

Research Article

Ion-6: A Positionless Self-Deploying Method for Wireless Sensor Networks

Shih-Chang Huang

Department of Computer Science and Information Engineering, National Formosa University, YunLin, Taiwan

Correspondence should be addressed to Shih-Chang Huang, schuang@nfu.edu.tw

Received 15 June 2011; Accepted 27 July 2011

Academic Editor: Yuhang Yang

Copyright © 2012 Shih-Chang Huang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sensor networks are deployed to monitor the interested area. Maximizing the sensing coverage depends on effectively determining the locations of sensors. In this paper, a self-deploying method named as Ion-6 is proposed. Sensors are modeled as ions, and the links between them are treated as ionic bonds. When the number of ionic bonds of a sensor is full, the sensor will expel others out of its field. Sensors organize themselves as the hexagonal format to maximize the network's coverage area, retain the network connectivity, and prevent from introducing the coverage holes. The sensors in the proposed method can compute their moving directions and distances independently without priori position information. Simulation results show that Ion-6 can organize sensors as the hexagonal format to maximize the coverage area. The deploying time of Ion-6 is less than the molecule model and efficiently eliminates the unnecessary movements of the virtual force (V-force) method.

1. Introduction

A sensor network consists of a large number of tiny devices, which can detect the variances of environment, execute simple computations, and use wireless communication to exchange information. Those devices are deployed over the interested regions to collect the environment information and help people to monitor the incidents.

Sensors can be deployed factitiously [1–6] or be randomly spread over the interested regions. However, in the applications to monitor the harsh terrain or the hostile environments, it is almost impossible to deploy sensors by human beings, for example, detect the radiation zone, monitor the noxious gas leaking area, trace the frontline battlefield, and explore unknown planets.

Spreading the sensors randomly cannot guarantee the precise settle-down location of each sensor. Sensors may cluster overly in a small region or may distribute too sparsely to retain the network connectivity. Those two possible unfavorable results cause the sensors losing the efficient surveillance on the environment. Therefore, the sensors with the mobile ability are used to adjust their positions after randomly spreading.

The deploying problem will become trivial if each sensor can obtain both the mobile ability and its position information [7–10]. Each sensor's final location can be scheduled to maximize the coverage area and minimize the moving distance. However, when the position information is not available, getting well coverage area and shortening the moving distance during the self-deploying will become a great challenge. Thus, some of the previous researches organize sensors into the cluster architecture [11–13]. A small set of mobile sensors are selected as the local controllers to determine the others' locations. In this architecture, the sensor-deploying problem is simplified to organize sensors in every cluster. The critical issue of this architecture is that it has the reliability problem when the cluster heads out of function.

Some previous works model the mobile sensors as the electrons [14, 15] or molecules [16, 17] to avoid the fault of cluster architecture. The received signal strength (RSS) of this message is treated as the force which pushes each other. The deploying procedure finishes when the forces work on every sensor are balanced. Sensors in this model may have *oscillation moving* that sensors move back and forth over a small region to adjust their positions before the force

becomes balance. It is not energy efficiency for the energy-limited sensors.

In this paper, we model the deploying problem as building the ionic bonds between ions. Sensors are ions, and the built links between them are the ionic bonds. Sensors do not need to have their position information. They only require the abilities to identify the direction of incoming signals and accurately estimate their distance to the other neighbors. These are two essential abilities in general self-deploying methods. The proposed method can effectively maximize the coverage and minimize the deploying time.

The rest of this paper is organized as follows. The related positioning systems are reviewed in Section 2. The proposed method is given in Section 3. In addition, this section also discusses how to adjust the deploying location when sensors do not move to their proper locations. In Section 4, we give the simulation results to prove that the proposed Ion-6 method works well. Finally, the paper conclusions are in Section 5.

2. Related Works

For the mobile sensors, the self-deploying methods have two classes. The first class is position-based methods which mobile sensors have the prior knowledge of their location information [7–10]. When sensors can get their position information, their positions can be prescheduled before they are cast over the interested region. The average moving distance of sensors can be minimized, and the coverage area can be maximized. Therefore, the object of position-based methods is to maximize the coverage area [8], minimize the moving distance [9], and eliminate the coverage holes [18, 19]. Jourdan et al. use the genetic algorithm to solve the deployment optimization problem [19]. To speed up the computational convergence of the method in [20], Xiaoling et al. use the particle swarm optimization (PSO) problem to model the self-deploying problem. Sensors are restricted to move in the limited region to save energy consumed during the sensor moving process. A similar method is proposed in [7] but involves the obstacles in the interested region. These methods divide the interested region into multiple regular grid areas. In [10], Li et al. considered the distance and orientation of every sensor to find its deploying location. The position information is necessary for this method to organize the sensors as a hexagon format.

The second class is positionless that mobile sensors do not have their position information. The simplest positionless method is organizing sensors into multiple clusters [11]. In each cluster, a mobile sensor takes charge to determine the others' locations. This mobile sensor is named *cluster header*. Sensors in each cluster will organize as the star topology that the cluster head is the center. Due to the fact that geography may greatly increase the difficulty on placing and electing the cluster heads, determining the cluster heads is the major challenge in this method. In additional, when cluster head crashes, a recovery mechanism [13, 14] is needed to retain the reliability of network.

Another positionless method is to model the mobile sensors as the molecules [12, 16, 17]. Each sensor has to

broadcast a dummy message periodically to announce its existence. By collecting the dummy message, each sensor can know how many neighbors it has. The number of neighbors implies the density of its neighborhood. Sensors will move step by step from high-density area to a low-density one. The moving direction can be computed from the direction of the incoming signal. Because the sensors use small step to adjust their next position, the deployment process usually consumes a lot of time. Mobile sensors also waste much additional energy on exchanging messages.

To deploy the sensors quickly, sensors are modeled as the electrons [5, 14, 15]. This model is called the Virtual Force (V-force) method. Similar to the molecule model, sensors have to send the dummy message. The receivers will transform the signal strength of the incoming message as the force. When the received message gives strong signal strength, a large force is created; otherwise, the force is small. The receiver computes the net forces and determines its moving direction and distance. Sensors continuously move until the value of net forces is less than the threshold. Although this method removes the reliability problem of the cluster architecture, it has a severe problem on redundant movement, which is caused by the oscillation moving.

To reduce the energy consumed on oscillation moving and to maximize the coverage area in the positionless methods, we proposed a novel method which models the sensors as ions. Sensors in the proposed method can expand their coverage area better than both molecule model and the V-Force method. Besides, sensors can be deployed rapidly than the molecule model and can save much energy wasted on the redundant movement than the V-Force method. The detail is given in next section.

3. The Ion-6 Method

3.1. Preliminaries and Assumptions. Each sensor has the following characteristics which are the common characteristics in current self-deploying methods.

- (1) Each sensor has a unique identity. Sensors are randomly spread over the interested region.
- (2) Each sensor can communicate with others without losing data.
- (3) All sensors have the same communication ranges. The coverage area of each sensor is a circular disk. The sensing range is equal to the communication range.
- (4) Sensors can precisely estimate the Euclid's distance to the sender from the received signal strength of incoming packets. The Friis transmission formula [21] is used to convert the RSS to Euclid's distance. In the later, we will consider inaccurate estimation and show how sensors detect the fault and adjust their positions.
- (5) Each sensor installs a precise antenna array, which can identify the angle of every incoming packet. Each sensor also has a precise compass to determine its moving direction.

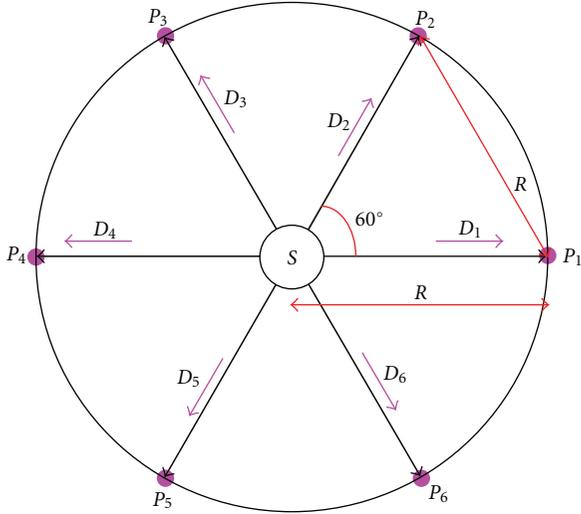


FIGURE 1: The six ionic bonds and stable slots of sensor S.

We have the following additional assumptions for our proposed method. The maximal number of neighbors which a sensor can have is defined as the number of *ionic bonds*. To organize the deploying topology as the hexagonal format, the ionic bonds of every sensor are set to six. The first sensor which starts the deploying procedure will define the direction of each ionic bond. The direction of these six ionic bonds will evenly divide the circular communication coverage into six same-sized sectors shown as Figure 1. All sensors follow the first sensor's decision and propagate this decision to others during the deploying procedure.

For each sensor, if there is a neighbor at the direction of its ionic bond I_i , denoted as D_i , and the distance between the neighbor and itself is equal to the sensing radius R , the ionic bond I_i is defined as a *stable ionic bond*. The location of this neighbor is called as the *stable slot* of ionic bond I_i , denoted as P_i . As Figure 1 shows, the directions of the six ionic bonds of sensor S are D_1, D_2, \dots, D_6 . The corresponding stable slots are P_1, P_2, \dots, P_6 . If a stable slot has obstacles in it such that the assigned neighbor cannot move to there, the sensor will remove this ionic bond.

3.2. The Ion-6 Method. Initially, all sensors are in passive mode waiting for combining with others. Because sensors have free ionic bonds, their state is *unsteady*. A random sensor S enters the active mode and starts the deploying procedure. Sensor S is called as *anchor*. The anchor S sets the default directions of the six ionic bonds and broadcasts to all neighbors via a *bond packet*. The directions of these ionic bonds are represented as unit vectors.

Let W be an unsteady sensor that can directly receive the bond packet from S , and its distance to S is $|SW|$. Sensor W uses the Frii's transmission formula [21] to compute the $|SW|$ from the RSS. We assume that $|SW| = d$. In addition, the incoming direction of the bond packet is \vec{V}_W shown as Figure 2. For each free ionic bond I_i in the bond packet,

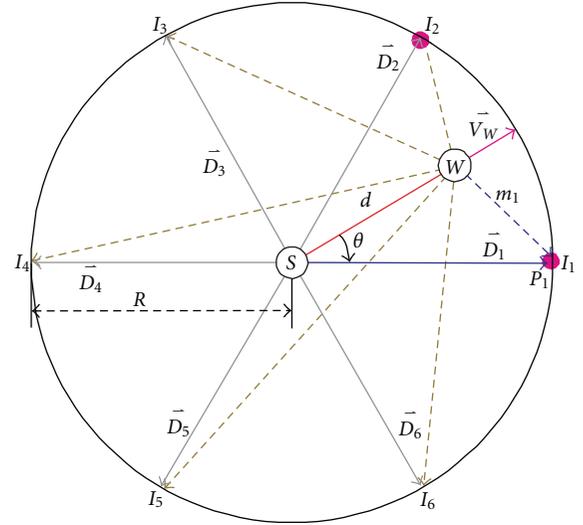


FIGURE 2: The directions of the six ionic bonds.

sensor W computes the distance m_i and direction \vec{U}_i to the corresponding stable slot P_i . Then, sensor W reports the results to S . Here, we denote the distance to stable slot P_i as m_i and the direction as \vec{U}_i .

By having the communication radius R and the distance $|SW| = d$, we can use the cosine law to compute the m_i

$$m_i = \sqrt{d^2 + R^2 - 2dR \cos \theta_i}. \quad (1)$$

The angle θ_i is the included angle of \vec{V}_W and \vec{D}_i . It can be obtained from the inner product of \vec{V}_W and \vec{D}_i

$$\theta_i = \cos^{-1} \left(\frac{\|\vec{V}_W \cdot \vec{D}_i\|}{\|\vec{V}_W\| \|\vec{D}_i\|} \right). \quad (2)$$

The moving direction \vec{U}_i can be computed from

$$\vec{U}_i = R \times \vec{D}_i - d \times \vec{V}_W. \quad (3)$$

After collecting the results from all neighbor sensors, anchor S instructs the sensor with minimal m_i to move to each P_i . These instructed sensors will change to active mode. We called them as *candidates*. After the candidates move to the stable slots, they park at the locations and notify S . Note that if a P_i has already been occupied by a sensor, S will not assign a candidate again. When all the ionic bonds become stable, S will expel all passive mode sensors out its sensing field. When S finished expelling the passive mode sensors, it changes to *lock* state. The lock state sensors will no longer move. All candidates become anchors and are notified to find its candidates. To prevent two adjacent anchor sensors from arranging two different candidates to their joint stable slots, only one anchor sensor is allowed to broadcast the bond packet at a time. The other anchor sensor must hold down

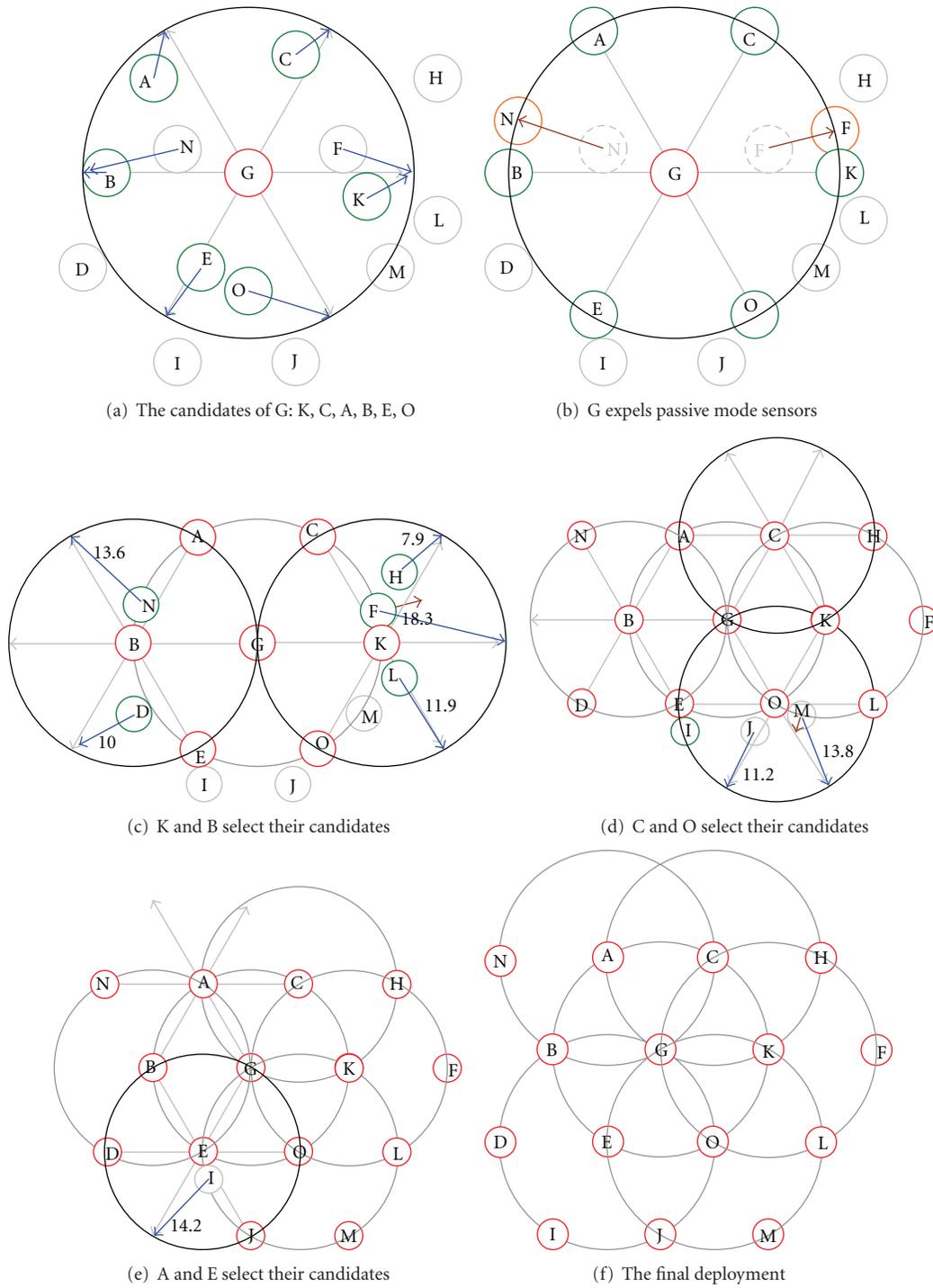


FIGURE 3: Deploying example.

until the candidate sensors of current anchor sensor move to their stable slots.

Furthermore, to prevent the expelled passive sensors from moving back to the sensing range of the lock state sensors, a level architecture is also built while determining the candidates. The first active sensor sets its level to 0. The candidates selected by active sensor X will set their level to $L_x + 1$, where L_x is the level of X . The active sensors in level

$N + 1$ can start to broadcast their bond packets when all active sensors in level N have found or not been able to find the candidate sensors. The deploying procedure stops until all sensors become the lock state.

Figure 3 gives a little example of the proposed method. Initially, sensors are deployed as Figure 3(a). The sensor G enters the active mode to broadcast the bond packet. G sets its level to 1 and determines the directions of the six ionic

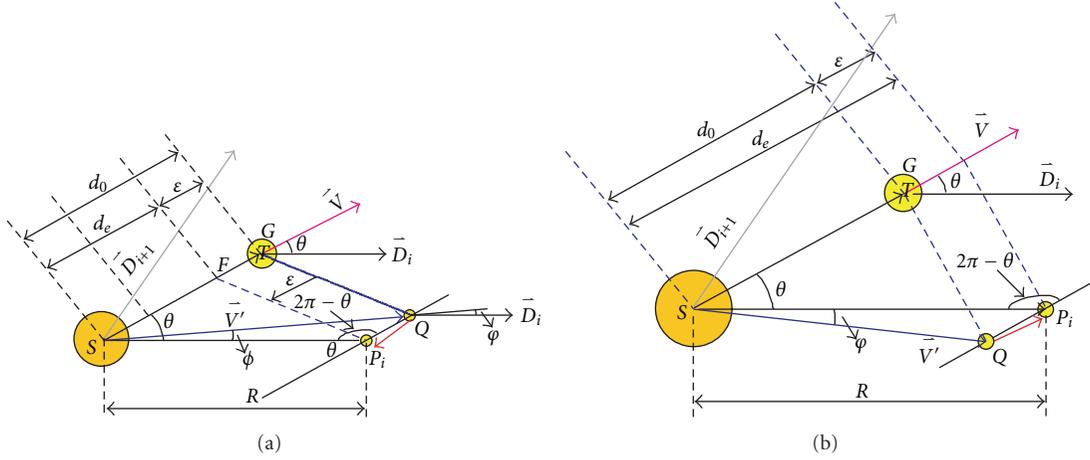


FIGURE 4: Distance adjusting mechanism.

bonds. The sensors $\{A, B, C, E, F, K, N, O\}$ receive the bond packet from G. They compute the m_i and \vec{U}_i to each stable slot of G. Sensor G selects the set $\Omega = \{K, C, A, B, E, O\}$ as its candidates after it collects the results from neighbors. The level information is also propagated to them. Sensors in set Ω set their level to 2, change to active mode, and start to move to the corresponding stable slots. When these six candidates reach the target locations, those passive mode sensors in G's sensing field will be expelled by G. The sensors N and F are expelled shown as Figure 3(b).

After G expels all passive mode sensors, it acknowledges all level 2 sensors to select their candidates. K and B are the first two sensors selected to choose their candidates shown as Figure 3(c). The level 2 candidates A, C, E, and O hold down their attempts to select candidates because the transmissions of K or B block their bond packets. Sensor K chooses H, F, and L as its candidates. Sensor B only chooses N and D because there are no other sensors. The sensors H, F, L, N, and D are set to level 3. When all the six ionic bonds of K become stable, sensor K starts to expel the passive mode sensor M out of its sensing coverage. So is the sensor B. The other level 2 candidates C, O and A, E are similar to K and B. Their results are shown as Figures 3(d) and 3(e).

After all sensors in level 2 have chosen their candidates, the sensors in level 3 will be notified to select their candidates. In this example, all level 3 sensors have no passive sensors in their sensing fields. Therefore, the deploying procedure terminates. Figure 3(f) shows the deploying results.

3.3. Adjusting the Location to Stable Slot. The accuracy distance from a sensor to each stable slot of the anchor is necessary for the Ion-6 method. However, the estimated distance may be shorter or longer than the real one in the practice case. In this section, we discuss the position adjusting mechanism to recover from inaccuracy distance estimation.

Let d_0 be the real distance between S and one of its candidate T, and d_e is the estimated distance where $d_0 > d_e$, shown as Figure 4(a). Before the candidate T moves, its position is at location G. However, candidate T thinks its location is at F because of the inaccuracy estimated distance. Candidate T uses the radius R, d_e , and includes angle of the vector \vec{V} and \vec{D}_i to compute the moving distance m_i | $m_i = |\vec{FP}| = |\vec{GQ}|$ and the moving direction $\vec{U}_i = |\vec{GQ}|$ to the stable slot P_i . By following the computed m_i and \vec{U}_i , the location of candidate T is at Q instead of stable slot P_i .

When candidate T reaches location Q, the angle of the incoming signal from S, \vec{V}' , is not aligned with \vec{D}_i . It indicates that candidate T has not moved to the stable slot P_i yet. The candidate T should adjust its location by moving toward the direction \vec{QP}_i with distance $\epsilon = |d_0 - d_e|$.

Let $\angle QSP_i$ be ϕ which can be computed from the inner product of \vec{V}' and \vec{D}_i . Because \vec{SG} is parallel to \vec{QP}_i , the $\angle SQP_i$ will be $\theta - \phi$. And the ϵ can be calculated from

$$\epsilon = \frac{R}{\sin(\theta - \phi)} \times \sin \phi. \quad (4)$$

Furthermore, the adjusting direction \vec{QP}_i is $\vec{D}_i - \vec{V}'$. Due to the \vec{SG} and \vec{QP}_i are parallel, $\vec{QP}_i = -\vec{V}$.

For the case $d_0 < d_e$, shown as Figure 4(b), the position adjusting procedure is similar. The adjusting direction \vec{P}_iQ is \vec{V} . The difference is the angle for the sine law changes as the following:

$$\epsilon = \frac{R}{\sin \theta} \times \sin \phi. \quad (5)$$

To determine $d_0 > d_e$ or $d_0 < d_e$, we need to compare the angle $\angle GSP_i$ and $\angle GSQ$. If $\angle GSP_i > \angle GSQ$, (4) is used; otherwise, (5) is used. The algorithm for the proposed deploying method is given in Algorithm 1.

Ω_q : the set of one-hop neighbors of sensor q .
 I_j : the j th ionic bond of a sensor q , where $j = 1, \dots, 6$.
 D_j : the direction of the j th ionic bond of a sensor q .
 P_j^z : the location of stable slot of the j th ionic bond of a sensor q .
 H_K : the location of sensor K .
 $L(S)$: the level of sensor S . The initial level of all sensors is infinite.
 G_i : the group of sensors in level i .
 V_S : incoming direction of received signal from sensor S .

- (1) Give a sensor S , sets its level $L(S)$ to 0.
 Sensor S determines the D_j of each I_j . D_j is a unit vector.
 Set $i = 0$.
- (2) If G_i is empty
- (3) Exit.
- (4) Else
- (5) For each sensor Z in G_i do {
- (6) Sensor Z broadcasts the D_j to sensor $X \mid X \in \Omega_Z$.
- (7) X computes the length and direction of $H_X P_j^z \mid j = 1, \dots, 6$.
- (8) Return the computed results to Z .
- (9) For each $P_j^z \mid j = 1, \dots, 6$, {
- (10) Z selects the sensor X_j' with $\min |H_{X_j'} P_j^z|$ as the candidate to the location P_j^z .
- (11) Skip the P_j^z , if {
- (12) (a) There is no selectable sensor in Ω_Z
- (13) (b) The location has occupied by a sensor not in passive mode
- (14) }
- (15) Sensor X_j' sets its level to $L(X_j') = L(Z) + 1$, moves toward the direction $H_{X_j'} P_j^z$.
- (16) X_j' compares the V_Z and D_j when X_j' stops moving.
- (17) If $V_S \neq D_j$, X_j' adjusts its location until $V_S = D_j$.
- (18) Else X_j' notifies the Z that it reaches location P_j^z .
- (19) }
- (20) }
- (21) Sender Z expels the passive mode sensors when all six X_j' are ready.
- (22) Set $i = i + 1$. Redo step (2) to (21).

ALGORITHM 1: Algorithm of the proposed deploying approach.

4. Simulation Results

This section shows the simulation results. A C++ program is developed to evaluate the proposed Ion-6 method. We also implement the molecule model [16] and the V-force method [14] for comparison. The evaluation items include the *coverage*, *link density*, *effective move ratio (EMR)*, *number of movements*, and *deploying time*.

The *Coverage* is to evaluate the area detecting utilization of a deploying method. When the number of sensors is fixed, the deploying method that can maximize the coverage area will have a higher area detecting rate. In order to return the sensing date back to the collector, an additional constraint on maximizing the coverage area is retaining the network connectivity. Our evaluation will consider maximizing the coverage and retaining the network connectivity simultaneously.

The *link density* evaluates the number of one-hop neighbors of each sensor. High-density deploying topology consolidates the network connectivity but sacrifices the sensing coverage. Low link density deploying one is just in contrary and may also generate coverage holes. The best deploying structure is to organize the sensors in hexagon

structure [15]. We can evaluate the number of linked neighbors to verify whether sensors are properly deployed.

The *effective move ratio (EMR)* of a sensor is defined as d_l/d_r , where d_l is the distance directly lines from initial location to its final location and d_r is the sensors' total moving distance from initial location to its final location. The optimal EMR is 1 that a sensor does not have redundant moving distance during the deploying procedure. High EMR implies that the sensor spends a lot of energy on useless moving. It can be used to estimate whether the deploying method is energy efficiency and whether the deploying method has serious oscillation moving problem. The simulation results will show the average EMR of all deployed sensors.

The *number of moving instructions* accumulates the number of received moving commands until a sensor becomes stationary. During the deploying procedure, a sensor will receive a moving instruction before it moves to its next location. Therefore, the communication overhead of a deploying method can also be reflected from the number of moving instructions. The more instructions are issued, the higher communication overhead a deploying method will be.

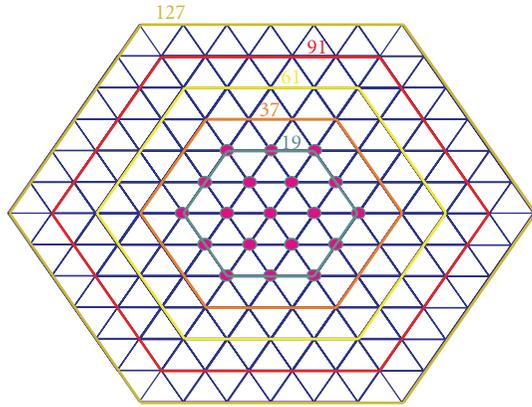


FIGURE 5: Perfect hexagon format.

The deploying time is the time period that the sensors require to finish the deployment. Short deploying time implies that the method can effectively direct each sensor moving to proper location. Sensor can quickly start the environment monitoring task. A good deploying time should shorten the deploying time.

4.1. Environment Setup. Initially, all deployed mobile sensors are randomly cast within a 100×100 area which is in the center of the interested area. Each sensor can identify its moving direction and the directions of the incoming signals. All sensors uniformly set their sensing range to 50 meters. The simulated network scale includes 19, 37, 61, 91, and 127 sensors. The number of sensors in each network scale can be organized as the perfect hexagon format shown as Figure 5. Therefore, we can draw out their optimal deploying topology for comparison.

In our simulation, the time spent for a sensor to change position is only proportional to its moving distance. The speed and geography factors are not considered. For estimating the deploying time, each sensor's moving speed is uniformly set to 2 meters/sec. Negotiating one deploying message between sensors requires 0.2 sec. The deploying procedure completes when all sensors become stationary. The simulation results are averaged from 500 random tests.

4.2. Numerical Results. Figure 6 compares the coverage area. The Ion-6 method has better coverage results than other methods in each network scale. The Ion-6 method exhibits well coverage results in large network scale. Except the network scale of 19 sensors, the Ion-6 method can have more 15% additional coverage area than the molecule model.

The V-force method has the worst coverage in our simulation. The V-force method will quickly expand the coverage area when network density is high. After a period of expanding, the network becomes sparse. The forces contributed by the neighbors will become weak. Sensors at the outer peripheral of network will stop being pushed even the sensors in the inner network still suffer great balanced forces. Therefore, the V-force's coverage area is

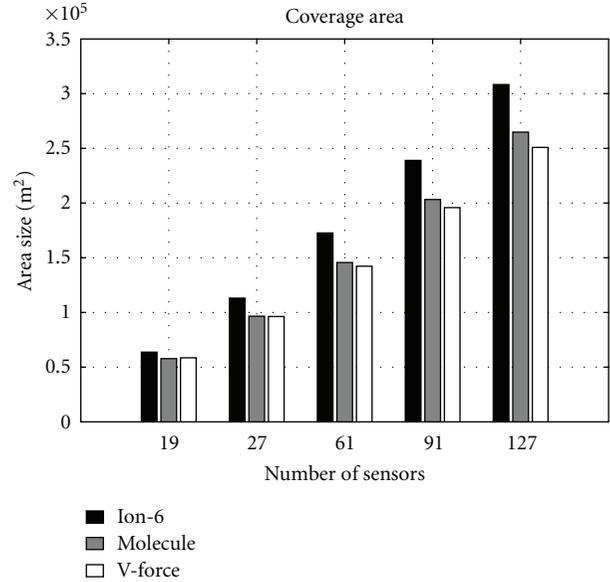


FIGURE 6: The coverage size of deploying methods.

slightly less than the molecule model. This phenomenon gradually becomes explicitly when the network scale grows.

Figure 7 shows the link density distribution of sensors in different network scale 19, 61, and 127. The optimal curves in these figures are the link density generated by the perfect hexagon topologies shown as Figure 5. The link density distribution of the Ion-6 method in each network scale is closer to the optimal one than other methods. The sensors with six neighbors dominate the major percentage. There is no sensor that has neighbors more than 6 and no sensors with a single neighbor. Similar to the optimal one, the second major percentage is four neighbors.

The first and second major percentage in V-force method and the Molecule model are 5 and 6. Because sensors have high link density, their coverage areas become small. In Figure 7, there is no sensor whose link density is zero. It implies that the Ion-6 can deploy sensors closer to the results of the optimal case.

Figure 8 displays the effective moving ratio. In the molecule model, sensors always move a small-step to adjust the moving direction. The average EMR of a sensor ranges from 1.3 to 1.6. In order to speed up the deploying of the molecule model, the sensors in the V-force method adapt a large move distance to expand quickly. When the network scale is small, each sensor's EMR is slightly more than the molecule model. However, when the network scale is more than 91, sensor's EMR rapidly increases because of the oscillation moving. The EMR in the network scale of 127 sensors is 2. It means that the redundant moving distance of every sensor is double.

By computing and selecting the suitable candidates, the Ion-6 method almost introduces zero redundant moving distance when network scale is 19. When the network scale is 127, the EMR of Ion-6 method is still less than 1.2. It also implies that Ion-6 method causes less redundant moving.

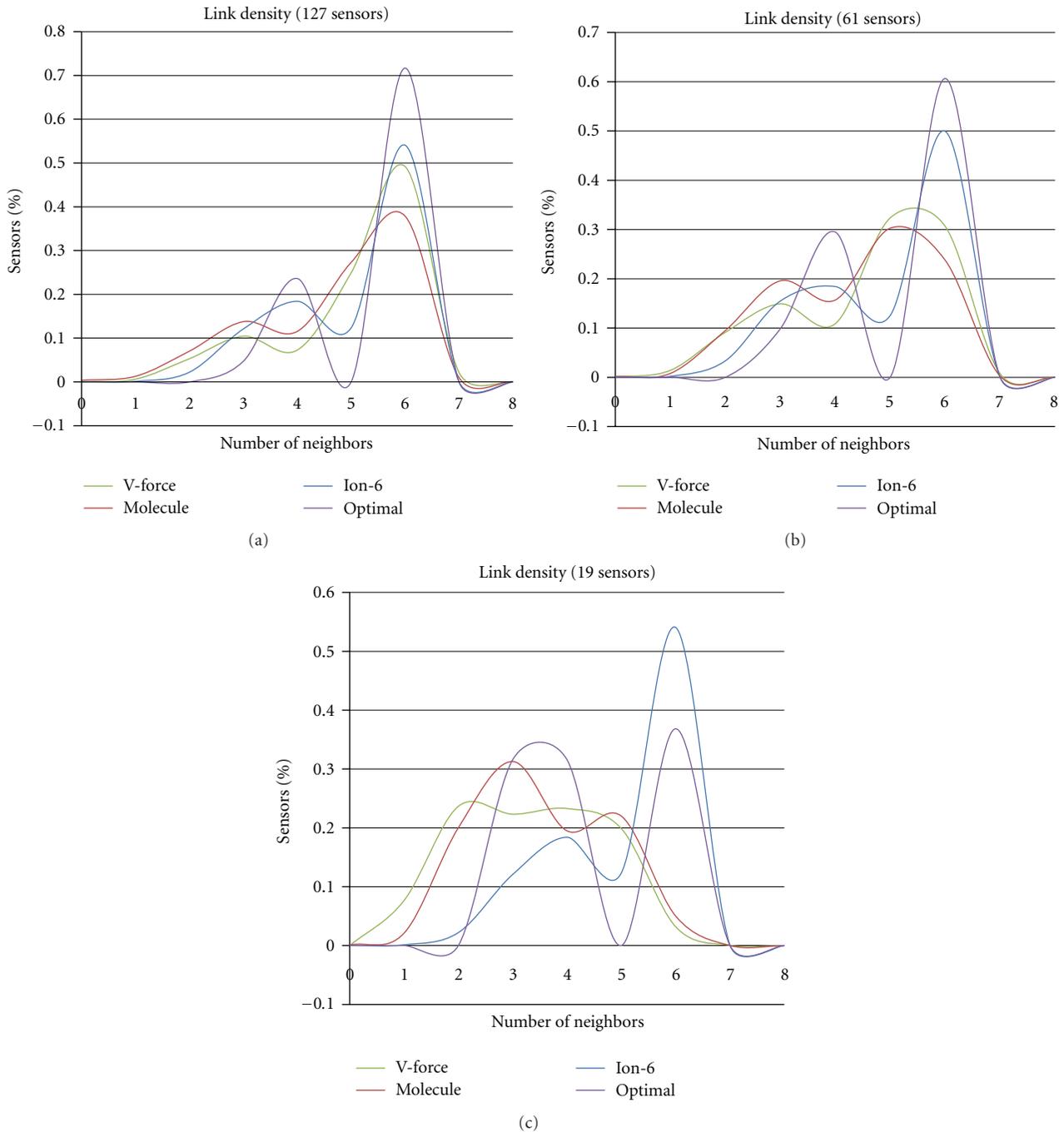


FIGURE 7: The link density (the distribution of the number of neighbors).

Figure 9 shows the average number of issued moving instructions of a sensor. In the molecule model, the small step adjusting strategy spends more communication messages to negotiate the moving direction with others. Therefore, when the network scale grows, the number of issued instructions rapidly increases. For the network scale of 127 sensors, each sensor has to send more than 170 messages. The communication overhead is heavy.

In contrary, sensors in the V-force method always try to move a large distance when they receive a moving

instruction. It can effectively reduce the number of issued moving instructions. Therefore, the average number of issued instructions of the V-force method in each network scale is less than half the number of molecule model.

The Ion-6 exhibits an outstanding result in this evaluation term. Sensors issue the moving instruction only when they are selected as candidates and ready to expel others. After a sensor has been selected as a candidate and expels others, it will no longer have to issue the moving instruction. Therefore, the Ion-6 method can effectively minimize the

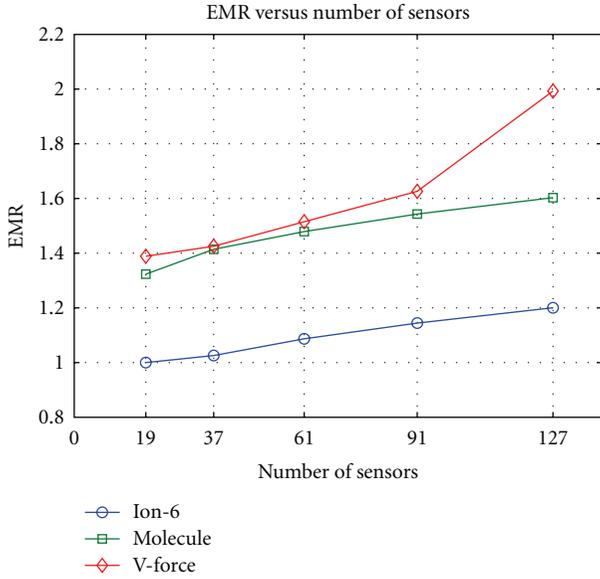


FIGURE 8: The effective move ratio (EMR).

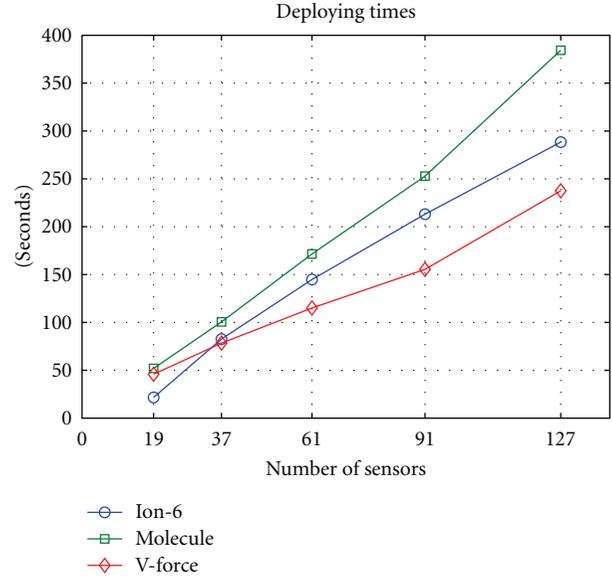


FIGURE 10: Deploying times.

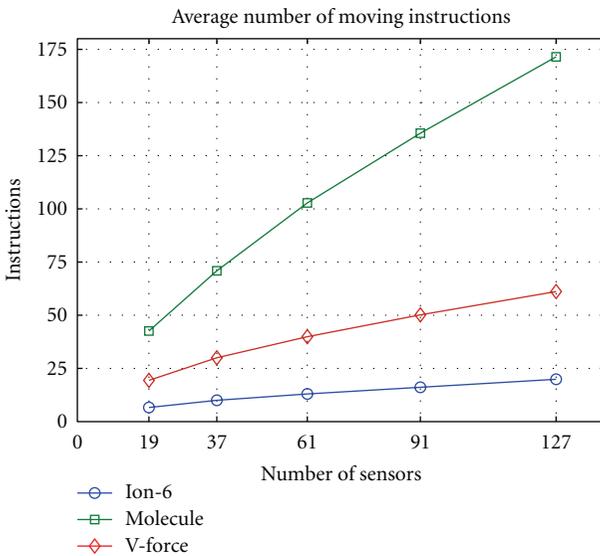


FIGURE 9: The number of moving instructions.

number of issued moving instructions. Even the network scale of 127 sensors in our simulation, the number of issued moving instructions of Ion-6 method is still less than 20.

Figure 10 shows the time to complete the deployment. Sensors in the molecule model use small moving step to adjust their position. When the number of sensors grows, time to complete the deployment increases rapidly. In the contrary, sensors in the V-force method use large moving step to adjust their positions when sensors are crowded. The adjusting step gradually shrinks when sensors spread out. By effectively adjusting the moving step, the V-force method can decrease the deploying time.

When the deploying procedure can be finished in one or two moving instructions, the deploying time of Ion-6 method is better than the V-force method, for example, the

scenario of network scale 19. When network scale is more than 37, the deploying time of Ion-6 gradually increases. In the network scale of 91 and 127, the deploying time of Ion-6 method is more than the one of V-force method about 50 seconds. However, Ion-6 is still better than the molecule model.

5. Conclusions

In this paper, we propose the Ion-6 self-deploying method which models the sensors as the ions. Sensors can compute their moving directions and distances independently. The deploying problem is to build ionic bonds between sensors. To organize the sensors as the hexagonal cellular topology, the number of ionic bonds of each sensor is set to six. The sensors selected as candidates are instructed to move. The other sensors will be expelled outside the sensing areas of the candidates. A location adjusting mechanism is also proposed to fix the fault caused by inaccurate estimating distance of the TDOA technique.

Simulation results prove that the proposed Ion-6 method can maximize the coverage and achieve the near-optimal hexagon topology without the explicitly position information. It can also efficiently control the movement of each sensor to minimize the oscillation moving problem in the V-force method. Furthermore, the Ion-6 method can reduce the communication overhead and deploying time in the molecule model.

Acknowledgment

This work was supported in part by the National Science Council, Taiwan, under Grant NSC100-2221-E-150-070.

References

- [1] Y. R. Tsai and Y. J. Tsai, "Sub-optimal step-by-step node deployment algorithm for user localization in wireless sensor networks," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC '08)*, pp. 114–121, Jun 2008.
- [2] Z. H. Yuan and G. F. Wang, "Sensor deployment strategy for collaborative target detection with guaranteed accuracy," in *Proceedings of the 4th International Conference on Mobile Ad-hoc and Sensor Networks (MSN '08)*, pp. 68–71, Dec 2008.
- [3] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," in *Proceedings of the 5th International Conference on Information Fusion*, vol. 2, pp. 1581–1587, July 2002.
- [4] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor network," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 3, pp. 1380–1387, April 2001.
- [5] D. O. Popa, H. E. Stephanou, C. Helm, and A. C. Sanderson, "Robotic deployment of sensor networks using potential fields," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, vol. 1, pp. 642–647, month year.
- [6] A. Howard, M. J. Mataric, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots*, vol. 13, no. 2, pp. 113–126, 2002.
- [7] N. Bartolini, T. Calamoneri, T. L. Porta, and S. Silvestri, "Mobile sensor deployment in unknown fields," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 1–5, 2010.
- [8] X. Z. Bai, L. Shu, C. J. Jiang, and Z. Z. Gao, "Coverage optimization in wireless mobile sensor networks," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom '09)*, pp. 1–4, 2009.
- [9] M. Tariq, Z. Zhou, Y. J. Park, and T. Sato, "Diffusion based self-deployment algorithm for mobile sensor networks," in *Proceedings of the IEEE Vehicular Technology Conference (VTC '10)*, pp. 1–5, 2010.
- [10] J. Li, L. Cui, and B. Zhang, "Self-deployment by distance and orientation control for mobile sensor networks," in *Proceedings of the International Conference on Networking, Sensing and Control (ICNSC '10)*, pp. 549–553, 2010.
- [11] K. H. Tan and M. A. Lewis, "Virtual structures for high-precision cooperative mobile robotic control," *Autonomous Robots*, vol. 4, pp. 387–403, 1997.
- [12] Y. Zou and C. Krishnendu, "Sensor deployment and target localization based on virtual forces," in *Proceedings of the IEEE 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 2, pp. 1293–1303, March-April 2003.
- [13] J. Lee, A. D. Dharne, and S. Jayasuriya, "Potential field based hierarchical structure for mobile sensor network deployment," in *Proceedings of the American Control Conference (ACC '07)*, pp. 5946–5951, Jul 2007.
- [14] X. Wu, L. Shu, M. Meng, J. Cho, and S. Lee, "Coverage-driven self-deployment for cluster based mobile sensor networks," in *Proceedings of the IEEE International Conference on Computer and Information Technology (CIT '06)*, p. 226, Sep 2006.
- [15] N. Heo and P. K. Varshney, "An intelligent deployment and clustering algorithm for a distributed mobile sensor network," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics '03*, vol. 5, pp. 4576–4581, Oct 2003.
- [16] R. S. Chang and S. H. Wang, "Self-deployment by density control in sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1745–1755, 2008.
- [17] M. R. Pac, A. M. Erkmén, and I. Erkmén, "Scalable self-deployment of mobile sensor networks: a fluid dynamics approach," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems '06*, pp. 1446–1451, Oct 2006.
- [18] C. Fang and C. P. Low, "A unified framework for movement-assisted sensor deployment," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08)*, pp. 2057–2062, 2008.
- [19] D. B. Jourdan and O. L. de Weck, "Layout optimization for a wireless sensor network using a multi-objective genetic algorithm," in *Proceedings of the IEEE 59th Vehicular Technology Conference (VTC '04)*, vol. 5, pp. 2466–2470, Springer, New York, NY, USA, 2004.
- [20] W. Xiaoling, S. Lei, W. Jin, J. Cho, and S. Lee, "Energy-efficient deployment of mobile sensor networks by PSO," in *Proceedings of the International Workshop on Sensor Networks (IWSN '06)*, pp. 373–382, 2006.
- [21] H. T. Friis, "A note on a simple transmission formula," in *Proceedings of the Institute of Radio Engineers (IRE '46)*, pp. 254–256, 1946.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

