheme RSGC. (a) Network density vs network lifetime; (b) k vs network lifetime.

(1) Coverage ratio for comparing the coverage intensity of the two algorithms in different network density and k (the given number of groups).

(2) Lifecycle of the network for comparing the network lifetime of the two algorithms in different network density and k (the given number of groups). Here, for any group, if all the nodes in a group are not connected any more or the ratio of dead nodes in a group is above 5%, then we regard the group as a dead group.

According to the above two indexes, the simulation results are shown in Figure 2 and Figure 3 respectively. Especially, in Figure 3, only the energy consumptions of working phase are taken into account, and the computation energy consumptions are omitted.

Figure 2(a) illustrates the comparisons of "network density vs coverage ratio" between the two algorithms THCNS and RSGC when $k = 5$. Since the sensing range turns to be larger, any given point in the sensory field can be covered by more sensor nodes, which means that the density of the sensor network is getting higher. So, as shown in Figure 2(a), the coverage ratio of THCNS is much better than that of RSGC.

Figure 2(b) illustrates the comparisons of "k vs coverage ratio" between the two algorithms THCNS and RSGC when $r_s = 30$. As is obvious in Figure 2(b), when the group number k becomes larger, the coverage ratio of RSGC has a drastic drop, but the coverage ratio of THCNS is still relatively high.

Figure 3(a) illustrates the comparisons of "network density vs network lifetime" between the two algorithms THCNS and RSGC when $k = 5$, from which, it is easy to see that the algorithm THCNS can achieve much longer network lifetime than RSGC in different network densities.

Figure 3(b) illustrates the comparisons of "k vs Network Lifetime" between the two algorithms THCNS and RSGC when $r_s = 30$, which also demonstrates that the algorithm THCNS can achieve much longer network lifetime than the algorithm RSGC.

6 Conclusions

Coverage ratio and connectivity preserving methods for densely deployed sensor networks have been reported by many papers, but in all these traditional researches, the location information of sensor nodes

is required, or coverage ratio cannot be guaranteed for each group nor can connectivity be preserved in group-based node scheduling. In this paper, a new distributed node scheduling algorithm is designed, which can allocate all the nodes in the m-covered and connected sensor network into $k(k \leqslant m)$ different groups $\{0, 1, \ldots, k-1\}$ without requiring location information, and at the same time, it can guarantee that each group will be still connected and can maintain high coverage ratio also. Analytical and simulation results demonstrate the effectiveness of our joint scheduling method.

Acknowledgements

This work was supported by Special Fund for Basic Scientific Research of Central Colleges, Hunan University, China and NSERC Discovery (Grant No. 239135-06), Canada.

References

1 Akyildiz I F, Su W, Sankarasubramaniam Y, et al. A survey on sensor networks. IEEE Commun Mag, 2002, 40: 102–114
2 Wang C, Yin Q Y, Wang W J. An efficient ranging method based on Chinese remainder theorem for RIPS measurement. Sci China Inf Sci, 2010, 53: 1233–1241
3 Wang L, Wei R Z. Reputation model based pair-wise key establishment scheme for sensor networks. Ad Hoc Sensor Wirel Netw, 2010, 9: 163–177
4 Tang Q L, Yang L Q, Qin T F, et al. Energy-saving PPM schemes for WSNs. Sci China Ser F-Inf Sci, 2008, 51: 571–585
5 Chen B J, Jamieson K, Balakrishnan H, et al. Span: an energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks. ACM Wirel Netw, 2002, 8: 481–494
6 Liu C, Wu K, Xiao Y, et al. Random coverage with guaranteed connectivity: joint scheduling for wireless sensor networks. IEEE Trans Parall Distr, 2006, 17: 562–575
7 Li X Y, Wan P J, Frieder O. Coverage in wireless ad hoc sensor networks. IEEE Trans Comput, 2003, 52: 753–763
8 TSAI Y R. Coverage preserving routing protocols for randomly distributed wireless sensor networks. IEEE Trans Wirel Commun, 2007, 6: 1240–1245
9 Cortes J, Martinez S, Karatas T, et al. Coverage control for mobile sensing networks. IEEE Trans Robotic Autom, 2004, 20: 243–255
10 Ding Y, Wang C, Xiao L. A connectivity based partition approach for node scheduling in sensor networks. In: Proceedings of DCOSS. LNCS, 2007. 354–367
11 Godfrey P B, Ratajczak D. Naps: scalable, robust topology management in wireless ad hoc networks. In: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks. Berkeley: ACM, 2004. 443–451
12 Wang X, Xing G, Zhang Y, et al. Integrated coverage and connectivity configuration in wireless sensor networks. In: Proceedings of the ACM International Conference on Embedded Networked Sensor Systems(SenSys). Los Angeles: ACM, 2003. 28–39
13 Tian D, Georganas D. Connectivity maintenance and coverage preservation in wireless sensor networks. Ad Hoc Netw, 2005, 3: 744–761
14 Kumar S, Lai T H, Balogh J. On k-coverage in a mostly sleeping sensor network. In: Proceedings of the ACM International Conference on Mobile Computing and Networking(MobiCom). New York: ACM, 2004. 144–158
15 Wu Y W, Li X Y, Liu Y H, et al. Energy-efficient wake-up scheduling for data collection and aggregation. IEEE Trans Parall Distr, 2010, 21: 275–287
16 Lu J, Bao L C, Suda T. Probabilistic self-scheduling for coverage configuration in wireless ad-hoc sensor networks. Int J Perv Comput Commun, 2008, 4: 26–39
17 Wang L, Lin Y P, Liu Y H. Key distribution for group-based sensor deployment using a novel interconnection graph. In: IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM). Helsinki: IEEE, 2007. 1–9