This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2007 proceedings.

PALM: A Partition Avoidance Lazy Movement Protocol for Mobile Sensor Networks

Kuei-Ping Shih, Hung-Chang Chen, Jing-Kuen Tsai, and Chun-Chih Li

Department of Computer Science and Information Engineering, Tamkang University, Tamshui 251, Taipei, Taiwan
Email: kpshih@mail.tku.edu.tw

***Abstract*— The paper proposes a distributed partition avoidance lazy movement (PALM) protocol for mobile sensor networks (MSNs). In general, connectivity and coverage are two major factors to the success of a sensor network. Therefore, PALM takes both connectivity and coverage into account to avoid network partition and keep high sensing quality. Since sensor movement is the major source of energy consumption, thus, in order not to cause frequent movement, PALM triggers sensor movement only when the network has a risk of partition, but not when coverage holes appear. The paper proposes a sufficient condition of keeping a network connected. Based on the condition, PALM adopts the *lazy movement policy* for a sensor to determine when to move and uses the *principles of an effective movement* for a sensor to decide where to move. Accordingly, PALM can keep the network connected and can make the effective coverage as large as possible to maintain high sensing quality. In comparison with the related work, PALM can reduce the energy consumption and further extend the network lifetime due to the *lazy movement policy* and the *principles of an effective movement*. Simulation results also verify the advantages of the proposed protocol.**

I. INTRODUCTION

Wireless sensor networks (WSNs) have been recognized as a new paradigm for data-centric computing. Mobile sensor network (MSN) is a kind of WSN. In addition to common capabilities in a WSN, such as sensing, processing, and communication, the sensor in an MSN is still capable of movement while needed. Mobile sensors are useful in some situations, such as they can move to prevent from coverage holes while they are deployed in a hostile environment or they can move to replace the malfunctioned sensors. However, movement is regarded as the most energy consuming part for mobile sensors. Thus, movement issues, such as when to move or where to move, should be carefully concerned for MSNs in order to benefit from movements.

Coverage and *connectivity* are two key factors to the success either for a WSN or for an MSN. The coverage issue concerns that how well the area to be sensed by the active sensors. It requires satisfying some requirements, such as the quality of surveillance (QoSv) or k-coverage of the sensing area, for some $k \geq 1$. On the other hand, the connectivity issue emphasizes how well the active sensors connect to each other or to the sink such that their data can be delivered to the destination or to the sink. Clearly, even though one scheme can obtain maximal sensing coverage, without ensuring the connectivity of sensors to the sink, it is also useless.

Based on the concept, the paper proposes a partition avoidance lazy movement (PALM) protocol, which takes both connectivity and coverage into consideration for mobile sensor networks. The primary goal of PALM is to connect as large amount of sensors as possible to one connected component, including the sink. In addition, since the movement of a mobile sensor is much power-consuming, in PALM, sensors will not move unless the sensor network has a high risk to be partitioned. It is so called *lazy movement* in the paper. In addition, for the consideration of connectivity, the movement of sensors has to obey the *principles of an effective movement*, which could guarantee that this movement is beneficial to the connectivity and the coverage of the sensor network. Simulation results show that, with the assistance of PALM, the lived sensors can keep connected to the sink constantly and the coverage of the MSN can be preserved in a certain quality in terms of the number of remaining lived sensors.

The rest of the paper is organized as follows. Related work is presented in Section II. Section III describes the proposed protocol, PALM. The simulation results are shown in Section IV. Finally, Section V concludes the paper.

II. RELATED WORK

Many researches take the coverage and connectivity issues into consideration on MSNs [1]–[5]. In [1], based on the concept of virtual force, two schemes, DSSA and IDCA, to spread the sensors around the sensor network in order to obtain as large coverage as possible are proposed. In addition, based on Voronoi diagram, three approaches, Vector-based (VEC), Voronoi-based (VOR), and Minimax schemes, are proposed in [2] to distribute sensors from densely deployed areas to sparsely deployed areas. In [3], a Scan-based Movement-Assisted sensoR deploymenT (SMART) scheme is proposed, which uses scan and dimension exchanges to balance the distribution of mobile sensors. On the other hand, in [4], a self aware actuation scheme for fault repair, named Co-Fi, is proposed for coordinating coverage fidelity maintenance in sensor networks. In [5], the authors utilize the motion capability of sensors to relocate the redundant sensors to overcome the failure of other sensors and to preserve the certain initial coverage. The former three schemes consider the deployment of mobile sensors to move sensors from an initial unbalanced state to a balanced state. The latter two schemes focus on the fault repair of the sensor network to preserve the sensing coverage.

As mentioned above, coverage and connectivity are two important factors to be considered by sensor networks. However, the above schemes, both the deployment schemes and the fault repair schemes, only take coverage issue into consideration, not taking the connectivity factor into account. Therefore, these schemes suffer the risk of disconnections of sensors to the sink. Hence, although maximizing sensing coverage is an important goal to achieve in sensor networks, delivering sensing data to the sink is also an important requirement in sensor networks. PALM can keep the network connected and avoid mobile sensors from blind movements in order not to lose the connection to the sink. Besides, in PALM, a sensor moves only when the network suffers a risk of disconnection. Thus, the number of movements in PALM is much smaller in comparison with the related work.

III. PALM: THE PROTOCOL

The goals of the proposed protocol, PALM, are to make lived sensors connect to the sink to avoid network partition and maximize the value of lived sensors. Simultaneously, the effective coverage is to be maximized as well.

A. Network Model and Assumptions

An MSN considered in the paper consists of several mobile sensors and a sink. Let $\mathbb{S}$ stand for the sink and the sensors be denoted s_i, $i = 1, 2, \ldots$. Without loss of generality, it is assumed that data is always destined to $\mathbb{S}$. The sensor with sufficient energy for sensing, communication, and computing is called a *lived sensor*. The set of lived sensors is denoted LS. In the paper, all sensors have the same sensing, communication, processing, and mobility capabilities. Let r_s and r_c respectively denote the sensing range and the communication range of a sensor. To simplify the discussion, it is assumed that $r_c \geq 2r_s$. Let $N(s)$ stand for the set of neighbors of s. That is, $N(s) = \{s' \in LS \mid d(s, s') \leq r_c\}$, where $d(s, s')$ is the Euclidean distance between s and s'. It is also assumed that a sensor is location-aware. The location of a sensor can be obtained either by GPS or by any existing localization scheme. Moreover, each sensor has the location of $\mathbb{S}$.

Initially, the MSN is connected. All sensors can connect to $\mathbb{S}$. Every sensor can obtain its own hop-to-$\mathbb{S}$ distance, which can be obtained by a message originated from $\mathbb{S}$ flooded throughout the network. Let $h(s)$ denote the hop-to-$\mathbb{S}$ distance of s and $h(\mathbb{S}) = 0$.

As mentioned above, the sensing coverage should take only those sensors which can deliver sensing data to $\mathbb{S}$ into account. Therefore, the area covered by the sensors capable of delivering sensing data to $\mathbb{S}$ is referred as the *effective coverage* of the sensor network and those sensors contributing to the effective coverage are termed *effective sensors* in the paper. Let ES denote the set of effective sensors. Obviously, $ES \subseteq LS$.

The formal definitions of effective sensors and effective coverage are described as follows.

Definition 1 (Effective Sensor and Effective Coverage): The sensor which can connect to the sink is an effective sensor. The area covered by effective sensors is the effective coverage of the network. □

B. Lazy Movement Policy (When to move?)

As long as a network has a risk to be partitioned, some sensors have to move to avoid the occurrence of partition. As a result, it would be of great help if it could be predicted in advance when a network would be partitioned. However, to know whether the network is partitioned not only costs much, but also is hard to achieve in a distributed manner just depending on one-hop neighboring information. Therefore, the paper proposes a distributed approach for a sensor to self-determine whether it needs to move in order to prevent from network partition and the sensor only needs one-hop neighboring information.

Suppose an MSN is composed of m connected components, $CC_1, CC_2, \ldots, CC_m$, after working for some period of time, where $m \geq 1$. Without loss of generality, let CC_1 be the connected component including $\mathbb{S}$. Ideally, the largest number of effective sensors in an MSN is that $ES = LS$, i.e., $m = 1$. According to the hop-to-$\mathbb{S}$ distance, $N(s)$, the set of neighbors of s, can be classified into three disjoint subsets as follows.

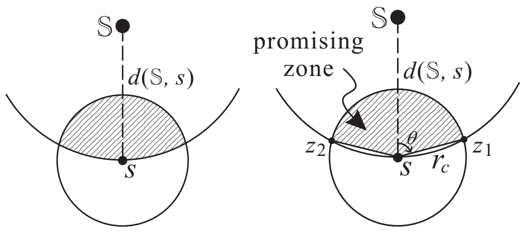
Definition 2 (Up-, Mid-, and Down-stream Sensors): Given a sensor $s \in LS$, let $s' \in N(s)$. If $h(s') < h(s)$, s' is an *upstream* sensor of s. If $h(s') > h(s)$, s' is a *downstream* sensor of s. Otherwise (i.e. $h(s') = h(s)$), s' is a *midstream* sensor of s. The sets of upstream, midstream, and downstream sensors are denoted $U(s)$, $M(s)$, and $D(s)$, respectively. Accordingly, $N(s) = U(s) \cup M(s) \cup D(s)$. □

The sufficient condition to keep a network connected is given below.

Lemma 1: $\forall\ s \in LS, U(s) \neq \emptyset \Longrightarrow$ the MSN is connected.

Proof: For any sensor $s \in LS$, there exists at least one upstream sensor, say s', because $U(s) \neq \emptyset$. Thus, s can connect to s' in the direction of $\mathbb{S}$ since $h(s') < h(s)$. Likewise, for sensor s', there also exists one upstream sensor, say s'' such that s' can connect to s'' and $h(s'') < h(s')$. Accordingly, it implies that for any sensor $s \in LS$, there must exist sensors $s_1, s_2, s_3, \ldots, s_i$, for some i, such that $s_1 \in U(s_2)$, $s_2 \in U(s_3), \ldots$, and $s_i \in U(s)$. Moreover, $\mathbb{S} \in U(s_1)$. Otherwise, s_1 must be able to find an upstream sensor since $U(s_1) \neq \emptyset$. Consequently, there exists one path from $\mathbb{S}$ along $s_1, s_2, s_3, \ldots, s_i$ to s. As a result, $s \in CC_1$. Thus, for any sensor $s \in LS, s \in CC_1$. That is, only one connected component exists in the MSN. It concludes that the MSN is connected. □

Based on Lemma 1 that each sensor should connect to at least one upstream sensor to keep the network connected, for any sensor, say s, it is useful to figure out the feasible and promising location of the upstream sensor, where the promising location means that the location of the upstream sensor should be closer to $\mathbb{S}$ than that of s. It is to make sure that data delivery of s will be directly toward $\mathbb{S}$. Therefore, a new term, the promising upstream sensor, is defined as follows.



(a) (b)

Fig. 1. (a) The shadow area is the possible location of the promising upstream sensor of s. (b). The promising zone of s, $\mathcal{Z}(s)$.

Definition 3 (Promising Upstream Sensor): Given a sensor $s \in LS$, if s' satisfies that $s' \in U(s)$ and $d(s', \mathbb{S}) < d(s, \mathbb{S})$, s' is a promising upstream sensor of s, and vice versa. □

Let $C(s, r)$ denote a circle centered at s with radius r. The following lemma shows the possible location of promising upstream sensors of s.

Lemma 2: Given a sensor $s \in LS$, s' is a promising upstream sensor of $s \Longrightarrow s' \in C(\mathbb{S}, d(\mathbb{S}, s)) \cap C(s, r_c)$.

Proof: Obviously, $s' \in C(s, r_c)$ since $s' \in U(s)$. On the other hand, since s' is promising, it implies $d(s', \mathbb{S}) < d(s, \mathbb{S})$. Therefore, $s' \in C(\mathbb{S}, d(\mathbb{S}, s))$. Consequently, $s' \in C(\mathbb{S}, d(\mathbb{S}, s)) \cap C(s, r_c)$. □

Fig. 1(a) is an illustration of the possible location of promising upstream sensors of s. For simplicity, a promising zone of s, denoted $\mathcal{Z}(s)$, is defined to stand for the possible location of the promising upstream sensors of s, instead of the intersection of $C(\mathbb{S}, d(\mathbb{S}, s))$ and $C(s, r_c)$. The definition of the promising zone of s is defined as follows.

Definition 4 (Promising Zone): $\mathbb{S}$ is the sink and $s \in LS$ is any sensor in the network. Let the intersection points of two circles $C(\mathbb{S}, d(\mathbb{S}, s))$ and $C(s, r_c)$ be z_1 and z_2. The promising zone of s, denoted $\mathcal{Z}(s)$, is defined as the circular sector formed by two radii $\overline{sz_1}$ and $\overline{sz_2}$, and the arc $\widehat{z_1 z_2}$. □

The promising zone of s can be figured out as follows. Since each sensor has the location of $\mathbb{S}$, thus, for sensor s, its promising zone is the area that, toward the direction of $\mathbb{S}$, left and right each spans an angle of θ, where $\theta = \cos^{-1} \frac{r_c}{2d(\mathbb{S}, s)}$. Fig. 1(b) is an illustration of the promising zone of s. Obviously, $\mathcal{Z}(s) \subseteq C(\mathbb{S}, d(\mathbb{S}, s)) \cap C(s, r_c)$. Although there is a slight difference between the two areas, the difference can be neglected, especially when s is far away from $\mathbb{S}$ since $\lim_{d(\mathbb{S}, s) \to \infty} \theta = \frac{\pi}{2}$. However, $\mathcal{Z}(s)$ is much easier to obtain than to calculate the intersection of two circles.

To define $\mathcal{Z}(s)$ is for s to check whether there exists any sensor in $\mathcal{Z}(s)$ such that s can regard that sensor as a promising upstream sensor. If there exists at least one sensor in $\mathcal{Z}(s)$, s can regard that sensor as its promising upstream sensor, even though the hop-to-$\mathbb{S}$ distance of the sensor is equal to that of s currently. On the other hand, Lemma 1 also implies that if a network is disconnected, there must exist at least one sensor whose promising zone contains no other sensor. Actually, each connect component CC_i, for some $i, i \neq 1$, has at least one sensor whose promising zone has no other sensor. As a result, if s can not find any promising upstream sensor, s will decide to move to connect to at least one sensor with smaller hop-to-$\mathbb{S}$ distance as its promising upstream sensor. Therefore, a new role for a sensor under such a situation that no sensor exists in its promising zone is defined as below.

Definition 5 (Risk Sensor): s is a risk sensor $\Longleftrightarrow \nexists\, s' \in \mathcal{Z}(s)$, where $s, s' \in LS$.

Accordingly, in this paper, only when a sensor turns to be a risk sensor, the sensor then moves to connect to a sensor whose hop-to-$\mathbb{S}$ distance is smaller than itself in order to keep the MSN always connected. That is the reason why it is called *lazy movement* in the paper. Thus, the policy that a sensor determines whether it needs to move is called the *lazy movement policy* and is specified as follows.

Lazy Movement Policy (*When to move*). *The lazy movement policy is that once a sensor becomes a risk sensor, the sensor decides to move.* □

We conclude the above in the following theorem.

Theorem 1: A sensor network is connected if all lived sensors obey the lazy movement policy.

Proof: It is obtained directly from Lemma 1 and the lazy movement policy. □

C. Principles of an Effective Movement (Where to move?)

According to Lemma 1, as long as each sensor can connect to a sensor with smaller hop-to-$\mathbb{S}$ distance, the MSN is connected. Moreover, the movement of a risk sensor should be effective and efficient. Therefore, a risk sensor has to connect to at least one sensor whose hop-to-$\mathbb{S}$ distance is smaller than itself and the attached sensor should be located in the promising zone of the risk sensor. In addition to the connectivity concern, coverage is also an important factor for sensor movement. Hence, for each movement, the moving sensor has to satisfy the coverage requirement of sensing quality. Moreover, the overlap of the sensing area of the moving sensor with those of attached sensors should be as least as possible. However, to maintain the sensing quality, the movement should not cause additional coverage hole. As a result, the principle of sensor movements to be followed are summarized as follows and are also termed *principles of an effective movement.*

Principles of an Effective Movement (*Where to move*). *An effective movement should obey the following principles:*

1) *The movement should make the risk sensor attach to at least one sensor with smaller hop-to-$\mathbb{S}$ distance and at least one attached sensor should be located in the promising zone of the risk sensor.*
2) *The movement should try to satisfy the coverage requirement of sensing quality.*
3) *The movement should neither make too much coverage overlap and nor create additional coverage hole with the attached sensors.* □

The coverage requirements highly depend on the needs of applications. For simplicity, the paper assumes that only 1-

cover of the area that the remaining lived sensors can cover as large as possible is required. Moreover, to meet the general coverage requirement that no coverage hole exists in the covered area, it is also assumed not to create any additional coverage hole while a moving sensor attaches to the attached sensors.

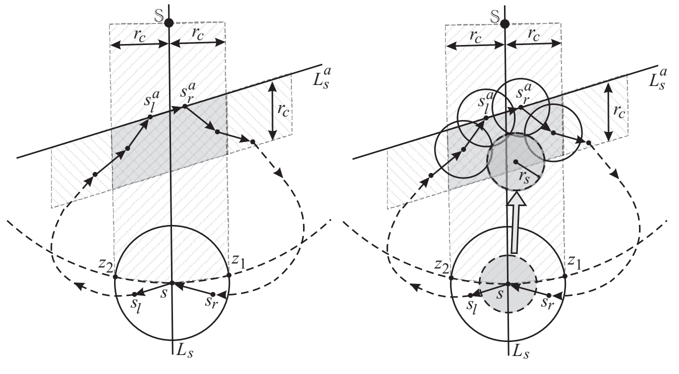
A moving sensor triggered by *lazy movement policy* moves according to the *principles of an effective movement*. The location compliant to the *principles of an effective movement* for a risk sensor to move can be obtained by the following two procedures. The first procedure is *attached sensors discovery* and the second one is *target position calculation*. They are described in detail as follows.

1) Attached Sensors Discovery: The goal of the attached sensors discovery is to discover promising attached sensors for the risk sensor to attach to such that the network can keep connected. Since a risk sensor is most likely to be on the boundary of a coverage hole or on the boundary of another connected component, not including the sink (i.e. $m \neq 1$). A common and popular scheme used to bypass the coverage hole is the right-hand rule (RHR) [6], [7]. PALM borrows the scheme proposed in [8] to bypass the hole in order to find promising sensors.

As mentioned in the first principle of an effective movement, it would be better that the promising attached sensors are located in the line from the risk sensor to $\mathbb{S}$ or therearound. Let the line passing through the risk sensor s and $\mathbb{S}$ be L_s. It has the form $L_s = \{(x, y)|ax + by + c = 0\}$. Basically, L_s divides a plane into three parts. One is the *positive* part, that is, $L_{s^+} = \{(x, y)|ax + by + c > 0\}$, one is the *negative* part, i.e., $L_{s^-} = \{(x, y)|ax + by + c < 0\}$, and L_s itself. Two points located in different parts will have different sign. According to the concept, the attached sensors can be found as follows.

Take Fig. 2(a) as an illustrated example. Without loss of generality, let the risk sensor be s. By RHR, s will initiate an **AttDis** packet to bypass the hole in order to find the attached sensors. During bypassing the hole, the sensors located in the slashed area should append their locations in the **AttDis** packet. It is because these sensors may affect the calculation of the target position. The sensor receiving the **AttDis** packet is the current visiting sensor. The part (L_{s^+} or L_{s^-}) as well as the location of the current visiting sensor will be recorded in the packet as well. While the current visiting sensor has different part with the part recorded in the **AttDis** packet, the current visiting sensor and the previous visited sensor are the attached sensors. As indicated in Fig. 2(a), the attached sensors are denoted s_l^a and s_r^a.

Once the two attached sensors are discovered, the region that the sensors may affect the calculation of the target position can be reduced to the dark gray area. Thus, the sensors recorded in **AttDis** packet, but not located in the dark gray area can be removed from the packet in order to shorten the length of the packet. Of course, if the following visiting sensors are located in the dark gray area, they all need being recorded in the **AttDis** packet. In this paper, those sensors which may affect the target position calculation are called *affecting sensors*. Finally, the **AttDis** packet will return back to s by RHR. By the data collected in the **AttDis** packet, s can calculate the target position accordingly.



(a) (b)

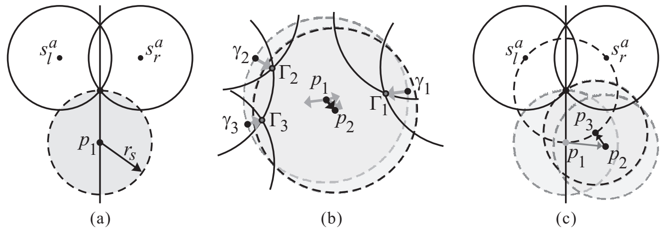
Fig. 2. Principles of an effective movement. (a) Attached Sensors Discovery. (b) Target position calculation.

It is worth noting that it is possible that L_s may cross the hole more than one place. PALM can also handle this situation correctly. However, the details will be omitted here due to space limitation. Based on the same reason, the reason why the sensors located in the dark gray area will affect the target position calculation is omitted as well. On the other hand, if s has less than two neighbors when it becomes to a risk sensor or does not receive the **AttDis** packet after a timeout, s will move alone L_s for a distance of r_c and redetermine whether it is still a risk sensor.

2) Target Position Calculation: Target position calculation is to find an appropriate position for a risk sensor to move. It should take both connectivity and coverage into consideration. By the previous procedure, the locations of the attached sensors as well as the affecting sensors have been collected to the risk sensor. As a result, the risk sensor can calculate a suitable position according to the *principles of an effective movement*.

The attached sensors found by the previous procedure are two sensors crossing the line passing through the risk sensor s and $\mathbb{S}$. If the risk sensor attaches to the attached sensors, it can make sure that the first principle of the *principles of an effective movement* is satisfied. According to [9], the best location to attach to the attached sensors for the risk sensor is the location p_1 shown in Fig. 3(a). As mentioned above, an affecting sensor is a sensor which will affect the calculation of the target position. In a formal description, an affecting sensor is a sensor whose sensing area has an overlap with $C(p_1, r_s)$, where $C(p_1, r_s)$ means a circle centered at p_1 with radius r_s, like the gray circle shown in Fig. 3(a).

It would be the best location if no affecting sensor exists. However, in most cases, the sensing areas of affecting sensors have overlaps with that of the risk sensor if it moves to p_1. Therefore, fine-tuning the position p_1 to reduce the overlap between the risk sensor and affecting sensors is necessary. As illustrated in Fig. 3(b), virtual forces of affecting sensors



(a) (b) (c)

Fig. 3. Target position calculation. (a) p_1, the best location to attach to the attached sensors. (b) p_2, the location to fine tune p_1 after taking affecting sensors into consideration in order not to cause much overlap between the risk sensor with the affecting sensors. (c) p_3, the final location, which is obtained by adjusting p_2 to reattach to the attached sensors.

to p_1 are used to adjust p_1 to p_2. The details are described as follows. For any two neighboring affective sensors, let the intersection point of the sensing circles of the two sensors be Γ. The ray $\overrightarrow{p_1\Gamma}$ intersects with $C(p_1, r_s)$ at γ. Thus, the force is defined as the vector $\overrightarrow{\gamma\Gamma}$ and is denoted $F_{\overrightarrow{\gamma\Gamma}}$.

The meaning behind the operation is as follows. As described above, the best way for a sensor to attach to two intersecting sensors and cause least overlap with the two sensors is to let the sensing circle of the sensor attach to the intersection point. Thus, the force $F_{\overrightarrow{\gamma\Gamma}}$ is to push $C(p_1, r_s)$ to attach to Γ. As a consequence, any two neighboring affecting sensors will generate a force to p_1. The summation of all forces will move p_1 to a new location, say p_2. In such a way, the overlap between $C(p_1, r_s)$ with the sensing areas of affecting sensors would be reduced accordingly. Fig. 3(b) is an illustrated example, where there are three forces exerted on p_1 to push it to p_2.

After changing p_1 to p_2, it is possible that the circle $C(p_2, r_s)$ loses the attachment to the attached sensors. Thus, it still needs to adjust p_2 to reattach to the intersection point of the sensing circles of the two attached sensors. The concept is illustrated in Fig. 3(c). Similarly, it operates like there is another force to push $C(p_2, r_s)$ to attach to the intersection point of the two attached sensors' sensing circles. As illustrated in Fig. 3(c), p_3 is the final target position for the risk sensor to move. The result of the target position calculation for the example shown in Fig. 2(a) is illustrated in Fig. 2(b). Accordingly, following the two procedures, attached sensor discovery and target position calculation, the *principles of an effective movement* can be satisfied.

IV. PERFORMANCE EVALUATIONS

A. Simulation Environment

To verify the effectiveness of PALM, a lot of experiments are performed by C++ language. In addition to PALM protocol, DSSA [1], DSSA+Sink, and VEC [2] are also simulated in the paper. DSSA+Sink is a modified version of DSSA. In DSSA+Sink, a sensor will add an extra force from the sink while calculating the sum of the forces from its neighbors. The interesting events are generated based on the random distribution.

The sensing field is $800m \times 800m$. A sink is located at left-down corner of the sensing filed, and 400 sensors are randomly deployed with uniform distribution. The energy consumption model of the sensor adopts the specification of MICA 2 [10]. In addition, the energy consumption of the sensor movement is $5.976J/m$ [11]. The energy consumed in communication and movement is counted in the simulation.

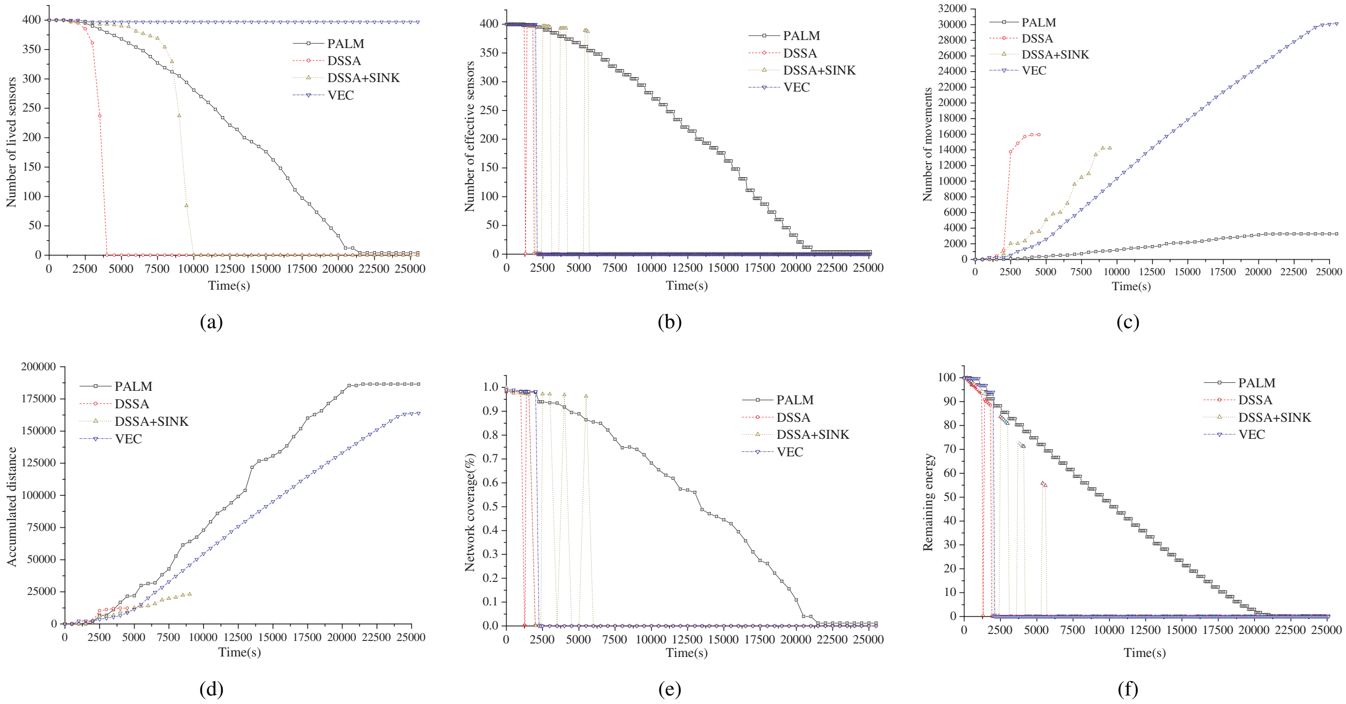
B. Simulation Results

Figs. 4(a) and 4(b) respectively show the total number of the lived and the effective sensors for all protocols. PALM can efficiently avoid network partition because almost all lived sensors are effective sensors. The network can keep connected until all sensors are malfunctioned. For DSSA and DSSA+Sink, the number of lived sensors decreases rapidly since all sensors move frequently (See Fig. 4(c)). Because the sensors near the sink have to relay sensing data, these sensors are prone to run out of energy. Moreover, these protocols can not recover from partitions quickly. Therefore, the network partition occurs frequently after sensor deployment. Due to the strict movement limitation, the number of lived sensors in VEC decreases slowly. However, the network is partitioned early due to slow network partition recovery (about 1250s).

Fig. 4(c) shows the total number of sensor movements during the simulation. Because of utilizing *lazy movement policy*, PALM has the fewest number of sensor movements. Moreover, the number of movements in DSSA+Sink is fewer than that in DSSA since the sensors in DSSA+Sink do not comparatively move to the sink back and forth. Because of the strict movement limitation, the number of sensor movements in VEC is fewer than those in DSSA and DSSA+Sink, but still higher than that in PALM. Therefore, we can conclude that PALM can avoid network partitions with fewer number of sensor movements. It is worth mentioning that, in PALM, a sensor's movement may result in the subsequent movements of the downstream sensors. However, each downstream sensor still needs to check whether the *lazy movement policy* is satisfied or not. Moreover, from the simulation result, the number of movements of PALM is still the least.

Fig. 4(d) demonstrates the accumulated distance of the sensor movements for different protocols. PALM has the longest moving distance comparing to VEC, DSSA and DSSA+Sink. Since the number of movements of PALM is the fewest among all protocols, it implies that the distance of each movement in PALM is longer than those by the others. It also means that the movement in PALM is more efficient than the others.

Fig. 4(e) illustrates the quality of network coverage for different approaches. By the *lazy movement policy* and the *principles of an effective movement* in PALM, the network coverage can keep a certain quality for a long time. However, the network coverage of the other protocols are very unstable. It depends on whether the sensors can connect to the sink. The network coverage will promptly decrease because of the network partitions, and promptly increase because of the recovery from network partitions.



(a) (b) (c)

(d) (e) (f)

Fig. 4. Simulation results. (a) Number of effective sensors. (b) Number of lived sensors. (c) Number of movements. (d) Accumulated distance of movements. (e) Network coverage. (f) Remaining energy of the effective sensors.

Fig. 4(f) shows the remaining energy of the effective sensors for different protocols. Similar to Figs. 4(b) and 4(e), the curve of PALM slowly and smoothly decreases. This means that the sensing tasks can be continually performed well until all sensor are malfunctioned.

V. CONCLUSIONS

In the paper, we propose a distributed protocol, called PALM, which mainly employs mobile sensors for network partition avoidance. Basically, PALM applies the proposed *lazy movement policy* and *principles of an effective movement* to not only decrease energy consumption of the sensor but also increase effective coverage. PALM favors the lived sensor to become the effective one so as to increase effective coverage. In addition, PALM also enables a sensor to self-determine the time it should start to move and the place where it should move. Simulation results show that PALM can always keep the network connected and incur fewer number of movements, less coverage overlap, and less energy consumption in comparison with related protocols. Obviously, by means of PALM, sensors are capable of delivering data to the sink, so that reliable sensing quality is guaranteed. Our further work is to experiment PALM on the hybrid sensor network, comprising the mobile and stationary sensors.

REFERENCES

[1] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, vol. 35, no. 1, pp. 78–92, Jan. 2005.

[2] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," in *Proceedings of the IEEE INFOCOM, the Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, Mar. 2004, pp. 2469–2479.

[3] J. Wu and S. Yang, "SMART: A scan-based movement assisted sensor deployment method in wireless sensor networks," in *Proceedings of the IEEE INFOCOM, the Annual Joint Conference of the IEEE Computer and Communications Societies*, Mar. 2005.

[4] S. Ganeriwal, A. Kansal, and M. Srivastava, "Self aware actuation for fault repair in sensor networks," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation(ICRA)*, May 2004, pp. 5244–5249.

[5] G. Wang, G. Cao, T. L. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *Proceedings of the IEEE INFOCOM, the Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, Mar. 2005, pp. 2302–2312.

[6] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guarnteed delivery in ad hoc wireless networks," in *Proceedings of the International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM'99)*, 1999, pp. 48–55.

[7] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the ACM International Conference on Mobile Computing and Networking (MOBICOM)*, Aug. 2000, pp. 243–254.

[8] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing routing holes in sensor networks," in *Proceedings of the IEEE INFOCOM, the Annual Joint Conference of the IEEE Computer and Communications Societies*, Mar. 2004.

[9] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," in *NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, 2004.

[10] Crossbow. (2003) MPR/MIB Mote Hardware Users Manual. Crossbow website. [Online]. Available: http://www.xbow.com/Support/manuals.htm

[11] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. S. G. Lee, "Energy-efficient motion planning for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 5, May 2004, pp. 4344–4349.