A Distributed Algorithm to Approximate Node-Weighted Minimum α-Connected (θ,k)-Coverage in Dense Sensor Networks

Yongan Wu, Min Li, Zhiping Cai, and En Zhu

School of Computer, National University of Defense Technology,
410073 Changsha, P.R. China
yongan@nudt.edu.cn

Abstract. The fundamental issue in sensor networks is providing a certain degree of coverage and maintaining connectivity under the energy constraint. In this paper, the connected k-coverage problem is investigated under the probabilistic sensing and communication models, which are more realistic than deterministic models. Furthermore, different weights for nodes are added in order to estimate the real power consumption. Because the problem is NP-hard, a *distributed probabilistic coverage and connectivity maintenance algorithm* (DPCCM) for dense sensor networks is proposed. DPCCM converts task requirement into two parameters by using the consequence of Chebyshev's inequality, then activate sensors based on the properties of weighted ε-net. It is proved that the sensors chosen by DPCCM have (θ,k)-coverage and α-connectivity. And the time and communication complexities are theoretically analyzed. Simulation results show that compared with the distributed randomized k-coverage algorithm, DPCCM significantly maintain coverage in probabilistic model and prolong the network lifetime in some sense.

Keywords: probabilistic model; (θ,k)-coverage; α-connectivity; dense sensor networks.

1 Introduction

Generally speaking, a wireless sensor network (WSN) is composed of a large number of small, autonomous sensors scattered in the hazardous or inaccessible environment. Applications of WSN include forest fire detection, vehicle traffic monitoring, battlefield surveillance, and so on [1-3].

The main goal of WSN is to provide information about a sensing field for an extended period of time. The quality of monitoring provided by WSN is usually measured by coverage. k-Coverage ($k \geq 1$) means that each point in the target area is monitored by at least k sensors. How to select appropriate active sensors to preserve required coverage as well as prolong the network lifetime at the same time is the coverage control problem, which is one of the most fundamental problems in WSN. Connectivity is closely-related to coverage, which ensures that there is at least one communication path between any pair of active sensors. The connected k-coverage problem has been studied extensively for more practical [4].

F.P. Preparata, X. Wu, and J. Yin (Eds.): FAW 2008, LNCS 5059, pp. 221–232, 2008.

However, there are some shortcomings in the traditional connected k-coverage protocols. First, some connectivity maintenance protocols assume the deterministic communication models for convenience, where node i successfully sends messages to j if j is in the communication range of i. Though they are accurate in wired networks, previous works have shown that communications of sensors are not deterministic but probabilistic [5]. Second, similarly to above, probabilistic sensing models are more practical than the deterministic ones assumed by some k-coverage protocols [6]. Third, traditional connected k-coverage protocols ignore the difference in power available (PA). They activate as few sensors as possible. Although they cost possibly little energy in one task, the protocols are not optimal in a series of tasks because of unbalanced energy. This also results in hotspots of energy consumption, which may cause premature death of sensors and even premature death of entire network.

In order to solve the problems, we propose a new connected k-coverage problem called *α-Connected (θ, k)-Coverage Set* ((α,θ,k)-*CCS*) problem, where $\alpha\,(0<\alpha<1)$ shows the connectivity metric, and (θ,k) $(0<\theta<1,\ k\in\mathbb{N}^*)$ represents the coverage. α-Connectivity means that any pair of active sensors communicate successfully with probability at least α. And (θ, k)-coverage means that each target point is monitored by at least k sensors with probability at least θ. It is worth of pointing out that θ and k aren't combined into expectation θk. This is because that θk-coverage may infeasible for (θ, k)-coverage. The result is obvious on the supposition that sensors have enough good sensing performance. To the best of our knowledge, this work is the first to address the connected k-overage problem under the probabilistic communication model as well as probabilistic sensing model. Moreover, we take the PA of each sensor into account to activate nodes for energy-consuming balance.

On the other hand, the densely deployment is a common method to avoid blind areas of coverage. In dense sensor networks, the (α,θ,k)-CCS is reduced to a generalization of the connected minimum dominating problem [3]. So we propose a *distributed probabilistic coverage and connectivity maintenance algorithm* (DPCCM) for the node-weighted (α,θ,k)-*CCS* problem in dense WSN because the problem is NP-hard. DPCCM utilizes the properties of weighted ε-net to find "good positions", then expands (θ, k)-flower with α-connectivity by two parameters which can be calculated according to probabilistic model and the request, *i.e.* (α,θ, k). Comparing with the randomized k-coverage algorithm [7], DPCCM significantly maintain coverage and connectivity in probabilistic model and prolong the network lifetime.

The remainder of the paper is organized as follows: Section II reviews the related work in the field. Section III introduces some necessary notations and preliminaries, including the formalization of (α,θ, k)-*CCS* and our approach. The pseudo code and the analysis of the proposed DPCCM are presented in Sections IV. Section V presents the simulation results. The paper concludes in Section VI.

2 Related Work

Because of its importance, the connected k-coverage problem has received significant research attention. Several protocols have been proposed in the literatures. Some of them assume the deterministic models and others assume the probabilistic models.

To the deterministic model, some protocols consider mainly coverage under the condition "the communication range is at least twice the sensing range" [7-9], and others study both coverage and connectivity [4,10,11]. Chakrabarty [8] formulates the k-coverage problem of a set of grid points as an integer linear programming. However, it is known well that localized algorithms (in which simple local node behavior achieves a desired global objective) may be necessary for sensor network coordination. Huang [9] presents a distributed node-scheduling algorithm to turn off redundant sensors. A node decides whether it is redundant only by checking the coverage state of its sensing perimeter. The authors in [7] propose an efficient approximation algorithm to achieve k-coverage in dense sensor networks. They model the problem as a set system for which an optimal hitting set corresponds to an optimal solution for k-coverage. For the connected k-coverage problem, Zhou [10] presents a distributed algorithm, DPA, which works by pruning unnecessary nodes. Wu [11] proposes several local algorithms to construct a k-connected k-dominating set. Yang [4] also presents two distributed algorithms. The first one uses a cluster-based approach to select backbone nodes to form the active node set. The second uses the pruning algorithm based on only 2-hop neighborhood information.

To the probabilistic model, new challenges are introduced in connected k-coverage protocols in sensor networks though they are more realistic. It is the first to address the k-coverage problem under the probabilistic sensing model in [12]. The authors address the problem to activate sensors one by one in a greedy fashion, in which the "contribution" or the "coverage merit" is computed based on the probability of detection of an event by that sensor within its sensing area. The authors in [6] propose a new probabilistic coverage protocol that is fairly general and can be used with different sensing models. However, how to maintain network connectivity is not considered. Hefeeda [13] designs a distributed probabilistic connectivity maintenance protocol that can employ different probabilistic models.

The closest works to ours are [6], [7] and [13]. Unlike DPCCM, node weight, *i.e.* power available is not considered. The algorithms for unweighted case can not be directly applied in weighted one. In addition, associating PCP [6] with PCMP [13] will provide probabilistic coverage and connectivity at the same time, but it only ensures (with probability at least required parameter) that each point in the target area is monitored by at least one sensor. In other words, it is only an approximate algorithm for $(\alpha,\theta,1)$-*CCS*. Therefore, DPCCM for (α,θ,k)-*CCS* is more general.

3 The Node-Weighted (α,θ,k)-CCS Problem and Our Approach

In this section, we formulate the α-connected (θ,k)-coverage set ((α,θ,k)-*CCS*) problem in WSN. Then an overview of our solution is stated with the assumption that sensors are deployed densely and have the same sensing radius R_s and communication radius R_c. Furthermore, localization and time synchronization have been finished, which can be done by many efficient schemes [14,15].

To study connectivity under probabilistic communication model, we represent the network with an undirected weighted simple graph $G=(V,E,c)$ called *communication graph*, where c is a communication probability function c: $V\times V\rightarrow[0,1]$. There exists an edge (i,j) if the distance between i and j is not more than R_c. For an edge (i,j), $c(i,j)$ is the probability of communication between i and j. For a path p: $i\rightarrow j$, $c(i,j)$, called $c(p)$, equals to $\prod c(e)$, where e is an arbitrary edge in p. Thus, for arbitrary i, $j \in V$, $c(i,j)$ is $1-\prod(1-c(p))$, where p is an arbitrary path between i and j in graph G.

To study coverage under probabilistic sensing model, a sensing probability function s: $V \times T \rightarrow [0,1]$ is defined, where T is the target set: if the distance between i and j is not more than R_s, i senses j with probability $s(i,j)$, otherwise $s(i,j)=0$.

Definition 1 (α-Connectivity and (θ, k)-Coverage). Given a communication graph $G = (V,E,c)$ and a sensing probability function s as above, and the target set T. G is said to have α-connectivity if $c(i,j) \geq \alpha$ for arbitrary $i, j \in V$, where $0<\alpha<1$. G is said to have (θ, k)-coverage on T if each element of T is sensed by at least k nodes in V with probability at least θ, where $0<\theta<1$.

Then the α-connected (θ, k)-coverage set problem is formally stated as follows.

Problem 1 (α-Connected (θ,k)-Coverage Set problem, (α,θ,k)-CCS). Given a communication graph $G=(V,E,c)$, a target set T, $0<\alpha<1$, $0<\theta<1$, $k \in \mathbb{N}^*$. Is there a minimum subset V^* of V whose induced subgraph $G[V^*]$ has α-connectivity and (θ, k)-coverage on T.

The above (α, θ, k)-*CCS* is NP-hard, because (1,1,1)-*CCS*, *i.e.* connected cover set problem, as a special case of (α, θ, k)-*CCS* is NP-hard [16].

PA is considered also by node weight in this paper. Generally speaking, the less its PA is, the larger its weight will be, and the smaller the activated probability will be. In this paper, as an example, the weight of node i is defined as $W(i) = \left\lceil a \bullet 2^{-b\left\lceil \frac{PA(i)}{\lambda} \right\rceil} \right\rceil$, where W reflects the relation between PA and activated probability, and the (a,b,λ) are constant parameters based on the type of sensor and environment.

When the target value is continuous variable and deployed sensors are sufficiently dense, area coverage can be approximated by point coverage [4]. That is, it is feasible to select a subset of sensors to cover the rest of sensors. Even now, the problem is still NP-hard because it is reduced to the minimum dominating set problem. We present an approach in dense sensor networks: first find out "good positions" where only a few nodes can cover as many nodes as possible, then activate sensors with small weight around good positions to achieve (θ, k)-coverage and α-connectivity.

The "good positions" problem can be stated that given some weighted disks with the same radius, how to select disks with the minimum total weight to cover all the centers, as shown in Figure 1. In order to reduce the total weight of the nodes around good positions, the weight of disk is defined as follows. According to the theory of geometric disk cover, we adopt a method based on *VC*-dimension and ε-net to find out "good positions". Differing from [7], the weight of node is considered.

Let ω: $2^V \rightarrow \mathbb{R}$, where $\omega(\varnothing)=0$, $\omega(i) = \left\lfloor \frac{W(i)+\sum_{j \in N(i)} W(j)}{|N(i)|+1} \right\rfloor$, $N(i) = \{ j: (i,j) \in E \}$ for $i \in V$ and $\omega(V') = \sum_{i \in V'} \omega(i)$ for $V' \subseteq V$. Let $F \subseteq 2^V$, $|F| = |V|$. Each of F states the nodes covered by the center. Together, the trine (V, F, ω) is a weighted set system.

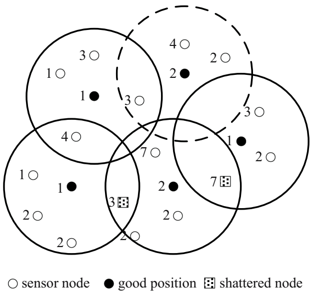

Fig. 1. An example of "good positions" and set shattering: the points are good positions, and the foursquare points are shattered by the four disks with fatter borderlines

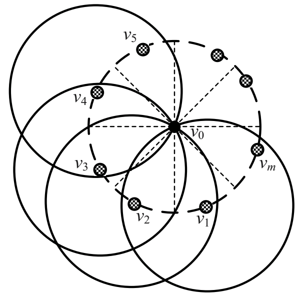

Fig. 2. An example of (θ, k)-flower: v_i senses v_0 with the same probability $p(r)$

Definition 2 (Weighted ε-Net). Given a weighted set system (V, F, ω), $N \subseteq F$ is called a weighted ε-net for (V,F,ω) if $N \cap L \neq \emptyset$ for all $L \in F$ with $\omega(L) \geq \varepsilon\, \omega(V)$.

The weighted ε-net is more general than uniform ε-net, because it is a special case of the former if $\omega(L) = |L|$. We interest in finding small weighted ε-nets. Typically, ε-net finder algorithms are designed for the uniform case. Thus we reduce the weighted case to unweighted one by taking $\lfloor \omega(v) + 1 \rfloor$ copies for $v \in V$, as outlined by [17]. In this paper, we adopt randomly selecting strategy to find ε-nets due to limited computing power and storage space of sensors.

The VC-dimension quantifies how "well behaved" of a set system. VC-dim(ψψψ) is the size of the largest subset of ψψ that is shattered by ψψψ Figure 1 shows an example of the concept of shattering. The authors in [7] prove the set system composed of the set of points in R^2 and all disks with the same radius for each point has a VC-dimension of 3. From Corollary 3.8 in [8], a distributed weighted ε-net finder is designed by randomly selecting biased based on the weight in this paper.

After finding out a weighted ε-net, we simply verify whether it can hit all disks. If it can not, we try another weighted ε-net by *the modified doubling process*. The main idea is to put another weight $\psi(i)$ (initially uniformly) on the node i, and let $\zeta(i) = a2^{-b\omega(i)} + \psi(i)$, where $a2^{-b\omega(i)}$ reflects the relation between node weight and the probability of being the node of a weighted ε-net. If a weighted ε-net doesn't hit some element L of F, we double $\psi(i)$ for all i in L. Then find another weighted ε-net.

With the concept of weighted ε-net, the repeat can be proved only finite times before finding out a hitting set.

Lemma 1. Given a weighted set system (V, F, ω). If there is a hitting set of size c, the modified doubling process as above iterate not more than $6c \log[(n-c)/cq\psi_0 - 1]$ times for $\varepsilon = 1/(2c)$, where $n = |V|$ and $q = \min\{\omega(i): i \in V\}$ and ψ_0 is initially ψ.

Proof. Let H be a hitting set of size c. Let L_j be the element of F that doesn't be hit by a weighted ε-net at the j th iteration, and the weight with subscript j be the one after j iterations. From Definition 2 $\zeta_{j\text{-}1}(L_j) < \varepsilon\zeta_{j\text{-}1}(V)$. So $\zeta_j(V) = \zeta_j(L_j) + \zeta_j(V-L_j) = \zeta_{j\text{-}1}(V) + \psi_{j\text{-}1}(L_j)$. Because $\omega(i) > 0$, we have $\zeta_j(V) < \zeta_{j\text{-}1}(V) + \zeta_{j\text{-}1}(L_j) < (1+\varepsilon)\zeta_{j\text{-}1}(V) < (1+\varepsilon)^j \zeta_0(V) < \zeta_0(V)\, e^{j/2c}$. Moreover, since $H \cap L_j \neq \emptyset$, there is at least one node in H whose

$\psi(i)$ has been doubled. That is, if each $h\in H$ has been doubled $d(h)$ times, then $\sum d(h) \geq j$. We have $\psi_j(H) = \psi_0 \sum 2^{d(h)} \geq c\psi_0 2^{j/c}$. Note that $\psi_j(H) < \zeta_j(V)$, we conclude that $c\psi_0 e^{2j/3c} < c\psi_0 2^{j/c} < \zeta_0(V)e^{j/2c} < [c\psi_0 + (n-c)/q]e^{j/2c}$ from which the proof follows. □

The following is how to achieve (θ, k)-coverage and α-connectivity. The authors in [7] introduce the concept of *k-flower* to guarantee the coverage. Similarly, in order to select *(θ, k)-flower* which is a set of k sensors that all intersect at the center point with probability p, our approach is to choose m center nodes with minimal weight at distance r ($r < R_s$) at m sectors $[2\pi i/m, 2\pi(i+1)/m]$ for $0 \leq i \leq m-1$. Note that the m sensors sense the center with the same probability $p(r)$ for given r. Differing from [7], the m and r are alterable parameters, and how to choose is shown as follows.

Theorem 1. Given r and $p(r)$. Selecting $m_{\min} = \min\{m: mp(r)/(mp(r)-k)^2 \leq 1-\theta$ and $m \geq k\}$ nodes as the above strategy yield a (θ, k)-flower.

Proof. Assume that $\{v_1, v_2, \ldots, v_m\}$ is a (θ, k)-flower with radius r, whose center is v_0, as shown in Figure 2. Let X_i be independent random variable attaining the value 1 when v_i senses v_0 and otherwise the value 0. Let $X = \sum X_i$, then $E[X] = mp(r)$, $\sigma^2 = mp(r)$. We have $P[X \geq k] = 1 - P[X < k] = 1 - P[X - mp(r) < k - mp(r)] \geq 1 - P[|X - E[X]| \geq mp(r) - k] = 1 - P[|X - E[X]| \geq (\sigma - k/\sigma)\sigma] \geq 1 - mp(r)/(mp(r)-k)^2$. The last inequality is derived from a consequence of Chebyshev's inequality that states $P[|X - E[X]| \geq k\sigma] \leq 1/k^2$.

According to the definition of (θ, k)-flower, we have $1 - mp(r)/(mp(r)-k)^2 \geq \theta$. □

For example, let $\theta = 0.8$, $k = 10$ and $p(r) = 0.6$, we find $m_{\min} = 34$. For convenience the following m means $m_{\min}$.

The guarantee of α-connectivity is shown as follows.

Lemma 2. Given a triangular mesh grid. If any pair of neighbors can directly communicate with probability at least $\max\{\alpha, 1/[1+(1-\alpha)^{0.5}]\}$, the triangular mesh has a α-connectivity.

Proof. Statements proven by math induction. First, the proposition is true if $|V| = 3$. Assume it is also true when $|V| = K$ for $K \geq 3$. Let G' with $|V(G')| = K+1$ is a triangular mesh expanded from G, as shown in Figure 3. By Definition 1, we have $c(i, j) \geq \alpha$ for arbitrary $j \in V(G)$. There exist two $i \to j$ paths, where $c(x,i) \geq \alpha$ and $c(y,i) \geq \alpha$ by the assumption, so we have $c(i,j) = 1 - [1 - c(j,x)c(x,i)]\,[1 - c(j,y)c(y,i)] \geq \alpha$. □

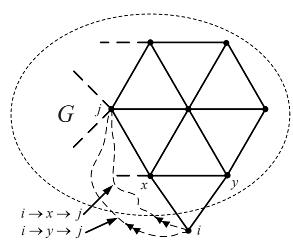

Fig. 3. The triangular mesh expansion

From Theorem 1 and Lemma 2, a sensor can gain appropriate r and m according to its probability model and given (α, θ, k). And the (θ, k)-flower based on $\min\{r, R_s\}$ (for convenience, still named r) and m has α-connectivity and (θ, k)-coverage.

In the following and the simulations, we leave out the communication link with probability less than α for the communication efficiency. This is because that information exchanges between neighbors are very frequent in practical application. Any pair of neighbors should firstly attempt to communicate directly.

4 Distributed Algorithm for Node-Weighted (α,θ,k)-CCS

In the previous section, we propose an approach to node-weighted (α,θ,k)-*CCS*, where the cost of computation is a little, and activating nodes does not rely heavily on global information. Therefore, a distributed algorithm called DPCCM for node-weighted (α,θ,k)-*CCS* is proposed. Its pseudo code is shown in Figure 4.

```
DPCCM SENDER
(1) Initialize parameters
ψ=1; state=TEMP; coverage=0; netSize=1; T=−1; calculate the W; broadcast W to
neighbors and wait for B time units; calculate the ω, ζ, r, m; totalWeight = n*ζ;
(2) Find out "good positions":
while (netSize < n) {
      if (state == TEMP and netSize×ζ/ totalWeight > rand()) {
           state = ACTIVE;
           broadcast OK message containing location to neighbors; break; }
      wait for a constant S time units;
      if (state == TEMP and 1 / (n − netSize) > rand()) {
           ψ= 2ψ; totalWeight = totalWeight + totalWeight / n; calculate ζ; }
      netSize = 2*netSize; }
(3) Verify the coverage and form (θ, k)-flower with α-connectivity:
while (true) {
          if (state == ACTIVE) {
           broadcast VERIFY message containing location to neighbors;
           wait for YES message;   % for a constant R time units
           if (coverage >= m) {break;}
           if (coverage < m) {
                   broadcast FLOWER message containing coverage and location
                   to neighbors; coverage = m; break; } }

DPCCM RECEIVER
if (msg.type == OK and state == TEMP and
     space(msg.source) < min{R_c, R_s})  {break;}
if (msg.type == VERIFY and space(msg.source) < r) {
      coverage = coverage + 1;
      if (state == ACTIVE) {send YES message to msg.source;}
      if (state == TEMP & coverage >= m) { state == SLEEP;}  }
if (msg.type == YES and state == ACTIVE) { coverage = coverage + 1;}
if (msg.type == FLOWER and state == TEMP and space(msg.source) < r) {
      calculate back-off timer T ;   % construction steps as follows
      if (T > 0) {   wait until T == 0; state = ACTIVE;
            broadcast TURNOFF message containing l to its neighbors; }  }
if (msg.type == TURNOFF and T > 0 and l == l.source) { T = −1;}
```

Fig. 4. A distributed algorithm for the node weighted *(α,θ,k)-CCS* (DPCCM)

DPCCM works upon receiving a task from the base station. In initialization, node i calculates and broadcasts $W(i)$ to its neighbors. After B time units $\omega(i)$ is calculated, where B is chosen beforehand to receive W of neighbors. Note that the weighted ε-net finder as above randomly selects nodes biased based on the weight. In order to locally estimate the total initial weight, we regard $\zeta(i)$ as the average weight. The estimation is practical since $\omega(i)$ is average and $\psi(i)$ is the same initially. Finally, it calculates the r and m based on Lemma 2 and Theorem 1.

In the process of finding out the "good positions", DPCCM works in rounds of equal S time units, where S is chosen beforehand according to the environment and the task requirement. In each round, some nodes switch randomly to be in ACTIVE state biased based on the weight $\zeta(i)$, and others uniformly double $\psi(i)$ with probability $1/(n-netSize)$. The reason is an under-covered node double $\psi(i)$ in the modified doubling process, the number of which is less than $n-netSize$. From the proof of Lemma 1, it is feasible to estimate the number by $n-netSize$ because it only increases iteration times. Every node with ACTIVE state broadcasts an OK message to its neighbors. When a neighbor is covered by an active node, it breaks the process of finding "good positions". At the end of each round, double *netSize*.

After $S\cdot\lceil \log n \rceil$ time units, every node with ACTIVE state begins to verify its coverage and connectivity. It broadcasts a VERIFY message containing location to neighbors and waits for R time units, where R is sufficient to reduce collision and guarantee that all neighbors can finish the response work. When a node receives a VERIFY message, it firstly compares r with the distance between itself and the message source. If the distance is more than r it rejects the VERIFY message, or else it checks its state. If its *state* $=$ACTIVE, it replies a YES message to the message source, otherwise it self-increases *coverage* and judges whether its *coverage* reaches k. As soon as a node achieves (θ, k)-coverage, it will change to be in SLEEP state. In R time units, *coverage* of node i self-increase every time receiving a YES message. If its *coverage* is less than k, it will activate some neighbors by broadcasting a FLOWER message to gain (θ, k)-coverage.

The FLOWER message contains location of source and its *coverage*. We prove that, as shown in Theorem 1 and Lemma 2, the centre node of (θ, k)-flower with α-connectivity has at least m active neighbors within radius r. In order to form (θ, k)-flower with α-connectivity, DPCCM chooses nodes with the minimum weight at distance r at $M(i)$ sectors $[2\pi l/M(i), 2\pi(l+1)/M(i)]$ for $0 \le l \le M(i)-1$, where $M(i) = m-coverage$. To make this decision locally, a back-off timer is adopted. The back-off timer $T(j)$ of the receiver j is determined according to its location, $r(i)$ and $M(i)$. The following steps are used in turn to decide $T(j)$: (1) if $d(i,j) < r(i)-\delta$, let $T(j) = -1$, where δ is a small positive constant; (2) if j is in the sector $[2\pi l/M(i), 2\pi(l+1)/M(i)]$, let $T(j) = l\cdot C + C\cdot W(j)/E$, where C is a constant and $E > W(j)$ for arbitrary j. When a sensor times out, the sensor changes its *state* = ACTIVE and broadcasts a TURNOFF message containing l to its neighbors. When a sensor receives a TURNOFF message before the timer expires, it compares l with own: if they are the same, it lets $T(j) = -1$, or else rejects the message.

The algorithm terminates when all sensors are in ACTIVE state or SLEEP state.

As the following, some analyses of DPCCM are shown. First, we prove the correctness of the proposed DPCCM.

Theorem 2. Given a node-weighted sensor network G as above. The active sensors chosen by DPCCM can (θ, k)-cover all nodes in G and have α-connectivity.

Proof: Firstly, we show that the active sensors are guaranteed to hit every sensing disk. In the process of finding out the "good positions", node i doubles $\psi(i)$ with probability $1/(n-netSize)$ until it is activate node or the neighbor of an activated one. Doubling $\psi(i)$ increases the probability to be activated. From Lemma 1, the active node set is a hitting set, otherwise node density is not enough to achieve a hitting set. Then DPCCM activates some nodes to guarantee every active node has m active neighbors within less than radius r. Since the algorithm terminates when all sensors are in ACTIVE state or SLEEP state, both of them satisfy the condition $coverage \geq m$. According to Theorem 1 and Lemma 2, the (θ, k)-flower based on the above r and m has α-connectivity and (θ, k)-coverage. □

The next theorem provides time complexity of DPCCM. We carry out our analysis in terms of the input parameters B, C, R and S, which are discussed in Figure 4. We assume that a message transferred between two neighbors takes one time unit, and so does continuous local computation. And we reduce the communication collision by waiting for some time.

Theorem 3. DPCCM terminates in at most $(m\cdot C+R+5)n+S\cdot\lceil \log n\rceil+B+2$, *i.e.* $O(n)$ time units, where m is determined by (α, θ, k) based on Lemma 2 and Theorem 1.

Proof. According to hypothesis, every node completes the initialization in $B+2$ time units. In the process of finding "good positions", the algorithm iterates for $\lceil \log n\rceil$ steps. Since each iteration works in rounds of S time units, the algorithm costs $S\cdot\lceil \log n\rceil$ time units. Within the following processes till the termination of algorithm, there exist three types of state change: ACTIVE→break, TEMP→ACTIVE→break, and TEMP →SLEEP. To the first, an active node broadcasts a VERIFY message to its neighbors and waits for R time units. Within R time units, either a TEMP node increases its coverage or an ACTIVE node replies a YES message. So it costs R+3 time units. To the second, it is certain that the node receives a FLOWER message from an active node. After it receives the message it will wait for T time units to be activated. From the construction of T, we have $T\leq C(l+1)\leq CM\leq Cm$. So it costs at most $mC+2$ time units. Then it costs $R+3$ time units from ACTIVE to break. To the last, the node receives at least k VERIFY messages and it costs k time units. On the other hand, the above three types of state change are repeated continuously until the algorithm terminates. Since every node is corresponding to one type and only, the total time is $B+2+S\cdot\lceil \log n\rceil+(R+3)\tau_1+(mC+R+5)\tau_2+k\tau_3$, where τ_i is the number of the i th case as above. Note that $m\geq k$ and $\tau_1+\tau_2+\tau_3=n$, the proof can be concluded. □

In the following theorem, we provide the communication complexity of DPCCM algorithm.

Theorem 4. The number of messages broadcasted or sent in the DPCCM is at most $6n$, *i.e.* $O(n)$.

Proof. In the initialization, every node broadcasts its W to its neighbors, so the number of messages is n. In the process of finding "good positions", every active node broadcasts an OK message. The number of active node is less than n, so is the number of OK messages. From the following processes till the termination of algorithm, analysis is similar to the proof of Theorem 3. One “ACTIVE→break” node broadcasts at most two messages: VERIFY and FLOWER. And one “TEMP →ACTIVE→break” node broadcasts at most three messages: TURNOFF, VERIFY and FLOWER. And one “TEMP→SLEEP” node does not broadcast. On the other hand, the YES messages are only sent by active but not break nodes, the number of which is less than n. So the proof can be concluded. □

The approximation factor of DPCCM algorithm is underway. Because it is difficult to analyze theoretically the approximation factor, we test experimentally the performance of DPCCM. The result is shown in Fig.7. In fact, it is exactly our aim to prolong the network lifetime on the requirement of connectivity and coverage. So the experiment result can support the performance of DPCCM in some sense.

5 Simulation

This section presents results from our simulation. The proposed DPCCM algorithm and the randomized k-coverage algorithm, named DRKC [7], were simulated in Prowler, a probabilistic sensor network simulator [19]. To assure the network is dense without coverage hole, 300 sensors are deployed as a grid of points and 300 sensors are randomly placed in a restricted 10×10 area. Some simulation parameters are shown here: sensing range is 1 and so is communication, the initial energy of each sensor is 5000, transmission, reception and idle are 5, 1, and 1, completeness a task consumes 300. The sensing mode is adapted from the exponential model, and the communication model is set to the log-normal shadowing model. The evaluation metrics include the percentage of active sensors with various (α, θ, k), coverage, and the network lifetime.

First we analyze the percentage of active sensors when (α, θ, k) is changed. We vary the requested coverage k between 1 and 8, sensing probability between 50% and 90%, connected probability between 50% and 90%. When one of them is varied, all other parameters are fixed, as shown in Figure 5. The Figure 5(a) and 5(c) shows that the percentage of active sensors increases fast while required sensing probability or required coverage increase. We think that it may be caused by r and m from Theorem 1 and Lemma 2, especially m increases rapidly. However, as is shown in Figure 5(b), the percentage increases slowly with higher connected probability. The result indicates that a triangle mesh is sufficient to ensure probabilistic connectivity.

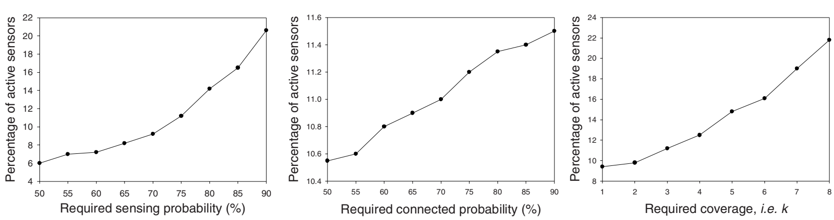

Fig. 5. Analyzing effect on percentage of active sensors when (α, θ, k) is changed: (a) vary $\alpha \in [0.5,0.9]$ while fix $\theta = 0.75$, $k = 3$; (b) vary $\theta \in [0.5,0.9]$ while fix $\alpha = 0.75$, $k = 3$; (c) vary $k \in [1,8]$ while fix $\alpha = 0.75$, $\theta = 0.75$

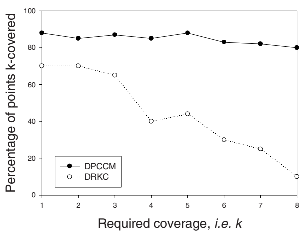

Fig. 6. Comparing the percentage of points k-covered with DRKC

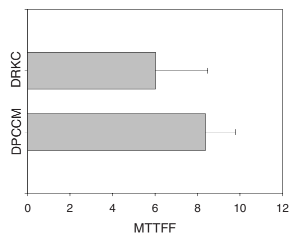

Fig. 7. The MTTFF under DPCCM and DRKC

The coverage by DPCCM and DRKC are compared. The achieved coverage at some random sampling points in the target area has been collected statistically. We fix $\theta = 0.70$ and $\alpha = 0.80$. As shown in Figure 6, DPCCM is significantly better than DRKC. This is important because, under the probabilistic sensing model, the active sensors chosen for deterministic model are not k-covered really.

Finally, we study the MTTFF (*mean time to first failure*) of any given task under DPCCM and DRKC, which can indicate the network lifetime in some application. We randomly gain k from 1 to 8 in each task. As soon as a task can not been completed, we calculate the MTTFF in this experiment. After repeating 50 times, the result is shown in Figure 7. Compared with DRKC the MTTFF under DPCCM has been prolonged about 21%. This is because that DRKC activate as few sensors as possible based on required coverage and isn't always optimal in a series of tasks because of unbalanced energy-consuming among nodes.

6 Conclusion

In this paper, we consider connected k-coverage problems under probabilistic sensing models and probabilistic communication models, which are more realistic than deterministic models. We represent the problems with the α-connected (θ,k)-coverage set problem and formulize it as (α,θ,k)-*CCS*. Moreover, in order to satisfy various coverages and realize energy-consuming balance, we also take power available into consideration with node weight. Because node-weighted (α,θ, k)-*CCS* is NP-hard, a distributed approximate algorithm, named DPCCM, is proposed for dense sensor networks. DPCCM utilizes the properties of VC-dimension and weighted ε-net to find "good positions", then expands (θ, k)-flower with α-connectivity by r and m. The two parameters can be calculated according to the tasks and performance indexes of sensors. We prove the correctness of DPCCM and theoretically analyze the time complexity and communication complexity. We also implement our algorithm in Prowler and compare it against the randomized k-coverage algorithms.

Acknowledgments. This research was supported by the National Natural Science Foundation of China under Grant Nos.60603062 and the Natural Science Foundation of Hu'nan Province of China under Grant No.06JJ3035.

References

1. Estrin, D., Govindan, R., Heidemann, J., Kumar, S.: Next century challenges: Scalable coordination in sensor networks. In: 5th ACM International Conference on Mobile Computing and Networking (MOBICOM 1999), pp. 263–270. ACM Press, Seattle (1999)
2. Cerpa, A., Elson, J., Hamilton, M., Zhao, J., Estrin, D., Girod, L.: Habitat monitoring: Application driver for wireless communications technology. In: Proc. of ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, pp. 3–5. ACM Press, Costa Rica (2001)
3. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless sensor networks: A survey. Computer Networks 38(4), 393–422 (2002)
4. Yang, S., Dai, F., Cardei, M., Wu, J., Patterson, F.: On connected multiple point coverage in wireless sensor networks. Journal of Wireless Information Networks 13(4), 289–301 (2006)
5. Hekmat, R., Van Mieghem, P.: Connectivity in wireless ad-hoc networks with a log-normal radio model. Mobile Networks and Applications 11(3), 351–360 (2006)
6. Hefeeda, M., Ahmadi, H.: A probabilistic coverage protocol for wireless sensor networks. In: 15th IEEE International Conference on Network Protocols (ICNP 2007), pp. 41–50. IEEE Press, Beijing (2007)
7. Hefeeda, M., Bagheri, M.: Randomized k-coverage algorithms for dense sensor networks. In: 26th IEEE International Conference on Computer Communications (INFOCOM 2007), pp. 2376–2380. IEEE Press, Anchorage (2007)
8. Chakrabarty, K., Iyengar, S., Qi, H., Cho, E.: Grid coverage for surveillance and target location in distributed sensor networks. IEEE Transactions on Computers 51(12), 1448–1453 (2002)
9. Huang, C., Tseng, Y.: The coverage problem in a wireless sensor network. In: 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA 2003), pp. 115–121. ACM Press, San Diego (2003)
10. Zhou, Z., Das, S., Gupta, H.: Connected k-coverage problem in sensor networks. In: 13th International Conference on Computer Communications and Networks (ICCCN 2004), pp. 373–378. IEEE Press, Chicago (2004)
11. Wu, J., Dai, F.: On constructing k-connected k-dominating set in wireless networks. In: 19th International Parallel and Distributed Processing Symposium (IPDPS 2005), pp. 81a–81a. IEEE Press, Denver (2005)
12. Lu, J., Bao, L., Suda, T.: Coverage-Aware Sensor Engagement in Dense Sensor Networks. In: Yang, L.T., Amamiya, M., Liu, Z., Guo, M., Rammig, F.J. (eds.) EUC 2005. LNCS, vol. 3824, pp. 639–650. Springer, Heidelberg (2005)
13. Hefeeda, M., Ahmadi, H.: Network connectivity under probabilistic communication models in wireless sensor networks. In: 4th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2007), pp. 1–9. IEEE Press, Pisa (2007)
14. He, T., Huang, C., Lum, B., Stankovic, J., Adelzaher, T.: Range-free localization schemes for large scale sensor networks. In: 9th ACM International Conference on Mobile Computing and Networking (MOBICOM 2003), pp. 81–95. ACM Press, San Diego (2003)
15. Ganeriwal, S., Kumar, R., Srivastava, M.B.: Network-Wide Time Synchronization in Sensor Networks. NESL Technical Report (2003)
16. Gupta, H., Zhou, Z., Das, S.R., Gu, Q.: Connected sensor cover: Self-Organization of sensor networks for efficient query execution. In: 4th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), pp. 189–200. ACM Press, New York (2003)
17. Brönnimann, H., Goodrich, M.: Almost optimal set covers in finite VC-dimension. Discrete&Computational Geometry 14(4), 463–479 (1995)
18. Haussler, D., Welzl, E.: Epsilon-nets and simplex range queries. Discrete and Computational Geometry 2(1), 127–151 (1987)
19. Simulator can be downloaded from, http://www.isis.vanderbilt.edu/Projects/nest/prowler/