Computer Networks 55 (2011) 2856–2872

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Limited mobility coverage and connectivity maintenance protocols for wireless sensor networks

Prasan Kumar Sahoo [a,*], Jang-Ping Sheu [b]

[a] Dept. of Computer Science and Information Engineering, Chang Gung University, Kwei-Shan 333, Taiwan
[b] Dept. of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan

ARTICLE INFO

Article history:
Received 16 April 2010
Received in revised form 17 May 2011
Accepted 8 June 2011
Available online 12 July 2011
Responsible Editor: N.B. Shroff

Keywords:
Wireless sensor networks
Coverage
Connectivity
Limited mobility
Network maintenance

ABSTRACT

Coverage and connectivity maintenance is an important research issue in wireless sensor networks, as sensors are deployed randomly over the monitoring region in large numbers. In the post deployment scenario, existing coverage or connectivity of the network is disturbed due to predictable or unpredictable death of the nodes and maintenance of the network manually is difficult due to physical condition of the monitoring region. In this paper, potential connectivity and coverage maintenance algorithms for the wireless sensor networks are proposed that let the sensors work alternatively by identifying the redundant sensing regions in the post deployed scenarios and maintain the network with limited mobility. Decision of mobility of the nodes among immediate neighbors of a dead node is totally autonomous and distributed, and it is made to maintain the network without disturbing the existing coverage and connectivity. Performance evaluation of our protocols shows that the average mobility distance and energy consumption of the nodes are limited to maintain the coverage and connectivity. Moreover, the proposed algorithms could be worked for the communication range (R_c) of a node is equal to sensing range (R_s) or for $R_c < 2R_s$.

1. Introduction

Wireless sensor network (WSN) is a special type of ad hoc network, where nodes form the network dynamically without help of any infrastructure. Normally the nodes are deployed randomly and are supposed to sense a phenomenon, process the collected sensing data in a collaborative manner, and route the results to an end user. For this, the active nodes of the network have to maintain both network connectivity and coverage. The network cannot guarantee the quality of surveillance without sufficient coverage. Besides, data routing cannot be achieved without proper connectivity. In wireless sensor networks, sensors are often intended to work in remote or hostile environments, such as a battlefield or desert and it is impossible to recharge or replace the battery power of the sensors. Hence, few nodes in the network may be dead due to exhaustion of battery power making the network uncovered and disconnected. In wireless sensor network, nodes can be classified into static and mobile nodes. Current research in wireless sensor networks has focused on fixed sensor networks [8,9] in which nodes are static. Static sensor nodes cannot change position by themselves after their deployment. On the other hand, mobile sensors can change their position autonomously, depending on the mission requirements and are able to dynamically adjust network topology to improve the performance of sensor networks.

Recent advances in wireless technology with demands for greater user mobility have provided a major impetus toward the development of a mobile network architecture [1,2]. Unlike the existing schemes in which sensors are stationary, if we deploy mobile sensors in the network such as a battlefield or environmental monitoring region and let few nodes move, the degree of coverage and connectivity

* Corresponding author.
E-mail addresses: pksahooind@yahoo.com (P.K. Sahoo), sheujp@cs.nthu.edu.tw (J.-P. Sheu).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.06.011

can be improved. Besides, it is more economical and versatile than the existing fixed stationary sensor networks and redeployment. Due to mobility of sensor nodes, mobile sensors can change their position depending on the requirement of the missions. Dynamic adjustment of the sensor node's position would change the topology of the nodes and hence promote the performance of sensor networks. When sensors are deployed in a disaster environment, where human interference is not possible, we need mobile sensors to accomplish the tasks such as coverage and connectivity compensation, location assignment and node replacement. In a post deployment scenario, it is possible that some nodes over certain region are destroyed due to intrusion, explosion or due to environmental factors like heat, vibration and failure of electronic components or software bugs in the network. In another scenario, power sources of the nodes may lead death of the nodes, thus affecting the coverage and connectivity of the original network. Hence, it is essential to reconfigure the network by mobile sensors to maintain the connectivity and coverage, and thereby avoiding the network partitions.

The large-scale multi-robot systems has been limited due to size, cost, off-the-shelf hardware platform and complexity of the various robots used. However, recently, several researchers have investigated techniques of mobile sensors, such as MICAbot [3], Mobile Robot [4], CotsBots [5] and Robomote [6] to obtain better solution for many issues. These mobile sensors are inexpensive and modular mobile robots built entirely from commercial off-the-shelf components. These robots provide a convenient platform on which to investigate algorithms, cooperation, and distributed sensing in large robot networks. Each robot is small (13 cm × 6.5 cm base) and costs under US$200. Each is equipped with on-board processing, radio communication, and a base platform for mobility. Mobility strategies have two modes of operation depending on the presence or absence of a detected target. In absence of a detected target that is known as the search mode, mobility strategies can also be classified into two groups, i.e. random and deterministic. In deterministic strategies, the monitoring region could be partitioned a priori into grids and sensor nodes assigned to each grid that becomes the home area of the node. When a node fails in the deterministic mobility setting, the remaining nodes need to quickly become aware of the failure and tries to maintain the network. Under random search strategies [7], each node moves at random and requires minimal coordination, especially when a node fails and experience graceful degradation in surveillance coverage in that event. Random mobility can be made somewhat more efficient by adopting strategies wherein nodes locally repel each other and are less likely to visit areas very recently visited. In presence of a detected target such as data acquisition or target tracking mode, the node that first detects an object becomes a coordinator and broadcasts the detection information to the rest of the network. Then some nodes deterministically move toward the target.

The hybrid and hierarchical mobility patterns could also be devised in which each element of the partition has more than one node and each node employs a random sweep pattern within its home area. In this paper, we focus on wireless sensors with deterministic mobility settings to maintain the network and propose the limited mobility based coverage and connectivity algorithms to maintain the wireless sensor networks dynamically. The main contributions of our work can be summarized as follows.

- We propose the distributed connectivity and coverage maintenance algorithms (*CoCo*), in which $1 \leqslant R_c/R_s < 2$. Since, longer communication range is the main consumer of energy resource, we minimize energy consumption by using shortest communication range.
- To compensate both coverage and connectivity, though we propose the mobility based maintenance algorithms, only one-hop neighbors of a dead node are involved in the mobility process without disturbing the existing network. Besides, the mobility distance of those nodes is limited within one hop so that the power consumption due to mobility is also limited.
- We design algorithms to maintain both communication and coverage problems of WSN, which are reactive as well as proactive by nature due to predictable and unpredictable death of the nodes, respectively.
- Our algorithms are distributed and self organized by nature and can maintain the network integrity dynamically due to death of nodes at multiple locations.

The rest of the paper is organized as follows. Section 2 describes the related work and motivations behind our connectivity and coverage maintenance protocols. System model of our protocols with assumptions and definitions are given in Section 3. Section 4 describes the limited mobility connectivity and coverage maintenance (*CoCo*) protocols. Performance evaluation of our algorithms and comparison of our simulation results with state-of-art algorithms are made in Section 5. Concluding remarks are made in Section 6 of the paper.

2. Related work and motivations

2.1. Related work

Sensing coverage is a fundamental problem in wireless sensor networks and has been well studied over past few years. However, most of the previous works address only one kind of redundancy, i.e. sensing or communication alone. The authors in [8,9] address how to combine consideration of coverage and connectivity maintenance in a single activity scheduling. In both of the work, it is proved that the communication range is at least twice of the sensing range. Based on the deployment nature of the wireless sensor networks, the authors in [10] consider the communication range is twice of the sensing range, which is the sufficient condition and tight lower bound to ensure that complete coverage preservation implies connectivity among active nodes, if the original network topology is connected. Though, all of these said works prove sufficient conditions for the coverage and connectivity, maintenance of the network due to loss of coverage or connectivity is not discussed. A distributed coverage-preserving protocol based on probabilistic detection model is proposed in [35]. The authors use Voronoi diagram to simplify the coverage check algorithm and present an approximate algorithm to evaluate the coverage percentage. However, the authors consider an arbitrary relationship between R_c and R_s based on probability, which is impractical.

The RF communication coverage using modulated backscattering in wireless passive sensor networks is analyzed in [32]. It is analyzed that wireless passive sensor networks designed to operate using modulated backscattering do not have the lifetime constraints of conventional WSN. In [11], authors explore the problem of determining the coverage, provided by non-deterministic deployment of sensors, using a more realistic probabilistic coverage model. They investigate the coverage issues in wireless sensor networks based on probabilistic coverage and propose a distributed coverage algorithm to evaluate the degree of confidence in a randomly deployed sensor network. Motivated with this idea, the authors in [33] propose a probabilistic approach to compute the covered area fraction at critical percolation for each transition of coverage and connectivity. However, these analysis may be useful only for the redeployment scenario, which is impossible due to geographical conditions of the monitoring regions such as harsh terrains. A wireless sensor network k-covers its deployment region, if every point in its deployment region is within the coverage ranges of at least k sensors. In [12], the authors study how the probability of a deployment region being k-covered by randomly deployed sensors changes with the sensing radius or the number of sensors. The work is totally probabilistic in nature and does not consider the connectivity issues.

The work in [13] addresses the area coverage problem with equal sensing and communicating radii. Though, they consider the WSN with equal sensing and communicating radii, their main goal is to minimize the number of selected sensors to be either active or sleep and they do not talk how to manage the connectivity and coverage due to death of a node. In [14], authors have proposed algorithms with which mobile sensors can track and converge on a series of events while also ensuring complete sensor coverage of their environment. However, it requires a large amount of computational cost to track all the sensors in the network and to find motion of every other sensor. A mathematical model to describe the redundancy in randomly deployed sensor networks is proposed in [15], where the authors present simple formulae to estimate the probability that a sensor is completely redundant and to estimate the average partial redundancy. The theoretical analysis of the work discusses observation concerning the minimum and maximum number of neighbors that are required to provide complete redundancy. In [16], authors investigate the criterion to decide whether a sensor is redundant, and propose an improved method to determine, if a sensor is completely covered by its neighbors, especially for boundary sensors. However, the authors consider that the communication range is less than twice of the sensing range and do not talk about the network maintenance.

In [17], nodes only use their sensed information in making the decision to move, making it not a cost effective solution to the coverage problem. In [18], Voronoi concept is used to discover the existence of coverage holes and a sensor node compares its sensing disc with the area of its Voronoi polygon to estimate any local coverage hole. A novel location-free coverage maintenance scheme in wireless sensor networks that exploits power control and radio connectivity information in constructing sparse structures in wireless sensor networks is proposed in [19]. They have studied the coverage property and have established the connection between area coverage and point coverage through analysis. However, the connectivity maintenance has not been included in their study. The problem of achieving k-coverage in the presence of mobile sensors is addressed in [20] to guarantee that a particular area is covered most of the time to a specific degree. However, the condition of connectivity has not been taken into account during coverage preservation. A mobility based coverage maintenance algorithm is proposed in [21]. They propose how to determine a movement plan for the sensors in order to maximize the sensor network coverage, considering a fixed and variable distance mobility model, though connectivity model is not designed in their work. A distributed algorithm for the coordinated coverage fidelity (Co-Fi) maintenance in sensor networks is proposed in [22], where the two hop mobile nodes are used to repair the coverage loss in the area being monitored by it. In their proposal, decision of mobility is done based on the residual energy and mobility cost of a node. The dying node notifies the network of its death, which is not practical for unpredictable death. Besides, they use larger communication range (R_c), where $R_c \geqslant 2R_s$ and the two hop neighbors of a dead node are involved to maintain the coverage, thereby consuming more energy due to long average mobility distance.

The coverage hole detection and recovery algorithm for the coordinate-free sensor networks is proposed in [34]. In this work, authors design distributed hole recovery algorithms especially for those scenarios with frequent topology change due to node mobility or unpredictable node failures. In their proposal, if the hole can be covered by circles of radii $2R_c$ centered on each of the boundary nodes, it can be recovered by redundant nodes covered by the boundary nodes. A Dynamic Coverage Maintenance (DCM) scheme is proposed in [23], which exploits the limited mobility of the sensor nodes taking ($R_c = 2R_s$). Though, authors propose a set of coverage maintenance schemes with help of one-hop neighbors of a dead node, their algorithms do not specify for unpredictable death of the nodes. A sensor movement control strategy is proposed in [24], where a commander controls a cluster of mobile sensors to monitor a target region ahead of the commander, and in the direction of the commander's movement. Once the speed and direction of the movement of the commander are changed, the new positions as well as the speed and direction of the sensor are decided by the algorithm. The commander knows all the information about the sensors through multihop communication and connectivity between sensors during movement is guaranteed. However, the protocol is not distributed, and average mobility distance of the nodes is longer. The existing approaches of coverage connectivity maintenance algorithms with existing challenges and un-

Table 1
Classification of existing approaches and challenges of maintenance problems.

References	Existing approaches	Drawbacks	Existing challenges
Reference 10	Considers $R_c \geqslant 2R_s$, Proposes sufficient conditions of coverage and connectivity	Larger communication range	To maintain both coverage and connectivity with $R_c = R_s$
Reference 12	Computes coverage	Probabilistic in nature, Does not consider connectivity	To maintain both coverage and connectivity with $R_c = R_s$
Reference 13	Considers $R_c = R_s$ Minimizes number of selected sensors to be active or sleep	Does not consider maintenance, though considers $R_c = R_s$	To maintain both coverage and connectivity due to death of nodes
Reference 16	Considers $R_c < 2R_s$	Investigates redundancy criterion	To maintain both coverage and connectivity with $R_c = R_s$
Reference 20	Considers mobility to achieve *k*-coverage	Connectivity is not considered during coverage preservation	To maintain coverage and connectivity, preserving existing coverage and connectivity
Reference 21	Considers $R_c \geqslant 2R_s$ to maintain coverage, considers mobility based on residual energy	Larger communication range, Longer mobility distance, cannot maintain network due to unpredictable death of a node	To maintain coverage and connectivity with smaller communication range, shorter mobility distance and due to death of a node
Reference 23	Considers $R_c = 2R_s$ to maintain coverage, mobility is limited within one hop	Larger communication range, Does not consider connectivity maintenance, cannot maintain network due to unpredictable death of a node	To maintain coverage and connectivity with smaller communication range, shorter mobility distance and due to death of a node

solved problems for mobility assisted coverage and connectivity maintenance are summarized in Table 1.

2.2. Motivations

From the review of several literature, it is observed that some papers design coverage maintenance algorithms, whereas few others propose connectivity maintenance issues of WSN. To the best of our knowledge, none of the work considers the limited mobility model to maintain both coverage and connectivity, simultaneously. Besides, most of the works consider that communication range is greater than or equal to twice of the sensing range. Since, communication is the main consumer of energy resource [25], the most different assumption in our work is that communication range is equal to sensing range ($R_c = R_s$) and mobility of the nodes is limited within only one-hop. Though, short communication range on the other hand means more nodes should be turned onto maintain the connectivity, it is not suitable for the wireless sensor networks as nodes are deployed densely and more nodes should be turned onto guarantee the quality of surveillance. Since, more nodes should be turned on and are deployed densely; we feel that larger communication range is not necessary as it consumes more energy. Moreover, our algorithms can also be extended to maintain the connectivity and coverage of the network for $R_c < 2R_s$. The distinguishing feature of our work is that the possible mobility distance of the nodes is determined pro-actively, which can work to maintain the network either for predictable or unpredictable death of a node. Since, sensors are deployed in the dense forest or harsh terrains and redeployment of nodes may be hard or impossible, our algorithms can maintain the network time to time remotely and dynamically, which could be accomplished by the limited mobility of the nodes.

3. System model

We consider the wireless sensor networks, where nodes are distributed randomly and densely with higher degree of neighbors over a squared monitoring region. Each node is aware of its location through location services [26], is capable of moving [3–6] and is equipped with digital compass [27] to find the direction. Each node has a unique ID to be distinguished from others.

3.1. Assumptions and definitions

Assumption 1 (*Sensing and communication range*). Each node is equipped with homogeneous sensing and communication devices. The ratio of communication to sensing range is $1 \leqslant R_c/R_s < 2$. However, our protocols could be implemented either for $R_c = R_s$ or for $R_c < 2R_s$ at a time.

Assumption 2 (*Initial coverage and connectivity*). Initially, the whole network is fully connected and the whole monitoring region is fully covered. Our algorithms are proposed to maintain the network, if any node is dead after deployment.

Assumption 3 (*Death of node*). An unpredictable death of a node may happen due to intrusion, explosion or environmental factors like heat, vibration and failure of electronic components or software bugs in the network. A predictable death of a node may happen due to its energy exhaustion. Each node knows its initial energy level and can keep track of its energy expenditure so that it can predict its own death in advance. A node is assumed to be dead if it fails to either sense or communicate.

Definition 1 (*Communication disc*). The communication disc of a node is its communication range of radius R_c, which is centered at the location of an active sensor. As shown in Figs. 1(a) and (b), the dotted circle represents the communication disc of nodes. Throughout the paper, radius of the communication disc is refereed to as communication range (R_c). **For any pair of nodes** A **and** B**, Euclidean distance between** A **and** B **is denoted as** $d(A,B)$**, where** $R_c \leqslant d(A,B)$**.**

Definition 2 (*Sensing disc*). The sensing disc of a node is its sensing range of radius R_s, which is centered at the location of an active sensor. Any object present within the sensing disc of a node can be detected by it perfectly. Throughout the paper, radius of the sensing disc is refereed to as sensing range (R_s). As shown in Figs. 1(a) and (b), the circles with shaded region represent the sensing disc.

Definition 3 (*Connecting neighbors*). Two nodes A and B are said to be connecting neighbors, if their Euclidean distance $d(A,B) \leqslant R_c$. As per our assumptions, it is shown in Fig. 1(a) that $R_c = R_s$, whereas, $R_c < 2R_s$ as shown in Fig. 1(b). Hence, nodes A and B are connecting neighbors in Fig. 1(a), as $d(A,B) = R_c$, where $R_c = R_s$. Similarly, node A and B are connecting neighbors in Fig. 1(b), as $d(A,B) < R_c$, where $R_c < 2R_s$. Besides, as shown in Fig. 1(b), nodes C and D are connecting neighbors of node B, as they are located within communication range of B, which is $<2R_s$. From this definition, it is clear that a connecting neighbor is always one-hop neighbor of a node.

Definition 4 (*Sensing neighbors*). Two nodes A and B are said to be sensing neighbors, if their Euclidean distance $d(A,B) \leqslant 2R_s$. As shown in Fig. 1(a), nodes A and B are sensing neighbors as $d(A,B)$ is $=R_s$. Similarly, as shown in Fig. 1(b), nodes B or C is a sensing neighbor of A, as $d(A,B)$ is $<2R_s$ and $d(A,C)$ is $=2R_s$. In order to relate sensing neighbors with connecting neighbors, it is to be noted that normally sensing neighbors are located within one or two-hops from each other. As shown in Fig. 1(b), node B and C are sensing neighbors of node A, whereas node B is the only connecting neighbor of node A. Node A and B are one hop away, whereas node A and C are two hops away from each other as $d(A,C) = 2R_s$. In another situation, nodes C and D are sensing as well as connecting neighbors of B and are one-hop away from B.

3.2. Problem formulation

In wireless sensor networks, nodes are deployed densely and it is very complex to monitor the mobility of the nodes. In our protocol, if a node is predictably or unpredictably dead, its one hop neighbors have to move to maintain the coverage and connectivity dynamically. It is to be noted that in our protocol, the mobility of the nodes is limited within one hop without losing their existing coverage and connectivity. In order to guarantee that the existing coverage of a node is not lost due to its mobility, we formulate the*critical sensing point (CSP)* set of each node. Besides, in order to guarantee that the existing connectivity is not lost due to mobility of a node, we develop the *disjoint transmission set (DTS)* algorithm for each node. The detail description of each concept is given as follows.

3.2.1. Critical sensing points (CSP) of a node

The critical sensing points (CSP) of a node are the points of intersection of that node with sensing discs of its sensing neighbors, which satisfy the conditions as follows.

Let, $N_s(A)$ be the set of sensing neighbors of node A such that $|N_s(A)| = l$ and a be the location of node A, and k_i be the location of K_i-th node, where $K_i \in N_s(A)$, $\forall i = 1,2,\ldots,l$ and $d(ak_i) \leqslant 2R_s$. If S is the set of points of intersection, formed by nodes K_m and K_n with sensing disc of A, where K_m, $K_n \in N_s(A)$ for $m \neq n$, a point $p \in S$ is said to be a critical sensing point (CSP) of node A, if p satisfies both of the following conditions:

1. $d(pa) \leqslant R_s$
2. $\forall K_i \in N_s(A)$, where $i = 1,2,\ldots,l, d(pk_i) \geqslant R_s$.

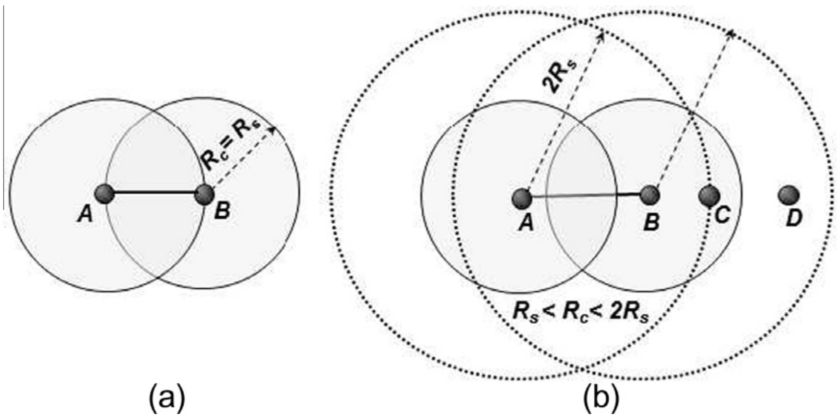

Fig. 1. Example of communication and sensing range and connecting and sensing neighbors.

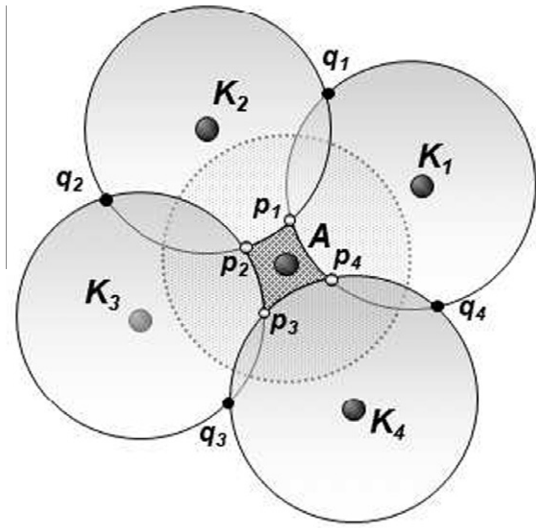

Fig. 2. The points p_1, p_2, p_3 and p_4, which enclose the one-cover shaded region are the critical sensing points of *A*.

For example, as shown in Fig. 2, p_1, p_2, p_3 and p_4 are *CSP* of *A* and the shaded region enclosed by these points and formed by node *A* with nodes K_1, K_2, K_3 and K_4 is always one-cover. The points q_1, q_2, q_3 and q_4 are not *CSP*, as they cannot satisfy the first condition. It is to be noted that upon receiving the flooding message, each node keeps information of its two hop neighbors and then finds its critical sensing points based on the above rule.

3.2.2. Disjoint transmission set (DTS) of a node

It is to be noted that for $R_c = R_s$ or $R_c < 2R_s$, distance between any two connecting neighbors must be $\leqslant R_c$. A connecting neighbor *X* of any node *A* is said to be member of an *i*th *DTS* D_i, $\forall i = 1,2,3,\ldots$, if any one of the following conditions holds.

(1) If *X* has a common connecting neighbor with other members of that D_i.
(2) If neighbor of *X* is also a connecting neighbor of *A*.
(3) If *X* is the single neighbor of *A* without having any common connecting neighbor.

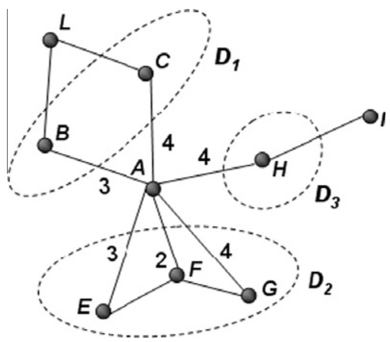

Fig. 3. An example of disjoint transmission set of node *A*.

As shown in Fig. 3, let us find the *DTS* of node *A*. Since, node *B* and *C* have the connecting neighbor *L*, which is two-hops away from *A*, only *B* and *C* belong to *DTS* D_1. In another scenario, node *E* and *G* are connecting neighbors of *A*. Though node *F* is common neighbor of node *E* and *G*, it is also a connecting neighbor of *A*. Hence, node *E*, *F* and *G* belong to the same *DTS* D_2. Node *H* is a connecting neighbor of node *A*, without having any common neighbor. Hence, it belongs to *DTS* D_3. It is to be noted that each node of the network classifies its connecting neighbors into member of the *DTS*. The algorithm for calculating the disjoint transmission set is given in Table 2.

3.2.3. Head node of a DTS

Let, $D_i(A)$ be the disjoint transmission set (*DTS*) of node *A*, for $i = 1,2,\ldots,n$. Any node, $N \in D_i(A)$ is said to be a *head node* in its *DTS*, if *N* is closest to *A*. For example, as shown in Fig. 3, the number along each link represents the Euclidean distance between two nodes. Since, $B, C \in D_1(A)$, and node *B* is closest to both *A*, *B*, hence *B* is the *head node* in D_1. Similarly, node *F* and node *H* are the *head nodes* in D_2 and D_3, respectively.

Table 2
Algorithm to classify the disjoint transmission sets.

Algorithm 1: *Disjoint transmission set (DTS) classification*

Notation:
1. A_{Nbr}: List of neighbors of node *A*;
2. D_i: *i*th disjoint transmission set;
3. *unclassified_nbr*: Set of neighbors who have not been classified as the members of Disjoint Transmission Set;
4. **Disjoint Transmission Set**(A_{Nbr})
5. $E := A_{Nbr}$; /* **Assigns all neighbors of** *A* **to** *E* */
DTS Classification( )
1. Initialize *i*;
2. **while:** *unclassified_nbr* $\neq \Phi$
/* **Checks here if the unclassified neighbor set is NULL or not** */
3. **do:** {
4. Select a node *E* from *unclassified_nbr*;
5. *unclassified_nbr* := *unclassified_nbr* − *E*;
/* **Updates the unclassified neighbor set** */
6. $D_i := D_i \cup E$; /* **Updates the** *i*th **disjoint transmission set** */
7. **for** each node $M \in$ *unclassified_nbr*
/* **Considers a node** *M* **from the set of unclassified neighbors** */
8. **if** $((M \cap E_{Nbr} \neq \Phi) \| (M_{Nbr} \cap E_{Nbr} \neq \Phi))$
/* E_{Nbr} **and** M_{Nbr} **are the set of neighbors of node** *E* **and** *M*, **respectively** */
9. $D_i := D_i \cup M$;
/* **Updates the** *i*th **disjoint transmission set** */
10. *unclassified_nbr* := *unclassified_nbr* − *M*;
/* **Updates the unclassified neighbor set** */
11. **End of for loop**
12. **for** each node $N \in$ *unclassified_nbr*
13. **for** each node $M \in$ *unclassified_nbr*
14. **if** $((M \cap N_{Nbr} \neq \Phi)$ OR $(M_{Nbr} \cap N_{Nbr} \neq \Phi))$
/* **Verifies intersection of node** *M* **or neighbors of node** *M* **with neighbors of node** *N* */
15. $D_i := D_i \cup M$;
/* **Updates the** *i*th **disjoint transmission set** */
16. *unclassified_nbr*: = *unclassified_nbr* − *M*;
/* **Updates the unclassified neighbors set** */
17. **end if**
18. **End of for loop**
19. **End of for loop**
20. Increment *i*;
21. }
22. **End of while loop**

3.2.4. Beacon packet transmission

The packet, which is broadcast among the connecting neighbors of each node and contains the sender's ID, location information and critical sensing points, one-hop neighbor's list and list of head nodes of the sender is known as *beacon packet*. The one-hop neighbor's list includes the neighbor's ID, location, whereas the list of head nodes contains the list of head node's location information and available distance for mobility.

4. Connectivity and coverage maintenance protocols

It is to be noted that our coverage connectivity maintenance (*CoCo*) protocols are used to maintain the network in the post deployment scenarios. However, in order to maintain the network due to unpredictable death of a node, each node should be proactive as follows. As soon as the nodes are deployed over the monitoring region, each node starts calculating its grid id from its location information. Then, nodes flood beacon packets with their location information and grid id to estimate the critical sensing points (CSP), disjoint transmission set (DTS) and selects the head nodes in each of its *DTS*, as per the definitions given in Section 3. If a node in the network is dead due to its power exhaustion (predictable death) or accidentally dead due to explosion or technical failure (unpredictable death), connectivity with its neighbors is lost and the existing coverage of the network may be deteriorated, unless it is a redundant node. Hence, the network should be maintained with some prior arrangements, taking its *Available Distance*(AVD) of mobility, so that nodes can move immediately as soon as such problem is occurred. For this, we assume that beacon packets are exchanged periodically among the one-hop neighbors of each node to know the location information of the corresponding nodes. Then, each node starts calculating the *AVD* in terms of *Maximum Transmission Mobility Distance* (MTMD) and *Maximum Sensing Mobility Distance* (MSMD), as described in the following subsections.

4.1. Available Distance (AVD)

The Available Distance (*AVD*) of mobility of a node is defined as the maximum distance that a node can move without affecting the existing coverage and connectivity and is calculated as the minimum value between the *MTMD* and *MSMD*. Prior to calculating the *AVD*, each node forwards the beacon packet to its connecting neighbors and each receiver starts calculating *MTMD* and *MSMD* without including the sender in the calculation procedure. Since, our algorithms also consider the network maintenance under unpredictable death of a node, each receiver assumes the sender as a dead node to calculate its *AVD*, so that it can execute the coverage and connectivity maintenance algorithms in case of unpredictable death of that sender.

4.1.1. Calculation of MTMD

The *Maximum Transmission Mobility Distance (MTMD)* of a node is defined as the maximum distance that a node can move from its current position to the position of the sender, which is assumed to be a dead node, such that connectivity is not broken between itself and with all of its *head nodes* of its *DTS*s. Upon receiving the beacon packet, a node (receiver) reclassifies its *head node* list and *DTS*s without considering the sender. It is obvious that the distance between each head node and the receiver must be $\leqslant R_c$, since they are connecting neighbors. Then, the receiver node finds points on the line joining the sender to the receiver such that the point is R_c units away from each of its head nodes. The distance between the nearest point to the receiver is termed as *MTMD* of that receiver.

As shown in Fig. 4(a), node *A* has two different *DTSs* D_1 and D_2, and node *P* has four different *DTSs* D_3, D_4, D_5 and D_6. When node *P* receives the beacon packet from *A*, it assumes *A* as the dead node for future use and reclassifies its current *DTS*s into three new *DTSs* without taking *A*, as shown in Fig. 4(b). To estimate *MTMD*, ω_1, ω_2 and ω_3 are taken on the line $\overline{PA}$ from the head nodes *J*, *G* and *R*, respectively, such that $J\omega_1 = G\omega_2 = R\omega_3 = R_c$. The point ω_3 is considered

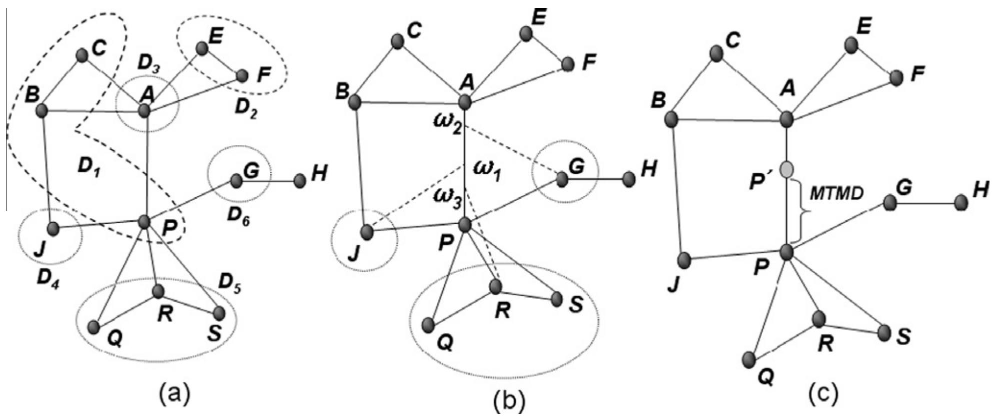

Fig. 4. (a) D_1 and D_2 are DTSs of node *A* and are represented by the dotted curves. D_3, D_4, D_5 and D_6 are the DTSs of *P* and are represented by the solid curves. (b) Taking *A* as a sender and *P* as a receiver, shortest distance from each head nodes of *P* is estimated without taking sender *A*. (c) *PP′* is the required *MTMD* of *P*. If *A* is dead, each of its head nodes can be connected with it. *P′* is the new position of *P* after mobility.

Table 3
Algorithm to calculate the MTMD of a node.

Algorithm 2: *MTMD Calculation*

Notation:
1. S: Sender node;
2. R: Receiver node that is going to calculate *MTMD*;
3. R_{Nbr}: Set of neighbors of receiver node R;
4. X: Set of disjoint transmission sets (DTS) of R;
5. D_i: i-th DTS of receiver R;
6. N_H: Head node;
7. $d(A,B)$: Euclidean distance between any two nodes A and B;
8. R_c: Communication range of any node;

MTMD Calculation (Node R)
1. $MTMD := \infty$;/* **MTMD is initialized to** ∞*/
2. for each set, $D_i \in X$/* **Considers** ith **DTS from all DTS of** R */
3.   for each node, $N_H \in D_i$/* **Considers head node of** ith **DTS***/
4.   Find a point,ω on $\overline{SR}$ such that $d(N_H,\omega) = R_c$;
5.   if $(d(R,\omega) < MTMD)$
6. $MTMD := d(R,\omega)$;

/* **Updates MTMD as the distance between receiver node** R **and** ω*/

7.   end if
8.   end of for loop
9. end of for loop
10. return *MTMD*;

Table 4
Algorithm to calculate the MSMD of a node.

Algorithm 3: *MSMD calculation*

Notation:
1. S: Sender node;
2. R: Receiver node that is going to calculate *MSMD*;
3. R_{CSP}: Set of critical sensing points of node R;
4. $d(A,B)$: Euclidean distance between nodes A and B;
5. R_s: Sensing range of any node;

MSMD Calculation(Node R)
1. $MSMD := \infty$; /* MSMD is initialized to ∞*/
2. for each critical sensing point $p \in R_{CSP}$
3.   Find a point γ on $\overline{SR}$, such that $d(p,\gamma) = R_s$;
4.   if $(d(A,\gamma) < MSMD)$

/* **Compares MSMD with the distance between a node** A **and the point** γ*/

5. $MSMD := d(A,\gamma)$;

/* **Updates MSMD as the distance between node** A **and** γ*/

6.   end if
7. end of for loop
8. return *MSMD*;

as the new mobility position of node P to move towards A, since it is min $(P\omega_1, P\omega_2, P\omega_3)$. Hence, *MTMD* of node P towards node A is estimated as $d(P\omega_3)$, as shown in Fig. 4(c), which is the maximum distance that P can move so that the links between nodes J, R and G are not broken. The algorithm for calculating the *MTMD* is given in Table 3.

4.1.2. Calculation of MSMD

The *Maximum Sensing Mobility Distance (MSMD)* of a node is defined as the maximum distance that a node can move without introducing any coverage problem in the existing network. *MSMD* is estimated by a node that wants to move due to accidental death of its neighbors and ensures that the existing coverage is still preserved and is not affected due to its mobility.

As shown in Fig. 5(a), the shaded region is one-covered and death of node P creates the coverage hole. When P receives the beacon packet from its connecting neighbor A, it estimates its maximum sensing mobility distance towards A, as if A is dead. From each of its CSP, p_1, p_2, p_3 and p_4, it finds points γ_1, γ_2, γ_3 and γ_4, respectively on the line $\overline{PA}$ such that $p_1\gamma_1$, $p_2\gamma_2$, $p_3\gamma_3$ and $p_4\gamma_4$ are equivalent to R_s. Then, $\min(P\gamma_1, P\gamma_2, P\gamma_3, P\gamma_4)$ is considered as the *Maximum Sensing Mobility Distance (MSMD)* of P towards A. As shown in Fig. 5(b), PP' represents the *MSMD* of node P. The algorithm for estimating the *MSMD* is given in Table 4.

4.2. The maintenance algorithms

As mentioned earlier, beacon packets are exchanged periodically among the connecting neighbors of each node. If a node is going to die due to its energy exhaustion, it immediately predicts about its death to its connecting neighbors using the beacon packets. Upon receiving the beacon packet, the receiver node verifies if any connectivity or coverage problem may arise due to its death. If a node does not receive any beacon packet from any of its

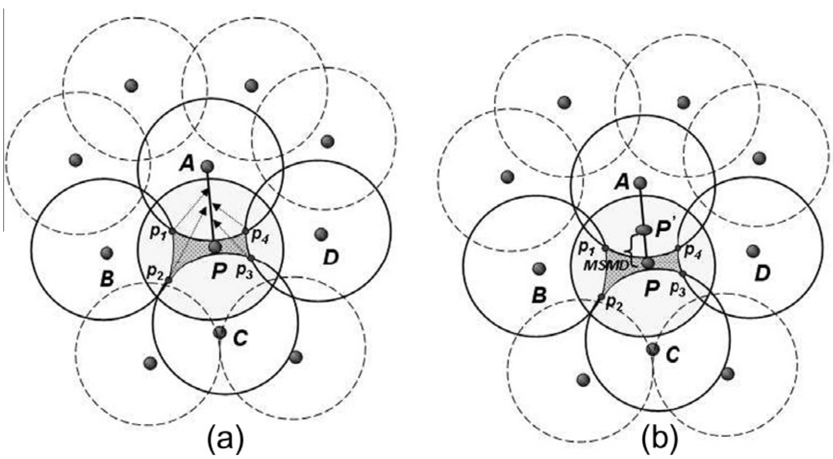

Fig. 5. (a) Here, the shaded region is one-covered and death of node P creates the coverage problem. Hence, P estimates its *MSMD* with respect to its *CSPs*. (b) P estimates its *MSMD* as PP' so that its mobility from P to P' does not affect the existing coverage.

connecting neighbors consecutively for a specific number of times either due to death of its neighbor or due to communication interferences (In our simulation, we have taken two times as the limitation, which can have different value as defined by the user), it assumes the unpredictable death of that connecting neighbor. Then, it executes the connectivity and coverage maintenance (CoCo) algorithms to maintain the network as given below. It is to be noted that each node keeps the location information of its original position. In case of a node incorrectly determines its connecting neighbor is dead (may be due to communication interference or failure) and moves to the location of the dead node to maintain the network, it can move back to its original location if it finds that its neighbor is alive. The detail procedures of our network maintenance algorithms are described as follows.

4.2.1. Connectivity maintenance

It is to be noted that the beacon packets are exchanged among the connecting neighbors of a node periodically and each node preserves the information in the beacon packet until it receives the next one. In case of predictable or unpredictable death of the connecting neighbors, a node scans information contained in the last beacon packet of that dead node. From the information of the beacon packet, if a connecting neighbor does not declare any critical sensing point, instead it declares more than one disjoint transmission set, it implies that it will create communication problem among its neighbors. Hence, the main goal of our connectivity maintenance protocol is to reduce the distance among the head nodes of different *DTS* so that the network can be reconnected.

A. Minimum Spanning Tree (MST) Procedure: We use the minimum spanning tree procedure to minimize the distance among the head nodes of different *DTS*, as described below.

Step 1: **Form a weighted graph taking** *head nodes* of all *DTS* of a dead node as vertices and Euclidean distance between two consecutive head nodes as weight of each edge.

Step 2: Use Kruskal's algorithm [28] to construct the minimum spanning tree, taking dead node as the root.

Step 3: Each head node, representing the vertex of the graph should move towards the root such that all edges of the minimum spanning tree are reduced to the communication range (R_c). The *Required Mobility Distance* (RMD) for any *head node P* with respect to another *head node S* and vice versa is defined as: $R_c^2 = (\overline{PD} - d)^2 + (\overline{SD} - d)^2 - 2(\overline{PD} - d)(\overline{SD} - d)cos\beta$, where, d is *RMD* of the head node, β is angle between node *P* and *S* with respect to dead node *D*. It is to be noted that one head node can get location information of another one from the last beacon packet sent by the dead node, though the head nodes are not connecting neighbors. A node may calculate several *RMD*s, but chooses the longest one as its final *RMD*. It is to be noted that the direction of mobility of each node can be calculated using Eq. 1.

(x_i, y_i): Location of the connecting neighbor of a dead node.

(x_j, y_j): Location of the dead node.

$$\theta = \begin{cases} \tan^{-1}\frac{\Delta y}{\Delta x}, & x_j > x_i,\ y_j \geqslant y_i, \\ \tan^{-1}\frac{\Delta x}{\Delta y} + \frac{\pi}{2}, & x_j \leqslant x_i,\ y_j > y_i, \\ \tan^{-1}\frac{\Delta y}{\Delta x} + \pi, & x_j < x_i,\ y_j \leqslant y_i, \\ \tan^{-1}\frac{\Delta x}{\Delta y} + \frac{3\pi}{2}, & x_j \geqslant x_i,\ y_j < y_i, \end{cases} \tag{1}$$

where $\Delta x = |x_j - x_i|$, $\Delta y = |y_j - y_i|$.

An example of the connectivity maintenance based on the Kruskal's algorithm [28] is given in Fig. 6. As shown in Fig. 6(a), let *P*, *Q*, *R* and *S* are head nodes of the dead node *D*, who form a complete graph with Euclidean distance among each pair of head nodes as the weight of each edge of the graph. The head nodes construct a minimum spanning tree, as shown in Fig. 6(b). When node *D* is dead, the head nodes of all *DTSs* of *D* move towards *D* such that the edges of the minimum spanning tree is reduced to R_c, as shown in Fig. 6(c). Assuming the connectivity among head nodes is regained after mobility, the *Required Mobility Distance (RMD)* is calculated and each head node has to move up to the distance equivalent to the value of the *RMD*. As discussed earlier, since, *Available Distance (AVD)* of mobility for each neighbor towards the dead node is known and each head node calculates other node's *RMD*, it can know if its neighbor who shares same edge with it has enough *AVD* or not. If all neighbors of the dead node have sufficient *AVD*, the head nodes move towards the dead node for the required distance and maintain the connectivity as shown in Fig. 6(c), otherwise the head node goes for executing the *Cascading Mobility Procedure*, as follows. The algorithm for calculating *RMD* of a node for connectivity maintenance is given in Table 5.

B. Cascading Mobility Procedure:

As discussed in the previous subsection, a *head node* moves to compensate the connectivity of the dead node and also needs itself to be connected with its neighboring head nodes using *MST Procedure*. However, if its *AVD* = 0, it gives up the *MST Procedure* and turns to *Cascading Mobility Procedure*. In this procedure, each head node resets its *RMD* as the distance between itself and location of the dead node. Out of all *head nodes*, a *leader node* is selected, which leads this procedure. A head node having *min*(*RMD* − *AVD*) is selected as a *leader node*. If multiple head nodes are having same value of (*RMD* − *AVD*), head node with least value of *RMD* is selected as the leader node. If value of (*RMD* − *AVD*) < 0 for a leader node, it moves to the position of the dead node without bothering any other nodes, else it sends a mobility request (*MOB_REQ*) packet to head nodes of the *leader node* in each of its disjoint transmission sets and to its sensing neighbors, which contains mobility direction and *RMD* of the *leader node*. It is to be noted that each node in the network has its own *DTS*, *CSP* and *head node sets*. The sensing and connecting neighbors of the *leader node*, in each of its disjoint transmission sets receive the *MOB_REQ* packet and move to compensate the connectivity loss.

For example, as shown in Fig. 7(a), *leader node L* has to lead the cascading mobility. Nodes *B*, *C*, *E*, *F* and *G* are

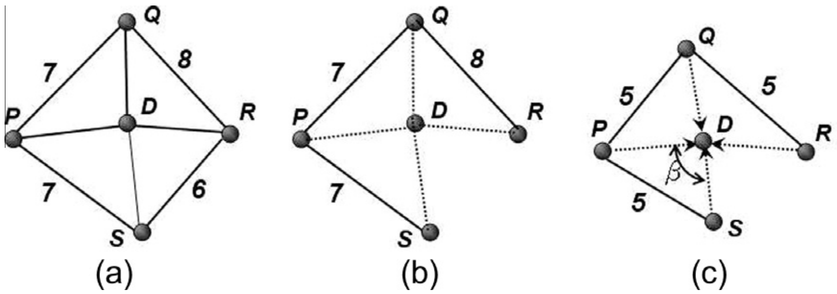

Fig. 6. (a) Head nodes form a weighted graph with dead node *D*. The wights represent the physical distance between any two nodes. (b) Head nodes *P*, *Q*, *R* and *S* construct the minimum spanning tree. (c) Head nodes *P*, *Q*, *R* and *S* move towards the dead node *D* such that the edges of minimum spanning tree is reduced to R_c, where R_c is 5 units taken in the example.

connecting neighbors of *L*, where *B* and *C* belong to one of its *DTS* with *C* as their head node and *E*, *F*, *G* are members of another *DTS* of *L*, with *F* as their head node. Since, nodes *C* and *F* are connecting neighbors of *leader node L*, all of them can receive the *MOB_REQ* packet forwarded by *L* and calculate their mobility direction and magnitude. As shown in Fig. 7(b), head nodes *F* and *C* move parallel to the mobility direction of *L*. Let, *RMD* of nodes *F* and *C* be *RMD*(*F*) and *RMD*(*C*), respectively. Node *F* moves *RMD*(*F*) units to *F*′ so that the distance from the new position of node *F* to the new position of leader node *L* is R_c units. Similarly, node *C* moves *RMD*(*C*) units to *C*′, so that the distance between *C*′ and *L* is R_c units.

Table 5
Algorithm to calculate the RMD of a node for connectivity maintenance.

Algorithm 4: *RMD Calculation for Connectivity*

Notation:
1. *AVD*(*i*): Available Distance of head node *i* to move;
2. *RMD*(*i*): Required Mobility Distance of head node *i*;
3. *D*: Dead node;
4. *d*(*i*,*j*): Euclidean distance between node *i* and *j*;
5. β(*i*,*j*,*D*): Angle between head node *i* and head node *j* with respect to the dead node *D*;
6. R_c: Communication range of each head node;
7. *d*: RMD of head node *i* with respect to head node *j*;

RMD Calculation(*i*,*j*,*D*)

```
for each head node i of dead node D;
    for each head node j having an edge with head node i;
        find d such that R_c^2 = (d(i,D) - d)^2 + (d(j,D) - d)^2
            -2(Dist(i,D) - d)(Dist(j,D) - d) cosβ(i,j,D);
/* Calculates the required mobility distance d*/
        if ((d ⩽ AVD(i)) and (d ⩽ AVD(j)))
/* Verifies if the required mobility distance (d) of node i or node j is
⩽ available mobility distance of i and j, respectively*/
            RMD(i) = d;
/* Required mobility distance of node i is set to be d */
            RMD(j) = d;
/* Required mobility distance of node j is set to be d */
        else if ((d ⩾ AVD(i)) and (d ⩽ AVD(j)))
            RMD(i) := AVD(i) and head node i executes
    Cascading Mobility;
/* Required mobility distance of node i is set to be available
    mobility distance of node i */
            RMD(j) := d;
/* Required mobility distance of node j is set to be d */
        else if ((d ⩽ AVD(i)) and (d ⩾ AVD(j)))
            RMD(i) := d;
/* Required mobility distance of node i is set to be d */
            RMD(j) := AVD(j) and head node j executes
    Cascading Mobility;
/* Required mobility distance of node j is set to be available
    mobility distance of node j */
        else if ((d ⩾ AVD(i)) and (d ⩾ AVD(j)))
            RMD(i) := AVD(i);
/* Required mobility distance of node i is set to be available
    mobility distance of node i*/
            RMD(j) := AVD(j);
/* Required mobility distance of node j is set to be available
    mobility distance of node j*/
            Both head nodes i and j execute Cascading
    Mobility;
    end of for loop
end of for loop
```

In order to explain the coverage maintenance in cascading mobility procedure, we display an example, as shown in Fig. 8(a) and (b). As shown in Fig. 8(a), node *P*, *Q*, *R* and *D* are sensing neighbors of *L*. If *L* moves towards the dead node *D*, the region enclosed by points p_1, p_2, p_3 and p_4 remains uncovered. Hence, goal of cascading mobility is how to decide the magnitude and direction of the mobility of the head nodes of a leader node *L*, if it moves towards the dead node, so that the connectivity and coverage can still be maintained.

As shown in Fig. 8(b), since node *P*, *Q* and *R* have to move parallel to the mobility direction of leader node *L*, consider a line starting from p_1, whose slope is same as the mobility direction of *L*. Then, find a point on the line such that the distance from that point to the destination of node *L* should be equal to R_s. Hence, the distance from p_1 to the new point is termed as the *RMD* for *P*, which is q_1 units, as shown in Fig. 8(b). Similarly, q_2 is calculated, taking point p_2 and $max(q_1,q_2)$ is the *RMD* for node *P*. This procedure is repeated for each related nodes to calculate the *RMD* for *Q* and *R*, estimating q_2, q_3 and q_4 and for finding maximum value among them. It is to be noted that a node calculates its *AVD* based on the value of *RMD* and mobility direction given in the *MOB_REQ* packet and sends back a *MOB_RPY* packet to the leader node, indicating its *AVD*. The node has to move, if it has sufficient *AVD* i.e. more than the value of *RMD*, otherwise, it sends the *MOB_RPY* packet with zero *AVD*. In this case, redeployment of new nodes in the network is required, as connectivity problem cannot be solved by any means.

4.3. Coverage maintenance

It is to be noted that beacon packets are exchanged among the neighbors periodically. If any node of the

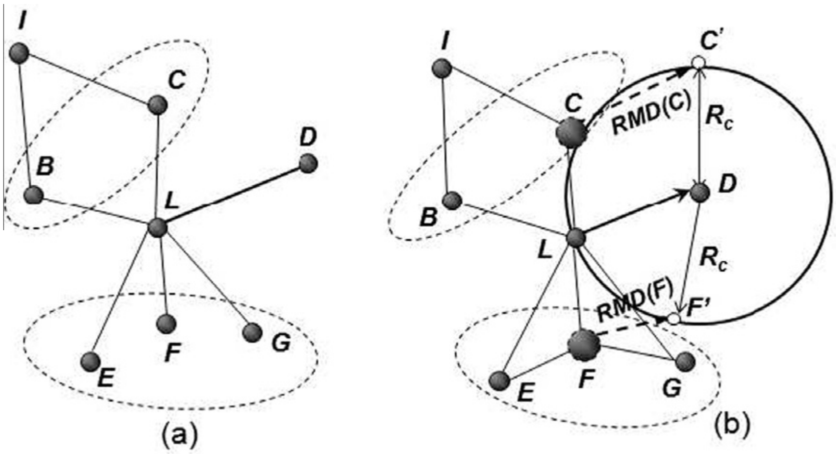

Fig. 7. (a) Disjoint Transmission Sets of *leader node L*, among its connecting neighbors. *L* has to move to the position of the dead node *D*. (b) Node *F* and *C* are head nodes in different *DTS* of leader node *L*. Hence, both *F* and *C* move parallel to the mobility direction of *L* with *RMD*(*F*) and *RMD*(*C*) units, respectively, so that the connectivity between *F* and *L* and *C* and *L* is not lost.

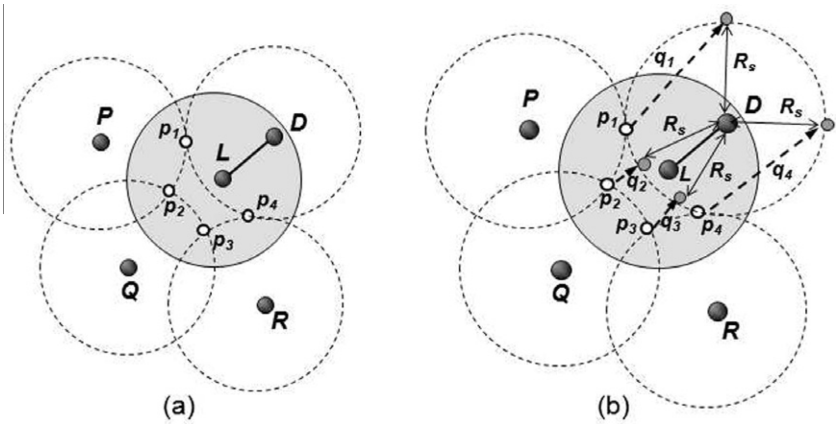

Fig. 8. (a) Critical sensing points formed by sensing neighbors of *leader node L*. The region enclosed by CSPs p_1, p_2, p_3 and p_4 should not remain uncovered, if *L* moves to the position of the dead node *D*. (b) *P*, *Q* and *R* are the sensing neighbors of *L*, which move parallel to the mobility direction of *L*, such that the critical sensing points p_1, p_2, p_3 and p_4 are displaced q_1, q_2, q_3 and q_4 units, respectively so that no coverage hole is created.

network is dead accidentally and unable to exchange the beacon packets or is going to die, the information received in the last beacon of the node is scanned. If that beacon declares some critical sensing points, whereas the node has only one disjoint transmission set, it implies that coverage hole could be created due to death of that node. In this case, the head nodes of the dead node execute *Coverage Maintenance Algorithm*, as given in Table 6 to estimate the required mobility distance.

It is to be noted that each *head nodes* of the dead node executes the coverage maintenance algorithm and estimates the *Required Mobility Distance (RMD)* and compares with its *Available Distance (AVD)* of mobility. As per the algorithm, the node having enough *AVD* has to move towards the location of the dead node. If more than one node has enough *AVD*, node having least value of *RMD* has to move. If all of the head nodes do not have enough *AVD*, node having least value of *(RMD-AVD)* becomes a *leader node*, which executes the *Cascading Mobility Procedure*. An example of coverage maintenance algorithm is shown in Fig. 9. Let, p_1, p_2, p_3, p_4 and p_5 are the *critical sensing points (CSP)* of node *D* created with neighboring nodes *A*, *B*, *C*, *E* and *F*. If node *D* is dead, its head nodes *A* and *B* have to estimate their *RMD*. As shown in Fig. 9, find a point γ on the line joining *B* through *D*, such that the farthest *CSP* of *D* is equal to sensing range R_s. Thus, $\overline{B\gamma}$ is the *RMD* for node *A*. After estimating *RMD* for each head nodes of *D*, the *AVD* of each head nodes are compared with the respective *RMD*s and node having enough *AVD* has to move towards the dead node *D*.

4.4. Theoretical analysis

It is to be noted that in our protocol, the head nodes, which are one hop neighbors of a dead node should move toward the location of the dead node to maintain the lost coverage and connectivity. Accordingly, the head node first verifies its Available Distance (AVD) of mobility and the required mobility distance (RMD). Let, a_{ij} and r_{ij} be the AVD and RMD of a head node, respectively, where *i* and *j* are

Table 6
Algorithm to calculate the RMD of a node for coverage maintenance.

Algorithm 5: *Coverage Maintenance*

Notation:
1. $AVD(i)$: Available Mobility Distance of node i;
2. $RMD(i)$: Required Mobility Distance of node i;
3. i_{CSP}: Critical sensing point of node i;
4. $d(A,B)$: Distance between node A and node B;
5. *Min_RMD*: Minimum RMD value among all head nodes;
6. *Coverage_Solver*: Node that goes to solve coverage problem;
7. *Farthest_Distance*: Farthest distance from all CSP;

Coverage RMD Calculation ()
1. *Min_RMD* := ∞;
/* **Minimum required mobility distance of a node is set to be** ∞ */
2. *Coverage_Solver* := Φ;
/* **Node that goes to solve coverage problem is initialized to** Φ */
3. for each head node i of dead node D
4. *Farthest_Distance* := 0; /* **Farthest distance is initialized to** 0 */
for each critical point p in i_{CSP}, find a point γ, on $\overline{iQ}$ such that $d(p,\gamma) = R_s$;
5. if ($d(i,\gamma)$ > *Farthest_Distance*)
6. *Farthest_Distance* := $d(i,\gamma)$; /*Farthest distance is updated */
7. end of *for* loop
8. $RMD(i)$:= *Farthesr_Distance*; /* **Farthest distance is assigned to RMD** */
9. if ($RMD(i) \leqslant AVD(i)$ and $RMD(i)$ < *Min_RMD*)
10. *Coverage_Solver* := i; /* **Node** i **is set to solve the coverage problem** */
11. *Min_RMD* := $RMD(i)$;
/* **RMD of node** i **is set to be the Minimum RMD value for all head nodes** */
12. end of *for* loop
13. if (*Coverage_Solver* $\neq \Phi$)
14. if (*Coverage_Solver*(i) = i
/* **If the node that is set to solve the coverage problem is a head node itself** */
15. *RMD* := *Min_RMD*;
/* **RMD of that node is set to be Minimum RMD value among all head nodes** */
16. else
17. *RMD* := 0;
/* **RMD of that node is set to be 0** */
18. else
19. Go for *Cascading Mobility Procedure*;

the dead node and head node (one-hop neighbor of a dead node), respectively. Taking (x_i,y_i) and (x_j,y_j) as the location of a dead node i and its head node j, respectively, the required mobility distance (r_{ij}) of a head node could be calculated as

$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (2)$$

From the coverage and connectivity maintenance algorithm, the available distance (a_{ij}) of mobility could be calculated as

$$a_{ij} = min\{mt_j, ms_j\}, \quad (3)$$

where, mt_j and ms_j are the MTMD and MSMD of the head node j, respectively. Let a dead node has n head nodes and coverage probability of each head node be p, which can be derived as

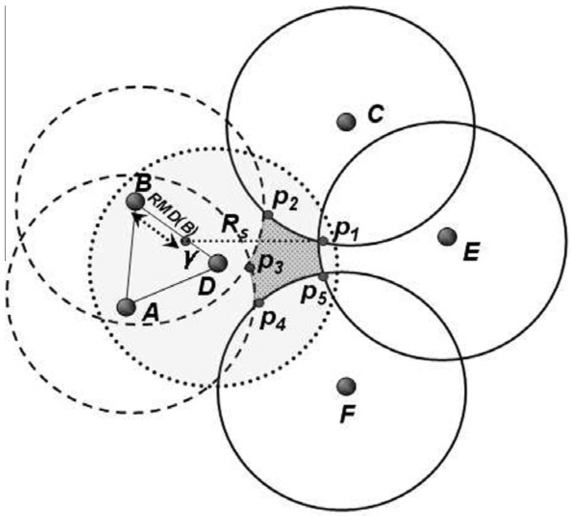

Fig. 9. p_1 is the farthest critical sensing point of node D. In order to estimate *RMD (B)*, find a point γ on $\overline{BD}$ from p_1 such that $\overline{p_1\gamma} = R_s$. Hence, *RMD* of node B is $\overline{\gamma B}$.

$$p(A) = p, \quad (4)$$

where, A is the coverage area of a node. Considering the coverage disc of a dead node and n number of head nodes, the effective coverage probability of all $(n+1)$ nodes can be calculated as

$$p(A + A + A \ldots (n+1)\text{times}) = 1 - (1-p)^{n+1}. \quad (5)$$

If a node is dead out of those $(n+1)$ nodes, the effective coverage probability (p_{eff}) of rest n nodes can be calculated as

$$p_{eff} = p(A + A + A \ldots n \text{ times}) = (1-p)[1-(1-p)^n]. \quad (6)$$

Suppose there are n head nodes those who move to maintain the coverage and connectivity due to death of a node and $A_1, A_2, A_3, \ldots, A_n$ be the coverage area of those n nodes, respectively. Then the effective coverage area (A_{eff}) of all head nodes can be calculated as

$$A_{eff} = \sum_{i=1}^{n} A_i - \sum_{i<k} A_i \bigcap A_k + \sum_{i<k<l} A_i \bigcap A_k \bigcap A_l. \quad (7)$$

In our protocol, since, $R_c = R_s$ and initially the whole network is connected, $A_{eff} > 0$ and therefore nodes must have enough overlapping area to compensate the coverage and connectivity. Let, A_d be the coverage area of a dead node, which may overlap with the head nodes (one-hop neighbors of a dead node). For any head node, if its AVD (a_{ij}) $\geqslant$ AMD (r_{ij}), there must be coverage overlapping between each head node with the dead node, which can be calculated as

$$A_{olp} = \sum_{i=1}^{n} A_i \bigcap A_d. \quad (8)$$

Let, A_r be the required area to be maintained due to death of a node. Based on our algorithms, the probability of coverage and connectivity maintenance (p_m) can be derived as

$$p_m = \begin{cases} 0, & \text{if } a_{ij} = 0, \\ \frac{A_{olp}}{A_r}, & \text{if } 0 < a_{ij} < r_{ij}, \\ 1, & \text{if } a_{ij} \geqslant r_{ij}. \end{cases} \quad (9)$$

In our protocol, since, a mobile node does not disturb its existing coverage and connectivity; there will be no further coverage or connectivity hole in the network. If value of $a_{ij} = 0$, it implies that there is no larger coverage overlapping among the nodes and therefore redeployment of few more sensors is necessary.

5. Performance evaluation

5.1. Simulation setups

In order to evaluate the performance of our coverage and connectivity maintenance algorithms, we have simulated our protocols using ns-2.29. In our simulations, 1000 sensors are deployed randomly over an area of $250m \times 250m$. A multi-hop and fully connected wireless sensor network is considered, in which all nodes of the network use the IEEE 802.15.4 [29] CSMA-CA to access the channel. The simulations are setup according to IEEE 802.15.4 MAC/PHY specification and radio characteristics of IEEE 802.15.4 compliant product CC2420 [30] along with AODV routing protocol and TwoRayGround propagation model. Since, MICAz is a hardware representative of IEEE 802.15.4 platform for TinyOS and widely used today, the packet length is set to be a constant length of 36 bytes with reference to the maximum packet length of MICAz specification [31]. Once the packet transmission starts, each node starts sending 1000 packets randomly. As per our assumptions of the coverage and connectivity maintenance algorithms, we have simulated under different communication range with fixed value of the sensing range. The sensing range is set to be 10 m. The traffic data rate is kept as 250 Kbps and beacon packets are sent in every 2 s. In order to justify our assumption, nodes are considered to be dead in our simulation in two different scenarios.

In the first scenario, a node is dead due to its energy exhaustion, which is assumed if its energy level reaches to zero. In order to justify the accidental death of nodes, few nodes are considered to be dead by selecting them randomly and by switching them off abruptly. Initially, each node is assumed to have fixed amount of 50 J reserved energy and energy cost due to mobility is taken as 1 J/m. Besides, a node is assumed to be dead, if it does not receive continuously two beacons from its neighbors. We first simulated the residual energy and percentage of alive nodes for different communication range. The residual energy is defined as the remaining energy of a node after the energy consumption due to its mobility. For example, if a node has 100 J as its initial energy and consumes 5 J due to its mobility, then its residual energy is 95 J. As per our assumption in the algorithms that $1 \leqslant R_c/R_s < 2$, communication range of the nodes in our simulation is considered to be 10 m and 15 m, since sensing range of each node is fixed to be 10 m. We have compared our simulation results with similar mobility protocols Co-Fi [22] and DCM [23], since both protocols propose the coverage maintenance due to limited mobility of nodes. Since, both Co-Fi [22], DCM [23] consider that $R_c \geqslant 2R_s$, communication range is considered to be 20 m in order to satisfy the conditions of these referred protocols and to compare their results with ours.

In order to simulate the coverage and connectivity maintenance, and average mobility distance, each node is given a unique ID seed and time seed, which are used to generate ID and next generating time, respectively. First, nodes use time seed to generate the next generating time, whose values lies between 4 and 20 s. When the generating time expires, each node uses the ID seed to generate a random node ID. If the generated ID of a node is same as its own ID, it sets its residual energy to zero and is assumed to be dead. Hence, nodes in our simulation die in a probabilistic manner. Since, *Co-Fi*, *DCM* and our algorithms are extremely applicable for the death of only one node at the same time, this arrangement of node lifetime can promise at most one node dies for the average of 12 s. In our simulation, life time of flooding message for updating sensing neighbors in *Co-Fi* is considered as 10 hops and we allow the nodes to send panic message [22] and to decide about a node that should move before the death of each node from a fair point of view. We do not deploy any additional node in our simulation, when cascading mobility is unable to solve the problem. Based on our algorithms, simulations are run for 20 different topologies. In order to compare the performance of our algorithms with *DCM*, we only consider the condition that $R_c = 2R_s$, as it totally complies with its assumptions.

5.2. Simulation results

The simulation results of average residential energy of all nodes of the network with different communication ranges, taking $R_c = 2R_s$ and with fixed coverage range of 10 m are shown in Fig. 10 for different intervals of time. It is observed that the average residual energy in *Co-Fi* [22] is least enough as compared to *DCM* [23] and *CoCo*. This is because of the higher communication range of the nodes and flooding of the panic message [22], as per the algorithms of *Co-Fi*. Since, communication range is allowed to be less than 20 m, as $R_c < 2R_s$ or equal to be 10 m, as $R_c = R_s$ in *CoCo*, it is found that the average residential energy is increased by reducing the communication range. From Fig. 10, it is to be noted that *CoCo* outperforms *DCM* in terms of energy consumption, and fulfil the same purpose of maintaining the coverage and connectivity through the limited mobility of nodes. As shown in Fig. 11, we have also simulated our algorithms and have compared the results with *DCM* and *Co-Fi*, in terms of percentage of alive nodes with time for various communication range. As we know, communication is the main source of energy consumption. Since, we have considered the least communication range as compared to *DCM* and *Co-Fi*, we got the most expected result, which is better than both *DCM* and *Co-Fi*.

In order to verify the performance in terms of average mobility distance of nodes, we present the performance of *CoCo*, as shown in Fig. 12, for different communication range. Since, limited mobility is one of the characteristic in *CoCo*, we compare the average mobility distance of *CoCo*

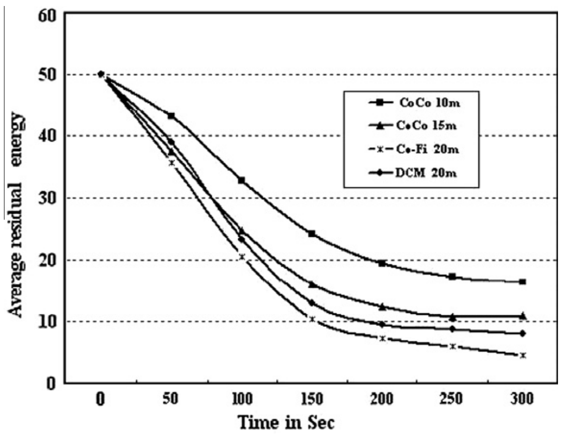

Fig. 10. Average residual energy for different communication range with fixed value of sensing range i.e. when R_s = 10 m.

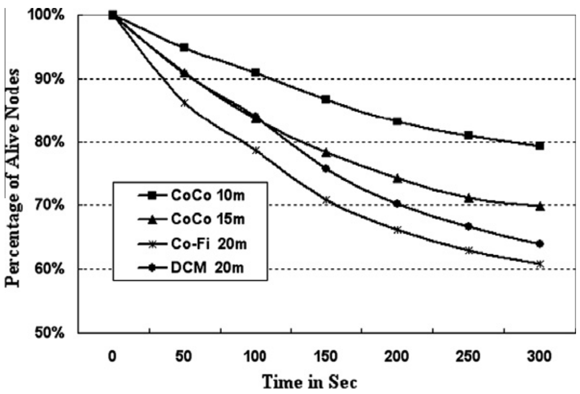

Fig. 11. Percentage of alive nodes with time, for different communication range and fixed value of sensing range i.e. when R_s = 10 m.

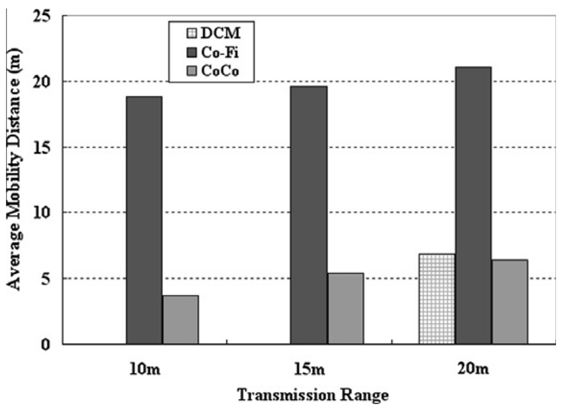

Fig. 12. Average mobility distance vs transmission range, for fixed R_s=10 m.

with similar protocols and find that the average mobility distance of *CoCo* is extremely less than *Co-Fi*, which is not a limited mobility model. In our algorithm, we find that the average mobility distance is increased, while communication range is increased. This is because larger the communication range, longer distance the sensor nodes have to move, when communication problem arise. On the other hand, with larger communication range, each node could have more *MTMD*, which may increase the value of *AVD*, for which average mobility distance in *CoCo* is also increased. *DCM* allows only one node to move after coverage hole is appeared, whereas, in *CoCo* multiple nodes are involved to solve the connectivity problem. Hence, each node has a least average mobility distance and average mobility distance in our work is slightly less than DCM, when we simulate our algorithms taking $R_c = 2R_s$.

The network coverage deterioration is defined as the weakening in overall coverage of the network with time. It is to be noted that the coverage of the network may be degraded due to death of several nodes. Hence, taking different values of coverage and communication range, our protocol is simulated and compared with similar coverage maintenance algorithms. In order to verify the network coverage deterioration, we have presented our simulation results in Figs. 13 and 14. In order to get a fair comparison, we have simulated *CoCo* taking R_c = 20 m, as both *Co-Fi* and *DCM* consider $R_c = 2R_s$. As shown in Fig. 13, we can see that our algorithm has better performance than *Co-Fi* for certain period of time i.e. within 50 s and always outperforms *DCM*. As shown in Fig. 14, *Co-Fi* outperforms *CoCo* for $R_c < 2R_s$ or $R_c = R_s$. It is observed that due to less value of R_c, less percentage of region is covered in *CoCo*. Obviously, R_c is correlated to *MTMD*, and due to reduction in R_c, number of disjoint transmission sets of a node is increased, thereby reducing the value of *MTMD* and *AVD*, as well. With less value of *AVD*, there is high probability that nodes may be unable to solve the coverage or connectivity problems. Hence, performance is degraded when R_c is reduced in *CoCo*. The better performance of *Co-Fi* over *CoCo* is due to the long average mobility distance of the nodes in *Co-Fi*, as they ask the redundant nodes to replace the dead nodes and thereby their performance is better than ours. However, the redundancy of nodes in the network may not happen always and no redundant node may be available after several replacements of the dead nodes. Besides, long average mobility distance of a node causes more energy consumption and thereby death of the nodes at a faster rate. From this point of view, our algorithm is better than *Co-Fi*, as we use limited average mobility distance.

The performance evaluation of *CoCo* in terms of connectivity maintenance is presented in Figs. 15 and 16. In Fig. 15, we compare our results with both *Co-Fi* and *DCM*, taking $R_c = 2R_s$, whereas in Fig. 16, we compare *CoCo* with *Co-Fi*, taking $R_c = R_s$ or $R_c < 2R_s$, which is not a case in *DCM*. As shown in Fig. 15 *CoCo* outperforms *DCM* and worse performs *Co-Fi* in terms of connectivity, as number of partitions are increased with time in *CoCo*. This situation arises in *CoCo*, as the number of neighbors is decreased when R_c is reduced and thereby increasing the probability of partitions. However, as discussed above, since we do not consider the redundant nodes to compensate the coverage or connectivity, we use limited mobility distance of the nodes to maintain the network, thereby save more node energy. As shown in Fig. 16, it is observed that *CoCo* outperforms *Co-Fi* in terms of connectivity, either for $R_c < 2R_s$

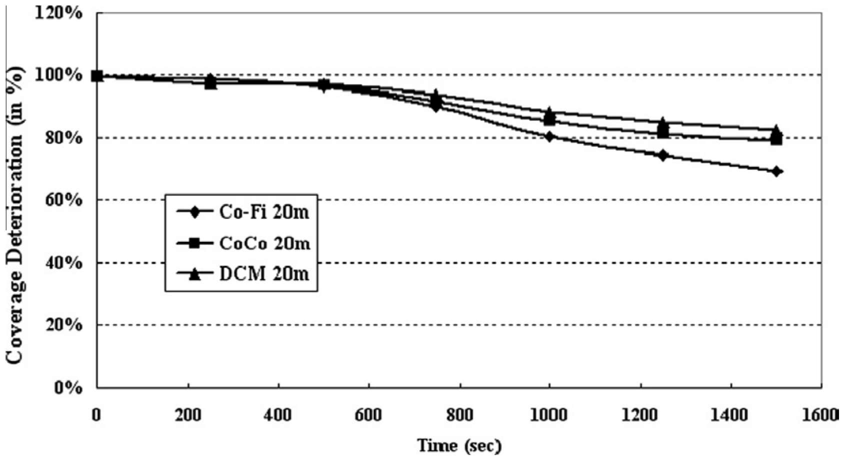

Fig. 13. Percentage of coverage deteriorations for $R_c = 20$ m and $R_s = 10$ m.

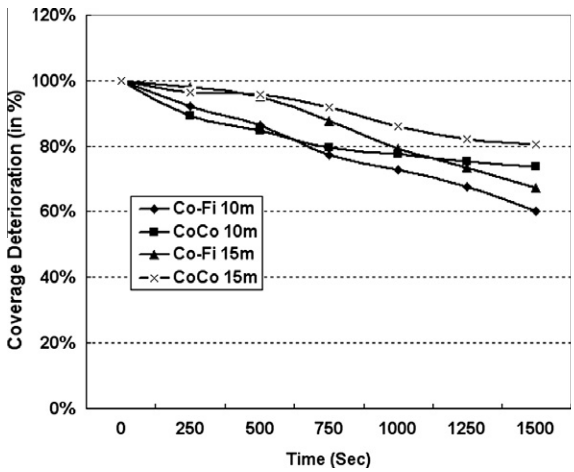

Fig. 14. Percentage of coverage deteriorations for $R_c = 10$ m, 15 m and $R_s = 10$ m.

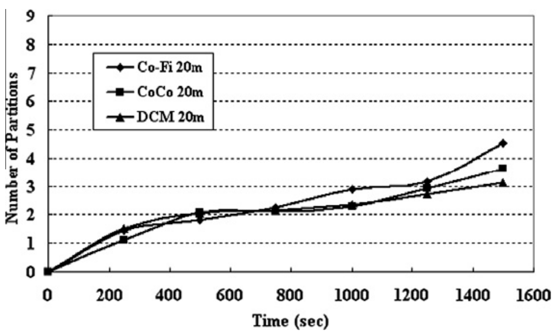

Fig. 15. Possible number of network partitions for $R_c = 20$ m and $R_s = 10$ m.

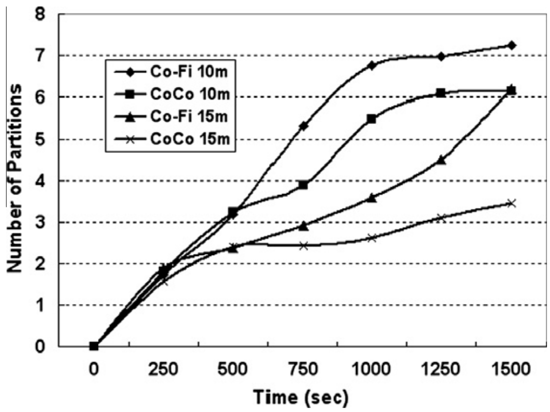

Fig. 16. Possible number of network partitions for $R_c = 10$ m, 15 m and $R_s = 10$ m.

or $R_c = R_s$. Our proposed algorithms can solve the network partitions or coverage holes with least energy consumption and least average mobility distance. Besides, our algorithms can outperform in terms of coverage and connectivity for smaller communication range. As shown in Fig. 17, the network lifetime is analyzed with average mobility distance of the nodes taking different values of communication ranges. Network lifetime is defined as the duration of time from the initial deployment until when the sensing coverage or connectivity of the network is lost due to death of a node. In order to better understand the effect of smaller and larger communication range on the network lifetime, we simulated our protocol with the baseline no replacement policy. The nodes at different locations are moved to maintain the network as soon as a node is dead. It is observed that network lifetime decreases with increase in communication range of a node. Further more, it is always shorter for longer communication range, irrespective of static or mobile nature of the node. For a fixed value of communication range, the network lifetime is longer in absence of node mobility, which is indicated in the figure for average mobility distance=0. Combining the joint effect of communication range and mobility distance, it is found that the network lifetime is degraded to a large extend as shown in Fig. 17.

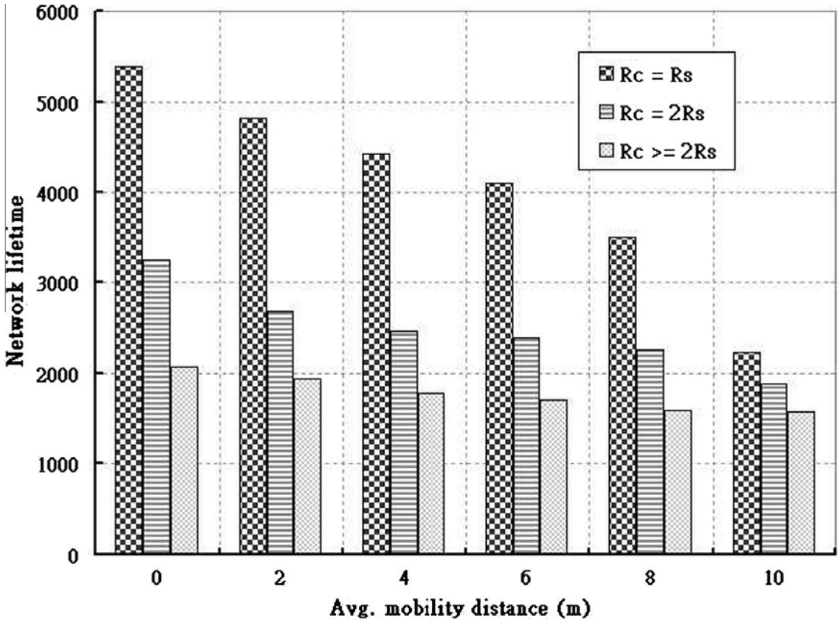

Fig. 17. Effect of various communication range on network lifetime for different values of mobility distance with fixed $R_s = 10$ m.

6. Conclusion

In this work, we design the limited mobility based connectivity and coverage maintenance protocols for the multi-hop wireless sensor networks. A distinguishing feature in our work is that communication range is equal to the sensing range ($R_c = R_s$) and mobility of nodes in our algorithms is limited within only one-hop neighbors of a dead node. Besides, our algorithms can also be useful to maintain the connectivity and coverage of the network for $R_c < 2R_s$. Our protocols can be applicable to unpredictable death of nodes, such as destruction of nodes due to explosion or technical failures and also work fine for the predictable death such as death due to energy exhaustion of the nodes. We design dynamic maintenance algorithms without disturbing the existing communication and coverage systems of the network taking $1 \leqslant R_c/R_s < 2$, and our limited mobility of nodes can guarantee the network integrity. From our performance analysis, it is clear that our algorithms outperform in terms of energy consumption and average mobility distance as compared to similar algorithms along this direction.

References

[1] A. Jardosh, E.M. Belding-Royer, K.C. Almeroth, S. Suri, Towards realistic mobility models for mobile ad hoc networks, Proc. IEEE MobiCom 2003 (2003) 217–229.
[2] M. Grossglauser, D. Tse, Mobility increases the capacity of adhoc wireless networks, Proc. IEEE INFOCOM (2001).
[3] M.B. McMickell, B. Goodwine, L.A. Montestruque, MICAbot: a robotic platform for large-scale distributed robotics, in: Proceedings of IEEE International Conference on Robotics and Automation, vol. 2, 2003, pp. 1600–1605.
[4] J.P. Sheu, P.W. Cheng, K.Y. Hsieh, "Design and implementation of a smart mobile robot, in: Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, vol. 3, 2005, pp. 422–429.
[5] S. Bergbreiter, K.S.J. Pister, CotsBots: an off-the-shelf platform for distributed robotics, in: Proceedings of IEEE International Conference on Intelligent Robots and Systems, 2003.
[6] G.T. Sibley, M.H. Rahimi, G.S. Sukhatme, Robomote: a tiny mobile robot platform for large-scale sensor networks, in: Proceedings of IEEE International Conference on Robotics and Automation, 2002, pp. 1143–1148.
[7] G. Kesidis, T. Konstantopoulos, S. Phoha, Surveillance coverage of sensor networks under a random mobility strategy, Proc. IEEE Sensors 2 (2003) 961–965.
[8] J.-P. Sheu, P.-C. Chen, C.-S. Hsu, A distributed localization scheme for wireless sensor networks with improved grid-scan and vector-based refinement, IEEE Trans. Mobile Comput. 7 (9) (2008) 1110–1123.
[9] H. Zhang, J.C. Hou, Maintaining sensing coverage and connectivity in large sensor networks, Ad Hoc Sensor Wireless Netw. 1 (2005) 89–124.
[10] D. Tian, N.D. Georganas, Connectivity maintenance and coverage preservation in wireless sensor networks, Ad Hoc Netw. 2 (2005) 744–761.
[11] N. Ahmed, S.S. Kanhere, S. Jha, "Probabilistic coverage in wireless sensor networks, in: Proceedings of IEEE Conference on Local Computer Networks, 2005, pp. 672–681.
[12] P.-J. Wan, C.-W. Yi, Coverage by randomly deployed wireless sensor networks, IEEE Trans. Inform.Theory 52 (6) (2006) 2658–2669.
[13] J. Carle, A. Gallais, D. Simplot-Ryl, Preserving area coverage in wireless sensor networks by using surface coverage relay dominating sets, in: Proceedings of IEEE Symposium on Computers and Communications, 2005, pp. 347–352.
[14] Z. Butler, D. Rus, Controlling mobile sensors for monitoring events with coverage constraints, in: Proceedings of IEEE International Conference on Robotics and Automation, vol. 2, 2004, pp. 1568–1573.
[15] Y. Gao, K. Wu, F. Li, Analysis on the Redundancy of Wireless Sensor Networks, in: Proceedings of Second ACM international conference on Wireless sensor networks and applications, 2003, pp. 108–114.
[16] S.-F. Jiang, M.-H. Yang, H.-T. Song, J.-M. Wang, An enhanced perimeter coverage based density control algorithm for wireless sensor network, in: Proceedings of IEEE Third International Conference on Wireless and Mobile Communications, 2007.
[17] A. Howard, M.J. Mataric, G.S. Sukhatme, Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem, in: Proceedings of International Symposium on Distributed Autonomous Robotic Systems, 2002, pp. 299–308.
[18] G. Wang, G. Cao, T.L. Porta, Movement-assisted sensor deployment, IEEE Trans. Mobile Comput. 5 (6) (2006) 640–652.

[19] R. Zheng, G. He, X. Liu, Location-free coverage maintenance in wireless sensor networks, Technical Report, University of Houston, USA, July, 2005.
[20] H. Yu, J. Iyer, H. Kim, E.J. Kim, K.H. Yum, P.S. Mah, Assuring k-coverage in the presence of mobility in wireless sensor networks, in: Proceedings of IEEE Global Telecommunications Conference, 2006, pp. 1–5.
[21] S. Chellappan, X. Bai, B. Ma, D. Xuan, C. Xu, Mobility limited flip-based sensor networks deployment, IEEE Trans. Parallel Distrib. Syst. 18 (2) (2007) 199–211.
[22] S. Ganeriwal, A. Kansal, M.B. Srivastava, Self aware actuation for fault repair in sensor networks, in: Proceedings of IEEE international conference on robotics and automation, vol. 5, May 2004, pp. 5244–5249.
[23] A. Sekhar, B.S. Manoj, C.S.R. Murthy, Dynamic coverage maintenance algorithms for sensor networks with limited mobility, in: Proceeding of IEEE International Conference on Pervasive Computing and Communications, 2005, pp. 51–60.
[24] Q. Qiu and A.E. Kamal, Coverage and connectivity control of wireless sensor networks under mobility, Proceedings of Workshop on High Performance Switching and Routing, pp. 177-181, 2005.
[25] R. Min, A. Chandrakasan, Energy-efficient communication for ad-hoc wireless sensor networks, in: Proceedings of Conference Record of the 35th Asilomar Conference on Signals, Systems and Computers, vol. 1, 2001, pp. 139–143.
[26] B. Karp, H.T. Kung, Greedy perimeter stateless routing for wireless networks, in: Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking, 2000, pp. 243–254.
[27] <http://www.kvh.com/DigiComp/>.
[28] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, second ed., MIT Press and McGraw-Hill, 2001.
[29] IEEE Std 802.15.4. Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs), 2003.
[30] Data sheet for CC2420 2.4GHz IEEE 802.15.4/ZigBee RF transceiver, <http://www.chipcon.com/files/CC2420_Data_Sheet_1_2.pdf>.
[31] Data sheet for Crossbow wireless sensor networks radio module: MPR2400-MICAz ZigBee series, <http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz>.
[32] A. Bereketli, O.B. Akan, Communication coverage in wireless passive sensor networks, IEEE Commun. Lett. 13 (2) (2009) 133–135.
[33] H.M. Ammari, S.K. Das, Integrated coverage and connectivity in wireless sensor networks: a two-dimensional percolation problem, IEEE Trans. Comput. 57 (10) (2008) 1423–1434.
[34] X. Li, D.K. Hunter, Distributed coordinate-free hole recovery, in: Proceedings of IEEE ICC International Conference Communications Workshops, May 2008, pp. 189–194.
[35] Y. Tian, S.-F. Zhang, Y. Wang, A distributed protocol for ensuring both probabilistic coverage and connectivity of high density wireless sensor networks, in: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), 2008, pp. 2069–2074.

Prasan Kumar Sahoo received Master of Technology degree in Computer Science from the Indian Institute of Technology (IIT), Kharagpur, India in 2000, Master of Science degree in Mathematics from Utkal University, India in 1994. He received 1st Ph.D. in Mathematics from Utkal University, India and 2nd Ph. D. in Computer Engineering from National Central University, Taiwan in 2002 and 2009, respectively. He is currently working as Assistant Professor in the department of Computer Science and Information Engineering, Chang Gung University, Taiwan. Earlier, he was working as Associate Professor in the dept of Information Management, Vanung University, Taiwan. He has also worked in the Software Research Center of National Central University, Taiwan. His current research interests include Wireless Sensor Networks and VANET related to mobility modeling, performance analysis and coverage connectivity maintenance issues. He has served as the Program Committee Member of several IEEE, ACM and international conferences and reviewer of several international journals including IEEE TPDS, IEEE TWC etc. Dr. Sahoo is a member of IEEE and ACM.

Jang-Ping Sheu received the B.S. degree in computer science from Tamkang University, Taiwan, Republic of China, in 1981, and the M.S. and Ph.D. degrees in computer science from National Tsing Hua University, Taiwan, Republic of China, in 1983 and 1987, respectively. He is currently a Chair Professor of the Department of Computer Science, National Tsing Hua University. He was a Chair of Department of Computer Science and Information Engineering, National Central University from 1997 to 1999. He was a Director of Computer Center, National Central University from 2003 to 2006. His current research interests include wireless communications and mobile computing. He was an associate editor of Journal of the Chinese Institute of Electrical Engineering, Journal of Information Science and Engineering, Journal of the Chinese Institute of Engineers, and Journal of Internet Technology. He is an associate editor of the IEEE Transactions on Parallel and Distributed Systems, International Journal of Ad Hoc and Ubiquitous Computing, and International Journal of Sensor Networks. He received the Distinguished Research Awards of the National Science Council of the Republic of China in 1993–1994, 1995–1996, and 1997–1998. He received the Distinguished Engineering Professor Award of the Chinese Institute of Engineers in 2003. He received the certificate of Distinguished Professorship, National Central University in 2005. He received the K.-T. Li Research Breakthrough Award of the Institute of Information and Computing Machinery in 2007. Dr. Sheu is an IEEE Fellow, a member of the ACM, and Phi Tau Phi Society.