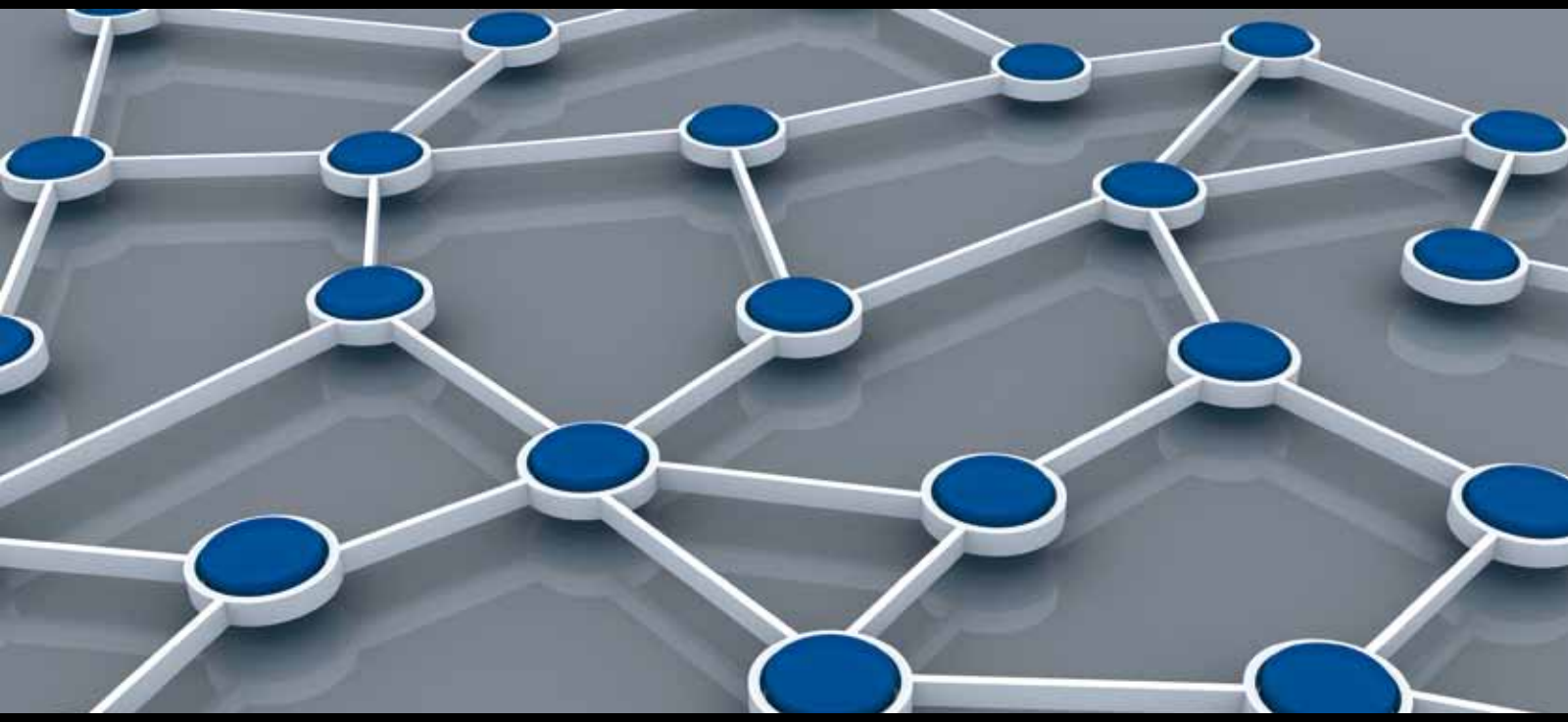
International Journal of Distributed Sensor Networks

Perception, Reaction, and Cognition in Wireless Sensor Networks

Guest Editors: Shuai Li, Congduc Pham, Arunita Jaekel, Mohammad Abdul Matin, Anang Hudaya M. Amin, and Yangming Li



Perception, Reaction, and Cognition in Wireless Sensor Networks

International Journal of Distributed Sensor Networks

Perception, Reaction, and Cognition in Wireless Sensor Networks

Guest Editors: Shuai Li, Congduc Pham, Arunita Jaekel, Mohammad Abdul Matin, Anang Hudaya M. Amin, and Yangming Li

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 696935, 2 pages
http://dx.doi.org/10.1155/2013/696935

Editorial

Perception, Reaction, and Cognition in Wireless Sensor Networks

Shuai Li,[1] Congduc Pham,[2] Arunita Jaekel,[3] Mohammad Abdul Matin,[4] Anang Hudaya M. Amin,[5] and Yangming Li[6]

[1] *Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA*
[2] *LIUPPA Laboratory, University of Pau (UPPA), Pau, 64100 Bayonne, France*
[3] *School of Computer Science, University of Windsor, Windsor, ON, Canada N9B 3P4*
[4] *Department of Electrical and Electronic Engineering, Institut Teknologi Brunei, Bandar Seri Begawan 8610, Brunei Darussalam*
[5] *Department of Computer and Information Sciences, Universiti Teknologi Petronas, 31750 Tronoh, Perak, Malaysia*
[6] *Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, Anhui 230031, China*

Correspondence should be addressed to Yangming Li; ymli@iim.ac.cn

Received 22 November 2012; Accepted 25 August 2013

1. Introduction

The past ten years have witnessed great developments of wireless sensor networks in both theory and application. The long been expected and advocated ubiquitous sensing is becoming increasingly popular and widely admitted with the great success of many applications of wireless sensor networks in environmental monitoring, precision agriculture, human health monitoring, and so forth. Correspondingly, theoretical foundations of wireless sensor networks, such as sensor positioning, time synchronization, communication protocols, data fusion, and operating systems, also have received intensive attentions. However, in current stage, most attentions in theory still regard wireless sensor networks as a means of data collection, instead of autonomous networks with self-decision making based on the collected data. The widely investigated type of wireless sensor network with perception but without reaction is in contrast to the autonomous network with perception, reaction, and cognition, which adapts itself to the monitored environment by exploiting information feedback (e.g., the collected data feedbacks to make adjustment of the electricity price for smart power grid network). Introducing reaction and cognition along with perception opens a door to transform wireless sensor network from a passive network for data collection to an adaptive and active network with self-intention, self-evolution, and self-intelligence and will open a new promising branch in the field of wireless sensor networks.

2. Major Topics around Perception, Reaction, and Cognition in Wireless Sensor Networks

With the capability of sensing, reacting, and thinking, a sensor node is increasingly like a live animal being and a fleet of them are connected together by information exchanging with others in the collection, which is in analogy to the social behaviors of animals. It is also in analogy to the skin of our human beings, which feels temperature with nerve endings in every area, makes decision with the nearby nerve cells, shrinks or constricts arterioles with the action of muscles, and thereby achieves the ability of regulating the body temperature and functions like a sensor network with perception, reaction, and cognition. From this perspective, biological systems or social behaviors of animals may give in-depth insight to the design of such a novel sensor network.

Integrating perception, reaction, and cognition into wireless sensor networks requires the effort of interdisciplinary researches, and this merging direction can be viewed from different perspectives. From the perspective of artificial

intelligence, the sensor network with perception, reaction, and cognition can be treated as a network with interactive agents; from a system perspective, the sensor network with perception, reaction, and cognition is a feedback system involving nonlinear estimation and control with interdependence. The goal of sensor networks with perception, reaction, and cognition lies in optimizing an objective function under constraints. Optimum for a single sensor node or a portion of nodes does not necessarily imply global optimum. As cooperative nonlinear optimization, game theory may find applications in this scenario. As each sensor node in this network needs to adapt to a possibly variant environment, real-time signal processing, data fusion, and data mining are often required requirements and pose more challenges than conventional wireless sensor networks, to which real-time performance is often not crucial. From the view of communication, congestion control, routing, protocol designs, and so forth may encounter new challenges as integrating real-time sensing and control together inevitably introduces more information exchanging and possibly communication burdens.

We accepted 5 papers press from the submissions of 16 to this special issue. The overall acceptance rate is 31.25%. The papers cover specific problems such as routing algorithms, indoor localization, energy harvesting, wormhole detection, and circle fitting algorithms. Some of the topics are crucial in conventional wireless sensor networks and also play important roles in the sensor network with the capability of perception, reaction, and cognition integrated. Some of the topics are more likely encountered in the latter case. Inspired by the large and single-celled amoeboid organism, slime mold *Physarum polycephalum*, the authors of the paper "*A novel Physarum-inspired routing protocol for wireless sensor networks*" propose a novel Physarum-inspired routing protocol (P-iRP) to address the routing issue in wireless sensor networks with a low complexity $O(\sqrt{n})$. The paper "*Circle fitting using a virtual source localization algorithm in wireless sensor networks*" solves the circle fitting problem in wireless sensor networks by formulating the problem into the special source localization one and employing the multidimensional scaling (MDS) analysis. The paper "*Indoor mobile localization in wireless sensor network under unknown NLOS errors*" solves the nonline-of-sight (NLOS) propagation problem by proposing a likelihood matrix correction based mixed Kalman and H-infinity filter (LC-MKHF) method. Results show that the LC-MKHF algorithm has higher estimate accuracy in comparison with no-filter, Kalman filter, and H-infinity filter methods and is robust to the NLOS errors. The paper "*Secure routing protocol using cross-layer design and energy harvesting in wireless sensor networks*" proposes a secure routing protocol based on cross layer design and energy-harvesting mechanism. This algorithm ensures efficient use of energy and performs better in many scenarios and in hostile attack-prone environment; Wormhole attack is a severe threat to wireless sensor networks. The authors of the paper "*MDS-based wormhole detection using local topology in wireless sensor networks*" propose a novel approach to detect wormhole attacks by only local topology information without requiring special hardware devices or depending on rigorous assumptions on the network settings. Extensive simulations demonstrate the effectiveness and superior performance of the proposed approach.

Acknowledgments

The guest editors would like to thank all the authors for their contributions. Special thanks go to all reviewers for their great effort, timely responses, and constructive comments and suggestions. Shuai Li would like to share the words by Rabindranath Tagore with the readers "If you shed tears when you miss the sun, you also miss the stars."

Shuai Li
Congduc Pham
Arunita Jaekel
Mohammad Abdul Matin
Anang Hudaya M. Amin
Yangming Li

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 247306, 7 pages
http://dx.doi.org/10.1155/2013/247306

Research Article

Indoor Pedestrian Positioning Tracking Algorithm with Sparse Anchor Nodes

Zhou Yong, Cai Zehui, and Chen Pengpeng

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, Jiangsu 221116, China

Correspondence should be addressed to Zhou Yong; zhouyongchina@126.com

Received 7 June 2013; Accepted 28 June 2013

Academic Editor: Shuai Li

In order to solve the indoor pedestrian positioning and tracking problems under the condition of sparse anchor nodes, this paper presents a new tracking scheme which predicts the staff position under the condition of indoor location fingerprints based on particle filter. In the proposed algorithm, the indoor topology is adopted to constrain and correct the results. Simulation results show that the proposed algorithm can significantly improve the accuracy of indoor pedestrian positioning and tracking more than the Kalman filter and k-nearest neighbor (KNN) algorithms. The simulation results also show that under the condition of sparse nodes deployment good tracking results can still be achieved through the adoption of indoor topology and the average positioning error is about 1.9 m.

1. Introduction

With the improvement of communication technology, location-based service (LBS) [1] has drawn more and more attention and will have a significant impact on human life and work. For example, the use of Global Positioning System (GPS) and electronic map in vehicle navigation and the use of intelligent mobile phone help people outside to find their place and route. Using a variety of sensors in indoor environments, for example, in a large supermarket or an exhibition hall can lead people to the places where they want to go. Although GPS can basically meet the requirements in outdoor environment for locating and tracking, it cannot work well in indoor environment. Currently the common adopted indoor localization techniques are mainly infrared, ultrasonic, ZIGBEE, wireless local area network (WLAN), ultrawide bandwidth (UWB), radio-frequency identification devices (RFID), magnetic signal, visual analysis and inertial measurement unit (IMU) [2], or the combination of multiple techniques.

Existing localization algorithms can be divided into the following three categories: range based, range-free, and event driven. Algorithms based on distance measurement need at least three sensors to locate trilateral using triangulation algorithm. Range-free algorithms are based on network connectivity information, which has lower location accuracy than the range-based algorithms. Event-driven localization makes use of localization events which are generated and propagated across the area where sensor networks are deployed. Although these algorithms are very effective, it is hard to employ them directly for the indoor positioning.

Indoor positioning and tracking order that sensors for locating should be deployed first. Increasing the density of coverage would increase the cost. In practice, the aim that the whole indoor area is covered by all anchors is often hard to achieve. Some dead angle inevitably exists. Considering the complexity of the indoor environment, the obstacle diffraction and reflection of the signal and the change of interior structure all have effects on the wireless signal for localization. Indoor localization and tracking should not only consider the normal usage, but also should consider the special cases, for example, the cases of fire or earthquake damage. Some anchor nodes or the order of the deployment of at least three sensors cannot be met [3]. In this paper, we put forward a positioning and tracking algorithm under the condition of sparse anchor nodes deployment, where the particle filter (PF) method based on the position fingerprint

is used and the constraints of the results of indoor topology and correction algorithms are also explored.

The rest of the paper is organized as follows. Section 2 briefly surveys previous localization methods. Section 3 presents the system overview and details the system design. Section 4 illustrates simulation results. Finally, Section 5 concludes the whole paper.

2. Related Work

Many excellent schemes have been proposed for the indoor localization. Most of them can be categorized into three classes: range-based localization, range-free localization, and event-driven localization.

Range-based localization algorithms are built on top of distance or angle measurements among the nodes in the networks, which require expensive hardware devices to estimate the distance between the nodes or need careful environment profiling. The Time of Arrival [4] and Time Difference of Arrival [5] schemes measure the propagation time of the signal and estimate the distance based on the propagation speed. The Angle of Arrival (AOA) schemes [6] estimate the node locations by sensing the received signal direction. The Received Signal Strength Indicator (RSSI) schemes [7] use either theoretical or empirical models to estimate the distance based on the loss of power during signal propagation. The fingerprint localization algorithm is based on signal strength, and it is with the benefit of simple calculation and high precision. The fingerprint based localization algorithm can be divided into two stages: the offline stage and the online stage [8].

To address the limitations of the range-based schemes, range-free localization schemes have been proposed, which attempt to locate sensors without costly ranging devices. The location of each node is estimated based on the knowledge of proximity to the anchor nodes. There are two kinds of localization schemes: anchor-based scheme and anchor-free distrusted localization scheme. Generally, range-free localization methods normally have low accuracy, highly depending on the density and distribution of the anchor nodes.

Recently, event-driven localization schemes have been proposed to simplify the node functionality and to provide high-quality localization. The key idea of these schemes is to use artificial events for localization. Although their effective range can reach hundreds of meters, it needs additional event generation devices and manual operations to generate artificial events.

In the tracking field, the location is often achieved through estimation and filtering like particle filters. It is a kind of method where Monte Carlo simulation is used to solve nonlinear and non-Gaussian problems of the Bayesian estimation [9]. It first uses a lot of weighted particles to represent the posterior distribution of the estimation. Then, particles are forecasted by transcendental motion equation information. Through the observed information, the weights could be updated accordingly. At last, the aforementioned two steps are run in cycles to realize the estimator of the distribution of the tracking.

3. Positioning and Tracking Algorithm under Sparse Anchor Nodes

In indoor wireless environments, various obstacles cause wireless signals irregular reflection and scattering. In addition, barrier properties such as metal, building materials, or human bodies could have different impacts on the propagation of wireless signals so that wireless signals in different buildings will have big gaps. Generally, positioning in indoor places requires the signals received from at least three anchor nodes, and receiving the signals from five or more anchor nodes can result in more accurate location (employing more than five or six nodes cannot further improve the positioning accuracy). Due to the specialty of indoor environments, wireless anchor node's deployment is hard to cover with at least three anchor nodes in every place. If the number is less than three, signals would be weak or the damaged results cannot provide the anchor nodes with any usable information for positioning, hence, causing intermittent positioning failure.

3.1. Tracking Algorithm. First, according to the principle of RSSI ranging, we set up the offline indoor radio frequency maps. Each point's signal strength is the average of several measurements, and the signal data format is $(x, y, \mathrm{RSS}_1, \ldots, \mathrm{RSS}_i, \ldots, \mathrm{RSS}_m)$, where x and y are coordinates and RSS_i is the detected signal strength of anchor node i. We calibrate the signal intensity maps and store them in the database. At the online stage, the moving target node receives the real-time RSS signal $(S_1, S_2, \ldots S_m)$. One should select the minimum Euclidean distance of the k results and look for the average location. As the observed value of the particle filter, one should also use the particle filter algorithm to get the final localization results. The formula of Euclidean distance between the received signal strength and the locating fingerprint can be expressed as follows:

$$d_i = \sqrt{\sum_{i=1}^{m} (\mathrm{RSS}_i - S_i)^2}. \quad (1)$$

According to the number of anchor nodes in positioning and tracking environments, indoor location tracking process should be discussed for the following two different situations: (1) the situation for sufficient anchor node's indoor positioning and (2) the situation for sparse anchor node's indoor positioning. In the first case, the target node is covered by three or more anchor nodes with the initial location being obtained by the KNN algorithm. Then, one can use particle filter to determine the final location. In the second case, the target node is covered by two, one or zero anchor nodes (note that the case of three collinear anchor nodes is similar to that of two anchor nodes). In this case, the location of the target node cannot be directly located, which will cause intermittent positioning failures. As shown in [10], the environments covered by two anchor nodes can locate two target locations which connect symmetrically with the two anchor nodes. When only covered by one anchor node, the range of the target node location is a circle with its radius the distance

from the anchor node, and zero anchor node coverage cannot be located [11].

Although sparse anchor nodes cannot directly locate the target nodes, some facts can be used for the positioning constraint. By setting the target node S's maximum speed as $v_{\max}$ m/s sampling time interval T, the anchor node coverage is r, and we can get the following constraint conditions.

(1) Set P_t as the current time's position, P_{t-1} for the location of previous time; there is $P_t < P_{t-1} + T * v_{\max}$; namely, the current moment's location is always in the scope of a circle whose radius is $T * v_{\max}$.

(2) If the target only receives the signals from two anchor nodes a and b, it means that the target cannot be covered by other anchors except a and b. We define the formula $x \in y$, which means that x node is within the communication range of y node. Then, the above situation can be expressed as $S \notin \{U - \{a, b\}\}$, where S is the target node and U is the collection of all anchor nodes in the network. By the same token, the target node is covered by one anchor node a, having the constraint $S \notin \{U - a\}$.

(3) If the target is not covered by any anchor, then we have $S \notin U$. In this situation, we will combine the following conditions to determine the target location as follows: the location at the last step, the maximum target speed limitation, and the area which cannot be covered by any anchor in the fingerprint database.

The aforementioned are the filtering conditions in the tracking process based on particle filter algorithm. If the particle cannot satisfy the above conditions, it should be filtered.

3.2. Particle Filter Process. For simplicity, the particle filter method refers to finding a random sample of groups in the state space transmission and, thereafter, to approximating the probability density function, where integral operation is replaced by a sample mean, and, hence, it achieves the minimum variance distribution process. The samples here refer to particles, while the number of samples $N- >\infty$ can approximate any probability density distribution. The detailed particle filter algorithm [9, 12] is given as follows.

(1) Initialization, sampling from the initial distribution of the particle:

$$x_0^i \sim p(x_0), \quad i = 1, 2, 3, \ldots, n, \tag{2}$$

where x_0^i is the ith sampling particle, $p(x_0)$ is the initial distribution of the ith particle, $\omega_0^i = 1/n$ is ith particle's weight, and n is total number of the particles.

(2) Weight calculation is as follows:

$$x_k^i \sim q\left(x_k \mid x_{0:k-1}^i, z_{0:k}\right), \quad i = 1, 2, \ldots, n. \tag{3}$$

The importance weights are calculated as follows:

$$\omega_k^i = \omega_{k-1}^i \frac{p\left(z_k \mid x_k^i\right) p\left(x_k^i \mid x_{k-1}^i\right)}{q\left(x_k^i \mid x_{0:k-1}^i, z_{0:k}\right)}, \quad i = 1, 2, \ldots, n. \tag{4}$$

And, the importance weights are normalized as

$$\overline{\omega}_k^i = \frac{\omega_k^i}{\sum_{i=1}^N \omega_k^i}. \tag{5}$$

(3) Resample x_k^i, $i = 1, 2, 3, \ldots, n$.

According to the importance weights $\overline{\omega}_k^i$, resample is carried out to get the updated n particles, x_0^i, $i = 1, 2, \ldots, n$. And the redistribution of the particle weight $\omega_k^i = \overline{\omega}_k^i = 1/n$.

(4) Output

state estimation:

$$\overline{\widehat{x}}_k = \sum_{i=1}^N \omega_k^i \overline{\widehat{x}}_k^i, \tag{6}$$

variance estimation:

$$P_k = \sum_{i=1}^N \omega_k^i \left(\overline{\widehat{x}}_k^i - \widehat{x}_k\right)\left(\overline{\widehat{x}}_k^i - \widehat{x}_k\right)^T. \tag{7}$$

Step (1) is performed only at the beginning of the algorithm, and the other steps are performed sequentially. Finally, the particle set $\{x_k^i, \omega_k^i \mid i = 1, 2, \ldots, n\}$ is updated to achieve the target posterior distribution tracking.

3.3. Indoor Topology Constraint. Many logic errors may occur in indoor locating cases, for example, positioning tracking information, jumping from one room to another room or to the corridor. In the process of fingerprint-based positioning, there may be a new position for penetrating a wall. Although the derived results are the optimum of the position, the actual route may be very long [13], which is much farther than the value of $T * v_{\max}$. In order to reduce these logic errors, we introduce indoor topology constraints into the positioning tracking algorithm.

Indoor topology structure can be described as a connected graph, and the distance between the two fingerprints can be obtained by using the Dijkstra algorithm and getting the monophyletic shortest path. The fingerprint of each adjacent position is taken as a node of graph where the connection line for the edge and the weights of edge represent the distance of position fingerprint.

In Figure 1(a), the gray box is the position fingerprint, and each box A, B, C, and D can be regarded as connected graph vertex, in which the adjacent points can be connected, as shown in Figure 1(b). Adjacent position fingerprint distance is set to 1, the available distance is 3 between A and B, and the distance from A to D though is 1, but actually the shortest path is 7. So, the indoor topology for positioning in the process of the shortest path can effectively filter out positioning location which is not in conformity with the conditions. As shown in Figure 2, P_{t-1} is the location a moment before; $P_{t'}$ is the location got from the KNN algorithm; R is the largest distance of target node; and $R = T * v_{\max}$. The two position's

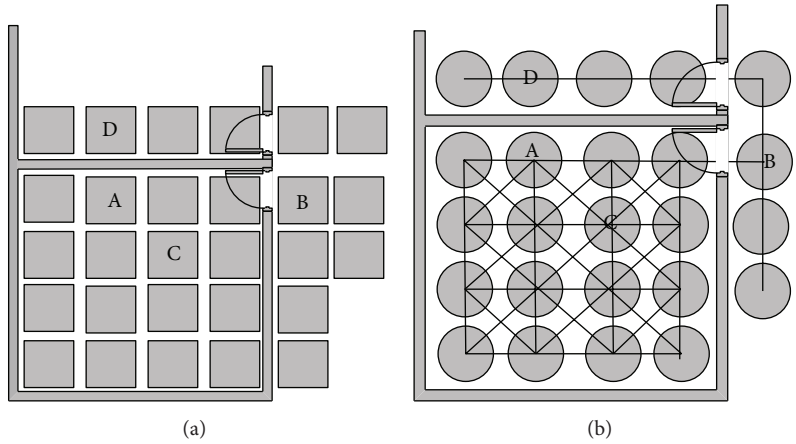


Figure 1: Position fingerprint diagram and connected graph example.

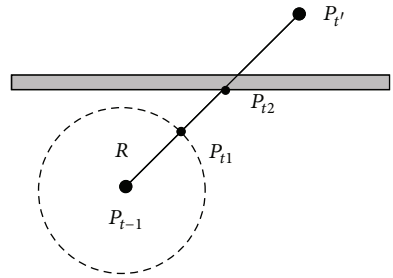


Figure 2: The confirmed location of the interior topology constraints.

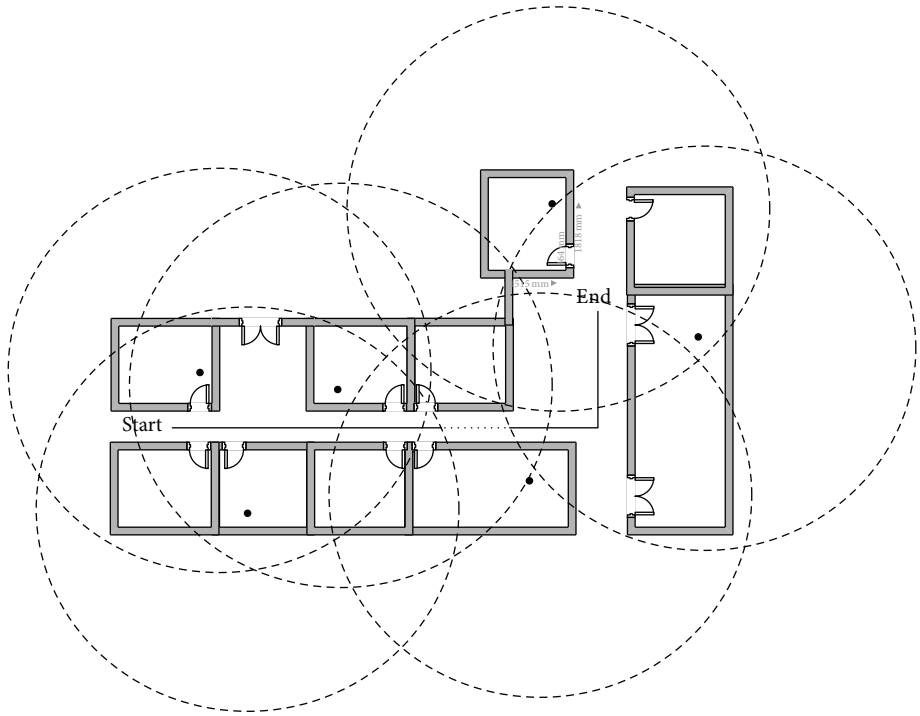


Figure 3: Simulation environment (• is an AP node).

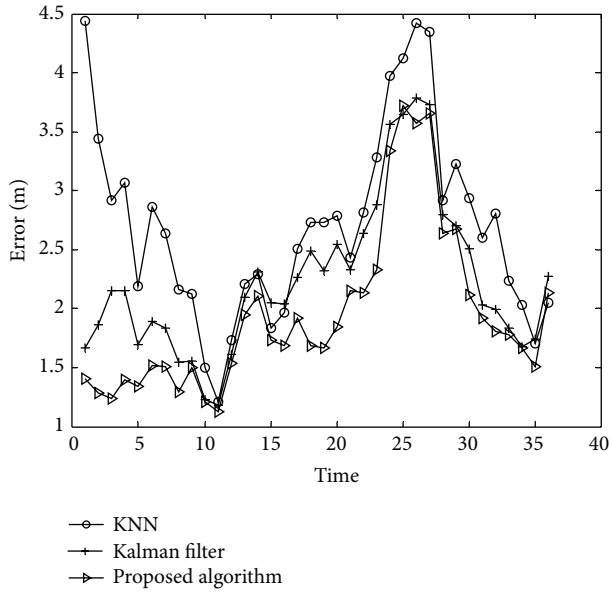


Figure 4: Accuracy comparison.

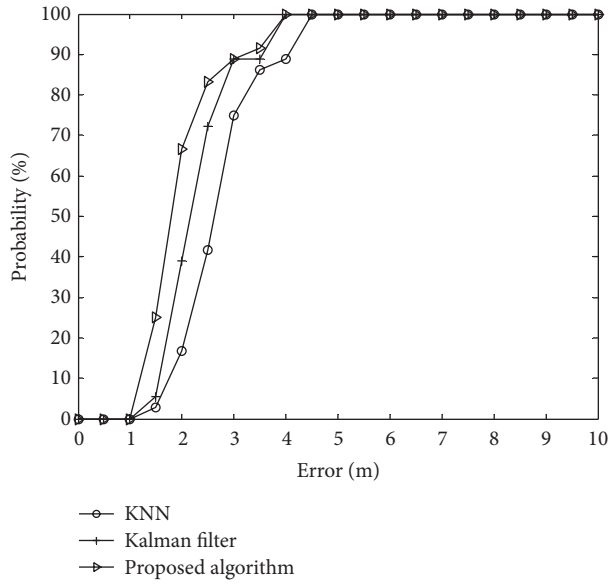


Figure 5: Accuracy comparison.

connections intersect the target node, respectively, and the target node moves along the circle and walls between P_{t1} and P_{t2}. The current position of P_t can be determined by

$$P_t = \begin{cases} P_{t1}, & \left(P_{t-1} + T * v_{\max} < P_{t2}\right) \\ P_{t2}, & \left(P_{t-1} + T * v_{\max} > P_{t2}\right). \end{cases} \quad (8)$$

So, the indoor topological constraints remove the location which mismatches the conditions, and it also identifies the location most close to the real location of P_t.

The indoor topological structure is basically the same. So, in the offline phase of the fingerprint procedure, one can calculate the shortest distance between two fingerprints' position and store the results in the database. Thereafter, one can get the new location from the database by using the particle filter under the indoor topology constraints and can get a more accurate estimated position.

3.4. Algorithm Flow. Algorithm 1 depicts the whole working flow. This is suitable for positioning and tracking under the sparse anchor node condition. Note that the algorithm is featured with the condition of small number of anchor nodes less than three which would lead to intermittent positioning failure.

Table 1: Tracking results comparison.

Algorithm	Percentage in 2 m error	Percentage in 3 m error	Average error (m)
KNN	16.67	75.00	2.7011
Kalman	38.89	88.89	2.2400
Proposed algorithm	66.67	88.90	1.9476

Input: fingerprint $(x, y, RSS_1, RSS_i, \ldots, RSS_m)$ of anchor location in indoor environment and the signals measured by target node $(S_1, S_2, \ldots, S_m)$.
Output: the target node's location curve resulting from the operation of the algorithm.
(1) (x, y) = KNN % Using the KNN algorithm to find k nearest position and get the observation values;
(2) $(X, Y) = \left(\sum_{i=1}^{N} \omega_k^i x_k^i, \sum_{i=1}^{N} \omega_k^i y_k^i\right)$ % Input the observation values (x, y) to the particle filter algorithm.
(3) The filtering condition is as follows:
$P_t < P_{t-1} + T * v_{\max}$ % The distance between two adjacent locations should be smaller than $T * v_{\max}$

$$\begin{cases} S \notin \{U - \{a, b\}\} \text{ \% When the target is covered by two anchors} \\ S \notin \{U - a\} \text{ \% When the target is covered by one anchor} \\ S \notin U \text{ \% When the target isn't covered by any anchor} \end{cases}$$

$$P_t = \begin{matrix} P_{t1}, (P_{t-1} + T * v_{\max} < P_{t2}) \\ P_{t2}, (P_{t-1} + T * v_{\max} > P_{t2}) \end{matrix}$$

% The topology constraints.
(4) Until all times' positioning and tracking is over, one can get the final positioning tracking curve.

Algorithm 1: Indoor pedestrian tracking.

4. Simulation Evaluation

To verify the effectiveness of the algorithm above, we simulated the algorithm using MATLAB platform by simulating an indoor corner in the building of the computer school of China University of Mining and Technology. We compare the proposed algorithm that combines the particle filter and topology constraints with the other indoor localization algorithms. The simulated environment and the AP distribution are shown in Figure 3.

Because the anchor node deployment is not dense, the changes of the complexity environment often cause indoor pedestrians' nodes positioning not always receiving more than three anchor node's signals. As shown in Figure 3, there is a dotted line which only receives two anchor node's signals, which will cause the intermittent failure positioning problem.

In order to evaluate the performance of the proposed algorithm, the algorithm was compared with the traditional KNN algorithm and Kalman filter algorithm. Experimental model parameters are specified as follows: the maximum rate of mobile target node $v_{\max}$ is 1.3 m/s and the targets locate themselves every T ($T = 1$ s) time. In the KNN algorithm, the parameter k is set to 4. The number of particles N is set to 200 in particle filter algorithm. To ensure the reliability of the experimental results, this study samples 36 times and locates and tracks 50 times repeatedly to get the average data. The location errors are shown in Figure 4. We can observe that the tracking performance of our algorithm is better than the KNN algorithm and the Kalman filter algorithm, although the distribution of errors is difficult to figure out in the figure. In Figure 5, we draw the percentage error of the cumulative distribution every 0.5 m error interval. It can be seen that the algorithm error percentage is 2 m and 3 m, so the proposed algorithm in this paper has a better advantage.

The statistical results of the test data are shown in Table 1. It can be seen that the proposed particle filter and the indoor topology constraints algorithm can obtain better tracking precision. The average error of KNN and Kalman filter algorithm is 2.7011 meters and 2.24 meters, respectively, while the error of the proposed algorithm is 1.9476 meters. Compared with the KNN and Kalman filter algorithm, the percentage of the error that is smaller than 2 meters is increased to 66.67%. And, the percentage of the error that is smaller than 3 meters is increased to 88.91%. The simulations show that when combined with indoor topology constraints, positioning effect can be improved efficiently.

5. Conclusion

Indoor localization is the research hotspot in location based on services. At present, most of the indoor positioning research focuses on anchor nodes deployed sufficiently without considering the change of indoor environment. This may lead to weaker signal that cannot be used to locate. Or anchor nodes' fault can cause the sparse deployment, which leads to intermittent positioning failure problem. To this end, we put forward an indoor positioning algorithm under sparse anchor nodes by building an indoor radio-frequency fingerprint map and using KNN algorithm to obtain initial position location under the condition of sufficient anchor nodes, while one gets optimal positioning location with

sparse anchor node by a series of constraints measures and uses particle filter tracking algorithm to solve nonlinear state space problems. Simulation experiments show that our algorithm can achieve good positioning and tracking results.

The indoor target tracking is a huge and complex engineering, and many issues still remain to be explored. Our ongoing work is as follows: (1) because the anchor placement has direct influence on the tracking accuracy, we will further study the indoor placement problem of anchors; (2) in the indoor environment, there may be multiple targets; thus, we will extend the proposed algorithm for multiple targets tracking; and (3) in the future, we shall consider the detailed hardware implementation and extend this work into the real scenario.

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2013XK10.

References

[1] P. Qiu, J. Zhang, and J. Zeng, "Study on the mobile LBS development model," in *Proceedings of the International Conference on Computer Science and Service System*, pp. 1070–1074, Nanjing, China, August 2012.

[2] K. Al Nuaimi and H. Kamel, "A survey of indoor positioning systems and algorithms," in *Proceedings of the International Conference on Innovations in Information Technology (IIT '11)*, pp. 185–190, Abu Dhabi, United Arab Emirates, April 2011.

[3] Y. Jin, M. Motani, W. S. Soh, and J. Zhang, "SparseTrack: enhancing indoor pedestrian tracking with sparse infrastructure support," in *Proceedings of the IEEE INFOCOM 2010*, pp. 1–9, San Diego, Calif, USA, March 2010.

[4] B. H. Wellenhoff, H. Lichtenegger, and J. Collins, *Global Positions System: Theory and Practice*, Springer, New York, NY, USA, 4th edition, 1997.

[5] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "Cricket location-support system," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 32–43, August 2000.

[6] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, vol. 3, pp. 1734–1743, San Francisco, Calif, USA, April 2003.

[7] G. Gracioli, A. A. Fröhlich, R. P. Pires, and L. F. Wanner, "Evaluation of an rssi-based location algorithm for wireless sensor networks," *IEEE Latin America Transactions*, vol. 9, no. 1, pp. 830–835, 2011.

[8] A. K. M. M. Hossain, Y. Jin, W. S. Soh, and H. N. Van, "SSD: a robust RF location fingerprint addressing mobile devices' heterogeneity," *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 65–77, 2013.

[9] J. M. del Rincón, D. Makris, C. O. Uruñuela, and J. C. Nebel, "Tracking human position and lower body parts using Kalman and particle filters constrained by human biomechanics," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 41, no. 1, pp. 26–37, 2011.

[10] A. H. Alasiry and S. Ohyama, "Range-based localization with area verification for sparse distributed wireless sensor networks," in *Proceedings of the International Conference on Advanced Computer Science and Information Systems*, pp. 43–46, Depok, Republic of Indonesia, 2012.

[11] J. Luo and Q. Zhang, "Relative distance based localization for mobile sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 1076–1080, Washington, DC, USA, November 2007.

[12] D. Anzai and S. Hara, "Does particle filter really outperform low pass filter in indoor target tracking?" in *Proceedings of the IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC '10)*, pp. 882–886, Instanbul, Turkey, September 2010.

[13] H. Shin and H. Cha, "Wi-Fi fingerprint-based topological map building for indoor user tracking," in *Proceedings of the 16th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '10)*, pp. 105–113, Macau, China, August 2010.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 483581, 12 pages
http://dx.doi.org/10.1155/2013/483581

Research Article

A Novel *Physarum*-Inspired Routing Protocol for Wireless Sensor Networks

Mingchuan Zhang,[1,2] Changqiao Xu,[1,3,4] Jianfeng Guan,[1,4] Ruijuan Zheng,[2] Qingtao Wu,[2] and Hongke Zhang[1,4]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China*
[3] *Institute of Sensing Technology and Business, Beijing University of Posts and Telecommunications, Wuxi 214135, China*
[4] *National Engineering Laboratory for Next Generation Internet Interconnection Devices, Beijing Jiaotong University, Beijing 100876, China*

Correspondence should be addressed to Mingchuan Zhang; zmc@bupt.edu.cn

Received 31 January 2013; Accepted 7 June 2013

Academic Editor: Shuai Li

There is a tradeoff between routing efficiency and energy equilibrium for sensor nodes in wireless sensor networks (WSNs). Inspired by the large and single-celled amoeboid organism, *slime mold Physarum polycephalum*, this paper presents a novel *Physarum*-inspired routing protocol (P-iRP) for WSNs to address the above issue. In P-iRP, a sensor node can choose the proper next hop by using a proposed *Physarum*-inspired selecting next hop model (P-iSNH), which comprehensively considers the distance, energy residue, and location of the next hop. As a result, the P-iRP can get a rather low algorithm complexity of $O(\sqrt{n})$, which greatly reduces the processing delay and saves the energy of sensors. Moreover, by theoretical analysis, the P-iSNH always has an equilibrium solution for multiple next hop candidates, which is vital factor to the stability of routing protocol. Finally, simulation results show that P-iRP can perform better in many scenarios and achieve the effective tradeoff between routing efficiency and energy equilibrium compared to other famous algorithms.

1. Introduction

With the development of communication, electron, and sensor technologies, wireless sensor networks (WSNs) have attracted wide concern of both researchers and application providers. WSNs consist of large numbers of sensor nodes deployed over a certain region. Each sensor node is a low-cost, short range wireless transceiver typically equipped with a low-computation processor and a battery operated power supply. Under many scenarios, the sensor nodes need to operate without battery replacement for several years. Thus, there are two questions need to be considered. One is how to achieve energy balance of these nodes to avoid the emergence of energy holes which commonly take place around the sink, since the data traffic follows a many-to-one communication pattern and nodes nearer the sink have to take heavier traffic load. The other is how to obtain high routing efficiency under multihop transmission circumstance, since WSNs can contain hundreds of such low-cost sensor nodes. Therefore, designing such networks should primarily focus on both routing efficiency and energy equilibrium in terms of trade-off.

Location-aware routing protocols seem to possess high routing efficiency, where GPS, phone, or other techniques are used for positioning nodes [1]. However, there are two extremes in location-aware routing—the greedy strategy and the robust strategy. Greedy strategies may suffer failures to route packets to destination, while robust strategies need very high flooding rates to ensure reliability and rapid delivery of data. Thus, many location-aware routing protocols are mostly to propose methods to overcome the mentioned drawbacks [2–4]. The energy-aware routing attracts more researchers'

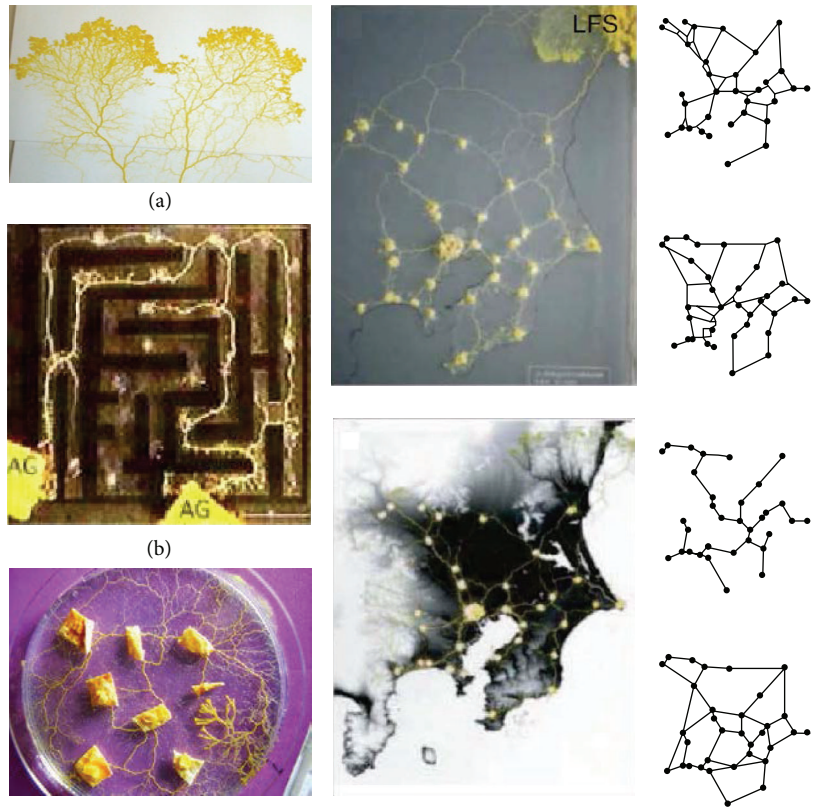


Figure 1: Photographs of *Physarum*. (a) *Physarum* is able to make complex comparisons between two food options based on the quality difference of the food and riskiness of the feeding environment, which comes from http://sydney.edu.au/news/sobs/1699.html?newsstoryid=4576. (b) Example of maze solving by *Physarum*, which comes from paper [25]. (c) Example of connecting path in a uniformly illuminated field which comes from http://cmr.soc.plymouth.ac.uk/research.htm. (d) Comparison of the *Physarum* networks with Tokyo rail network, which comes from paper [26].

attentions than that of location-aware routing for the significance of energy. There are many results relating to energy-aware routing recently [5–8] to save energy or prolong WSNs' lifetime, where energy harvesting [9] is shown to be a promising technique.

Unsurprisingly, the combination between location-aware routing and energy-aware routing becomes another researchers' focus to balance energy and efficiency for WSNs' routing protocol [10, 11]. In addition, some researchers focus on other aspects of WSNs' routing protocol, for example, the distributed characteristic [12, 13] and the trade-off of other two or more indexes [14, 15].

In recent years, the *slime mold Physarum polycephalum*, a large single-celled amoeboid organism, becomes a new researchers' pet and has shown to be a good technique for solving the shortest-path problem, since it can adapt its organism to forage for patchily distributed food sources, as shown in Figure 1. In this paper, we draw the inspiration from the *Physarum*, introduce the *Physarum* model into WSNs, and improve it through ignoring its dimension and preserving its logical meanings to make it suit for routing selection based on our prior works [16–20]. Our focus is to choose the proper next hops to transmit data to sink in thinking of both routing efficiency and energy equilibrium, which is partially similar to [11].

The rest of this paper is organized as follows. Section 2 gives a brief description of related work. Section 3 formulates the proposed models. Section 4 details the P-iRP. Section 5 discusses the feasibility of P-iSNH. Section 6 evaluates our P-iRP by simulations. Finally, the conclusion is presented in Section 7.

2. Related Work

In the aspect of efficient routing, GPSR [2] is a famous greedy routing protocol, which makes greedy forwarding decisions using only information about a router's immediate neighbors in the network topology. Li et al. [21] present a neural network approach to plan the shortest path from the target position to the start position in real time. Kuhn et al. [3] utilize face (or perimeter) routing to go around voids in the topology. Padmanabh et al. [4] consider unbiased random walk on a regular deployment of nodes, forming a hexagonal lattice pattern.

In the aspect of the energy-aware routing, Trajcevski et al. [5] construct a data aggregation tree that minimizes the total

energy cost of data transmission, which is shown as an NP-complete problem, and propose algorithms for addressing it. A battery aware power allocation model was studied in [6] for a single-hop transmission scheme to balance the network energy consumption based on the nonlinear battery parameters proposed in [7]. Chau et al. [8] consider that a portion of the lost charge can be recovered due to the battery's recovery effect and present a battery model. The approaches propose some of the routes that would otherwise need to bypass the hole along the boundary and should start to deviate from their original path further from the hole instead.

Moreover, distribution and clustering problems are also important branches of WSNs' routing. Li et al. [13] consider the problem of nonlinear constraints defined on a graph and give a better solution incorporated with Laplacian eigenmap as heuristic information to solve the problem in distributed scenarios. For maximizing the network lifetime, Rao and Fapojuwo [22] present a battery aware distributed clustering and routing protocol which incorporates the state of the battery's remaining charge and health parameters in computing the charge utility metric at each cluster formation round. Wang and Syue [23] propose a relay selection protocol based on geographical information, in which multihop transmission is realized by concatenation of single cluster-to-cluster hops, where each cluster-to-cluster scheme forms the simplified cooperative network that consists of a single source destination pair and a set of available relays.

However, the trade-off is not comprehensively considered in those papers, which is very necessary for WSNs' routing due to the features of WSNs, for example, nodes' failures, limited bandwidth, and power energy. Bai et al. [10] route the connections in a manner that link failure does not shut down the entire stream but allows a continuing flow for a significant portion of the traffic along multiple paths to address the issues of reliability and energy efficiency. Trajcevski et al. [24] present heuristic approaches to relieve some of the routing load of the boundary nodes of energy holes in location-aware WSNs to balance load and latency. Yu et al. [11] use energy aware neighbor selection to route a packet towards the target region and recursive geographic forwarding or restricted flooding algorithm to disseminate the packet inside the destination region. By allowing the battery to rest for certain duration, without being subjected to heavy loads, Yang and Heinzelman [14] propose sleeping multipath routing, which selects the minimum number of disjoint paths to achieve the trade-off of given reliability requirement and energy efficiency. Sivrikaya et al. [15] propose randomized routing based on Markov chains to balance the load and routing performance.

In recent years, the *Physarum* becomes a new focus of bioinspired method. It is also the original source of our inspiration. Nakagaki et al. [25] validate that the *Physarum* is apparently able to solve shortest path problems as shown in Figure 1. They build a maze, cover it with pieces of *Physarum* (the *Physarum* can be cut into pieces that will reunite if brought into vicinity), and then feed the *Physarum* with oatmeal at two locations. A few hours later, the *Physarum* retracts to a path that follows the shortest path connecting the food sources in the maze. Tero et al. [26] use *Physarum* forms networks comparable efficiency, fault tolerance, and cost to those of real-world infrastructure networks—Tokyo rail system. Tero et al. [27] propose a mathematical model for the behavior of *Physarum* and argue extensively that the model is adequate.

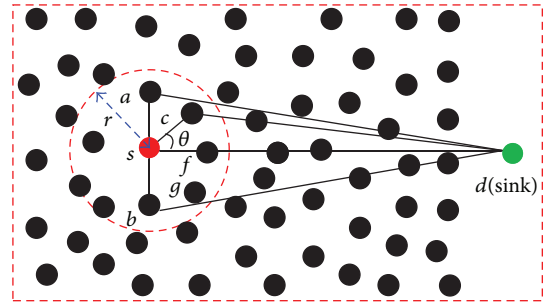


Figure 2: An example of sensor nodes' deployment in WSNs.

3. System Models

Our research is built on three assumptions. The first is that all nodes are aware of their locations, which may be achieved through GPS receivers at network deployment time, employing a distributed location discover algorithm shortly after deployment or adopting other positioning methods [1, 28]. The second is that each node is aware of its energy residue [22]. The third is that the link is bidirectional, that is, if a node hears from a neighbor, then its transmission range can reach to the neighbor.

3.1. Typical WSNs Scenario. We consider the large multi-hop WSNs which consist of n static sensors. Each node i has a fixed circular transmission range r which determines the set of sensors in which each node can communicate with node i in one hop. We abstract such WSNs using a graph $G = (V, E)$, where each node $v \in V$ represents a sensor, and each edge $e \in E$ represents the existence of one-hop wireless link between two sensors.

We suppose that node s is the source node and node d is the sink, as shown in Figure 2. In most cases, the sink d is placed in the middle of WSNs field to ease traffic burdens of nodes in the right of the WSNs field. In this paper, we only think of the nodes in left of the WSNs field for simplicity and clarity. The transmission range of s is drawn as a dashed circle whose radius is r and center is s. We call the angle θ is the angle of deviation of node c, which represents a measurement of node c deviating from the sink d. The Euclidean distance of any two nodes, i and j, and the angle θ_{jid} can be calculated following from (1) and (2), respectively

$$L_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{1}$$

$$\theta_{jid} = \arccos \frac{L_{sc}^2 + L_{sd}^2 - L_{cd}^2}{2 \times L_{sc} \times L_{sd}}, \tag{2}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of nodes i and j, respectively.

(a) Traditional data packet format

(b) Our data packet format

SA: Source Address
NA: Next Hop Address
ER: Energy Residue

Figure 3: Formats of data packets.

If the node s needs to transmit data to sink d, it will select its next hop in the dashed circle. Since our scenario is location aware, we select the next hop in the right semicircle under normal circumstances. Obviously, the smaller the angle θ is, the closer the next hop is to the sink for a fixed distance. That is to say, we are apt to choose the node whose θ is smaller as the next hop. In order to avail discussion, we define the N_s is the set of neighbors of node s (in the dashed circle), N_s^L is the set of left neighbors of node s (in the left semicircle), and N_s^R is the set of right neighbors of node s (in the right semicircle). Then, N_s, N_s^L, and N_s^R meet the following:

$$\begin{aligned} N_s &= N_s^L \cup N_s^R \\ N_s^L &\cap N_s^R = \emptyset. \end{aligned} \tag{3}$$

In addition, the energy residue of each node is also important for choosing next hop for balancing the energy of WSNs' nodes. When a node chooses its next hop, it would consider the energy residue of the candidates and be apt to pick the node which has much higher energy residue as the next hop. Therefore, it is important to acquire the energy residue of neighbors.

We think of the basic theory of wireless transmission combined with Figure 2 and the data packet format shown in Figure 3(a). If node s transmits a group of data to node c, all of the nodes in N_s would receive the wireless radio and check the packet header. The node c matches the field NA and receives the packet. Other nodes mismatch the field NA then ignore the packet and go on sleeping.

In order to acquire the energy residue of neighbors, we add a new field ER to the packet header, which is shown in Figure 3(b). When node s transmits a group of data to node c, all of the nodes in N_s extract the fields of SA and ER from the packet header and save ER in local memory according to SA. Then, the node c matches the field NA and receives the packet. Other nodes mismatch the NA then ignore the packet and go on sleeping.

Since each node needs to listen in real time to every packet and try to match its field NA, only adding an operation of saving ER would not add a considerable effect on energy consumption. Therefore, we neglect the cost of acquiring energy residue of neighbors.

3.2. Physarum-Inspired Path-Finding Model. Papers in [25–27] exploit the *slime mold Physarum polycephalum* to develop a *Physarum*-inspired path-finding model (PiPf). Suppose that (1) the initial shape of a *Physarum* organism is represented by a graph, (2) the edges represent plasmodial tubes in which protoplasm flows, and nodes are junctions between tubes, (3) the pressures at nodes i and j are P_i and P_j, respectively, and the two nodes are connected by a cylinder of length L_{ij} and radius r_{ij}, and (4) the flow is laminar and follows the Hagen-Poiseuille equation. Then, the flux through the tube is calculated as in the following

$$Q_{ij} = \frac{\pi r_{ij}^4 \left(P_i - P_j\right)}{8\eta L_{ij}} = \frac{D_{ij}\left(P_i - P_j\right)}{L_{ij}} = \frac{D_{ij}\Delta P_{ij}}{L_{ij}}, \tag{4}$$

where $\Delta P_{ij} = P_i - P_j$ is the difference of pressures, η is the viscosity of the fluid, and $D_{ij} = \pi r_{ij}^4/8\eta$ is a measure of the conductivity of the tube. As the length L_{ij} is a constant, the behavior of *Physarum* is described by the conductivities, D_{ij}, of the edges.

Equation (4) represents that the flux through the tube ij is determined by D_{ij}, ΔP_{ij}, and L_{ij}. The better the conductivity of the tube ij is and the larger the pressure difference ΔP_{ij} is, the more the flux through the tube ij is, while the longer the length of the tube ij is, the less the flux through the tube ij is.

Suppose that the capacity of each node is zero, the conservation law of each node is calculated from the following:

$$\sum_j Q_{ij} = \begin{cases} I, & i = s \ \ j \in N_i, \\ -I, & i = d \ \ j \in N_i, \\ 0, & \text{others} \ \ j \in N_i, \end{cases} \tag{5}$$

where I is the flux flowing from the source node (or into the sink node). It should be noted that I is a constant in *Physarum* model, which means that the total flux is fixed constant throughout the process.

Equation (5) illuminates the flux relationship in each node. In the source node s, I is the flux flowing from it; in the sink node d, I is the flux flowing into it; and in intermediate nodes, the sum of flowing from and flowing into is zero.

Physarum forages for distributed food sources through the adaptive behavior of the plasmodium. The adaptive behavior is illustrated as follows combined with Figure 4(a), where two food sources are connected by two tubes. Because of $\Delta P_{ij}^1 = \Delta P_{ij}^2$ and $L_{ij}^1 > L_{ij}^2$, the flux Q_{ij}^2 will be greater than Q_{ij}^1 from (4). Note that L_{ij}^1 and L_{ij}^2 are kept constant throughout the adaptation process in contrast to D_{ij}; therefore, the adaptive behavior can be described by the evolution of $D_{ij}(t)$,

$$\frac{d}{dt}D_{ij} = \varphi\left(\left|Q_{ij}\right|\right) - \delta D_{ij}, \tag{6}$$

where δ is a decay rate of the tube. Equation (6) implies that the conductivity tends to vanish if there is no flux along the edge, while it is enhanced by the flux. It is natural to assume that $\varphi(\cdot)$ is a monotonically increasing continuous function satisfying $\varphi(0) = 0$.

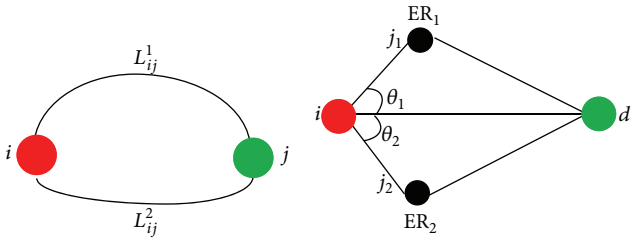


Figure 4: (a) If two food sources are connected by two tubes, the longer tube will vanish with time going by. (b) If there are two candidates for next hop, the node that has the greater $k \times \mathrm{ER} + (1 - k)L \cos\theta$ will be picked as the next hop.

Equation (6) illustrates the variation relationship of the conductivity with time to accommodate the flux distribution of the multipath transmission. In equilibrium ($\varphi(|Q_{ij}|) = \delta D_{ij}$ for all edges), the flow through any edge is steady. In nonequilibrium, the diameter grows or shrinks if $\varphi(|Q_{ij}|)$ is larger or smaller than δD_{ij}, respectively.

The PiPf consisting of (4), (5), and (6) describes the evolutionary process of *Physarum* to solve the path-finding behavior of self-organized networks.

3.3. Physarum-Inspired Selecting Next Hop Model. In this section, we improve the PiPf and make it fit for routing in WSNs based on dimensionless analysis method. That is to say, we improve the PiPf to achieve a *Physarum*-inspired selecting next hop model (P-iSNH). In order to obtain that, there are two problems that need to be solved. The first is which physical quantities are used to replace the conductivity D_{ij}, the length L_{ij}, and the pressure difference ΔP_{ij}. The second is how to select the proper next hop.

Equation (4) derives from fluid dynamics. D_{ij} is a measure of the conductivity of the tube; L_{ij} is the length of the tube; and ΔP_{ij} is the differential pressure of tube on both ends. However, the D_{ij}, L_{ij}, and ΔP_{ij} cannot be directly used in WSNs where we need to consider the link quality, energy residue, transmission direction, and the distance of one hop.

First, because the D_{ij} is an inherent characteristic of the tube, we should replace the D_{ij} by an inherent physical quantity. Apparently, the link quality Φ_{ij} is an inherent characteristic relating to wireless link, so we replace the D_{ij} by Φ_{ij} in our model.

Second, the meaning of L_{ij} is the same as in fluid dynamics. However, in wireless communication, there is a path-loss exponent α, which has a great effect on transmission. Therefore, we replace the L_{ij} by L_{ij}^{α} in our model.

Third, we discuss the ΔP_{ij} combind with Figure 4(b). On one hand, suppose that there is a potential field from node i to sink d. The potential difference between i and j can be expressed by $K \times L_{ij} \cos\theta$. Because parameter K is unimportant in the judge process, we use $L_{ij} \cos\theta$ expressing $K \times L_{ij} \cos\theta$. When node i chooses its next hop, it is apt to pick the node whose potential difference is much greater. On the other hand, because the next hop needs to consume energy to deal with data packets, it is apt to pick the node with much higher energy residue as the next hop. Therefore,

ER$_{ii}$	ER$_{ii}$				
ER$_{ij}$	ER$_{i1}$	ER$_{i2}$	ER$_{i\cdots}$	ER$_{ij}$	ER$_{i\cdots}$ $j \in N_i$
θ_{jid}	θ_{1id}	θ_{2id}	$\theta_{\cdots id}$	θ_{jid}	$\theta_{\cdots id}$ $j \in N_i^R$
β_{ipj}	β_{ip1}	β_{ip2}	$\beta_{ip\cdots}$	β_{ipj}	$\beta_{ip\cdots}$ $j \in N_i^L$
L_{ij}	L_{i1}	L_{i2}	$L_{i\cdots}$	L_{ij}	$L_{i\cdots}$ $j \in N_i$

Figure 5: Conserved data structures.

we replace the P_j by $k \times \mathrm{ER}_j + (1 - k)L_{ij} \cos\theta_{jid}$. Since P_i is the base pressure, we replace $\Delta P_{ij} = P_j - P_i$ by P_j through omitting P_i. Using (4), we have

$$Q_{ij} = \frac{\Phi_{ij} \times \left[k \times \mathrm{ER}_j + (1 - k)\, L_{ij} \cos\theta_{jid}\right]}{L_{ij}^{\alpha}}, \tag{7}$$

where Q_{ij} is the virtual communication fluxes through the wireless link ij; Φ_{ij} is the link quality; ER_j is the energy residue of node j; L_{ij} is the Euclidean distance of nodes i and j; α is path-loss exponent; θ_{jid} is the angle of deviation and its range is $[-\pi/2, \pi/2]$; k is a proportionality factor which uses to adjust the weight of ER_j and $L_{ij} \cos\theta_{jid}$.

Then, we discuss how to choose the proper next hop. As related in Section 3.2, since $\Delta P_{ij}^1 = \Delta P_{ij}^2$ and L_{ij}^1 and L_{ij}^2 are kept constant throughout the adaptation process in contrast to D_{ij}, the PiPf can only achieve the adaptation by the evolution of $D_{ij}(t)$. In our scenario, node i chooses the next hop form candidates as shown in Figure 4(b). Since (1) $\Phi_{ij}^1 = \Phi_{ij}^2$ and L_{ij}^1 and L_{ij}^2 are kept constant according to the assumptions and (2) ΔP_{ij_1} and ΔP_{ij_2} are different and time-varying, we can achieve the adaptation by the evolution of $\Delta P_{ij}(t)$. If letting the monotonically increasing continuous function $\varphi(Q) = Q^{\mu}$, we have

$$\frac{d}{dt}\Delta P_{ij} = \varphi\left(|Q|\right) - \delta\Delta P_{ij} = \left(\frac{\Phi_{ij}\Delta P_{ij}}{L_{ij}^{\alpha}}\right)^{\mu} - \delta\Delta P_{ij}, \tag{8}$$

where δ is a decay rate of ΔP_{ij} and μ is a constant satisfying $\mu > 0$. We use (8) to determine the next hop in our P-iSNH; namely, we choose the node whose $(d/dt)\Delta P_{ij}$ is maximal as the next hop.

4. P-iSNH Based Routing Strategy and Algorithm

4.1. Data Conserved. In this section, we introduce the data which should be conserved in each node. Because of the same characteristic of each node, we suppose that the Φ_{ij} of each link is the same and ignore it to simplify discussion.

Each node i needs to conserve the following information: L_{ij} ($j \in N_i$), ER_{ij} ($j \in N_i$), ER_{ii}, $|\theta_{jid}|$ ($j \in N_i^R$), and β_{ipj} ($j \in N_i^L$) which are shown in Figure 5, where node p

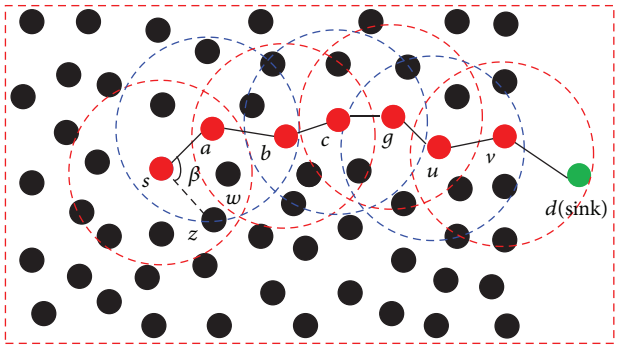


Figure 6: Process of routing selection.

is the previous hop of node i, ER_{ii} represents the ER_i stored in node i, and ER_{ij} represents the ER_j stored in node i. As our WSNs are location aware, the L_{ij}, θ_{jid}, and β_{ipj} are easily acquired following from (1) and (2). Note that the nodes in our WSNs are static, and we only need to calculate the L_{ij}, θ_{jid} and β_{ipj} once at WSNs deployment time. For the difference of ER and L_{ij}, we normalize them to $\widehat{\mathrm{ER}}$ and $\hat{L}_{ij}$, respectively. Therefore, we obtain

$$Q_{ij} = \frac{k \times \widehat{\mathrm{ER}}_j + (1-k)\,\hat{L}_{ij}\cos\theta_{jid}}{\hat{L}_{ij}^{\alpha}} \tag{9}$$

$$\frac{d}{dt}\Delta P_{ij} = \left(\frac{k \times \widehat{\mathrm{ER}}_j + (1-k)\,\hat{L}_{ij}\cos\theta_{jid}}{\hat{L}_{ij}^{\alpha}}\right)^{\mu} - \delta\left[k \times \widehat{\mathrm{ER}}_j + (1-k)\,\hat{L}_{ij}\cos\theta_{jid}\right]. \tag{10}$$

4.2. Routing Strategy. If node s needs to send data to the sink d, it searches for a routing in the following method. We illustrate the routing strategy combined with Figure 6.

Step 1. Each $(d/dt)\Delta P_{sj}$ $(j \in N_s^{\mathrm{R}})$ is calculated following from (10), where θ_{jsd}, $\widehat{\mathrm{ER}}_{sj}$, and $\hat{L}_{sj}$ are conserved and stored in node s beforehand.

Step 2. Each node $j \in N_s^R$ is saved into a temporary array variable *Temp* in descending order by $(d/dt)\Delta P_{sj}$.

Step 3. The first node in *Temp* is picked as the next hop of the routing.

Step 4. If the next hop a of node s satisfies $N_a^R = \emptyset$, namely, there is an energy hole in the right side of node a, the node a will not send *ACK* to s. Then, the node s will trigger a specific processing routine.

Step 5. If $|Temp[0]\cdot\theta_{jsd} - Temp[1]\cdot\theta_{jsd}| \geq \pi/2$, node s will choose the node $Temp[1]$ as the next hop. Then, the regular processing routine is going on.

Step 6. Otherwise, each $(d/dt)\Delta P_{sj}$ $(j \in N_s^L)$ is calculated following from (11) and the nodes are saved into the *Temp* in ascending order by $(d/dt)\Delta P_{sj}$. Then, the first node in *Temp* is chosen as the next hop of the routing and the regular processing routine is going on

$$\frac{d}{dt}\Delta P_{ij} = \left(\frac{k \times \widehat{\mathrm{ER}}_j + (1-k)\,\hat{L}_{ij}\beta_{asj}}{\hat{L}_{ij}^{\alpha}}\right)^{\mu} - \delta\left[k \times \widehat{\mathrm{ER}}_j + (1-k)\,\hat{L}_{ij}\beta_{asj}\right], \tag{11}$$

where β_{asj} is the angle of line sa and line sj. Equation (11) indicates that it tends to choose a node which sharply deviates from the failing node, for example, a, as the next hop to avoid entering the energy hole again.

Step 7. The process is repeated, like a rolling wheel, until the sink d is found.

4.3. Routing Algorithms. Given the data conserved and routing process in the preceding sections, the P-iRP's algorithms of initialization, regular processing routine, receiving routine, and specific processing routine are described by Algorithm 1, Algorithm 2, Algorithm 3, and Algorithm 4, respectively.

Based on the WSNs scenario in Section 3.1, suppose that the degree of graph G is $D(G)$ which can be regarded as a constant, the complexity of Algorithm 1, Algorithm 2, Algorithm 3, and Algorithm 4 are $O(n \times D(G)^2) = O(n)$, $O(D(G)) = O(1)$, $O(1)$, and $O(D(G)) = O(1)$, respectively. From Figure 2, the number of intermediate nodes from node s to sink d is approximately $\sqrt{5n}/2$ in the worst case. Note that Algorithm 1 is run only once at WSNs deployment time. Therefore, the complexities of P-iRP is $O(\sqrt{5n}/2) = O(\sqrt{n})$ in running time, which greatly reduces the processing delay and saves the energy of sensors.

5. P-iSNH Analysis

In this section, we analyze the feasibility of P-iSNH by mathematical theoretical analysis. We study the cases in which two nodes connected to the same node compete to be the next hop, as shown in Figure 4(b).

There are four nodes i, j_1, j_2, and sink d. For simplicity, we hereafter replace $L_{ij_1}, L_{ij_2}, Q_{ij_1}, Q_{ij_2}, \Delta P_{ij_1}$, and ΔP_{ij_2} by L_1, L_2, Q_1, Q_2, ΔP_1, and ΔP_2, respectively. In multipath routing, the virtual fluxes along each path are calculated as

$$Q_1 = \frac{\Delta P_1/L_1^{\alpha}}{\Delta P_1/L_1^{\alpha} + \Delta P_2/L_2^{\alpha}}$$
$$Q_2 = \frac{\Delta P_2/L_2^{\alpha}}{\Delta P_1/L_1^{\alpha} + \Delta P_2/L_2^{\alpha}}. \tag{12}$$

Since Q_1 and Q_2 are nonnegative, adaptation equation (8) becomes

$$\frac{d}{dt}(\Delta P_1) = \varphi(Q_1) - \delta\cdot\Delta P_1$$
$$\frac{d}{dt}(\Delta P_2) = \varphi(Q_2) - \delta\cdot\Delta P_2. \tag{13}$$

```
(1)  for each node i do
(2)    for each node j(j ∈ N_i^L) do
(3)      R_ij = 1;
(4)      if (L_ij is not initialized)
(5)        initializing and normalizing L_ij and L_ji following (1);
(6)      end if
(7)      for each node k(k ∈ N_j^R) do
(8)        calculating β_ijk following (2);
(9)      end for
(10)   end for
(11)   for each node j(j ∈ N_i^R) do
(12)     R_ij = 1;
(13)     calculating θ_jid following (2);
(14)     if (L_ij is not initialized)
(15)       initializing and normalizing L_ij and L_ji following (1);
(16)     end if
(17)   end for
(18)   R_ii = 1;
(19) end for
```

Algorithm 1: Initialization.

```
(1) for each node j(j ∈ N_i^R) do
(2)   calculating (d/dt)ΔP_ij following from (10);
(3)   save j and (d/dt)ΔP_ij into Temp in descending order by (d/dt)ΔP_ij;
(4) end for
(5) P · SA = i;
(6) P · NA = Temp[0];
(7) P · ER = R_ii;
(8) send P;
```

Algorithm 2: Regular processing.

Setting $\varphi(Q) = Q^{\mu}$, $(d/dt)(\Delta P_1) = 0$, and $(d/dt)(\Delta P_2) = 0$, we have

$$\begin{aligned}\left(\frac{\Delta P_1/L_1^{\alpha}}{\Delta P_1/L_1^{\alpha} + \Delta P_2/L_2^{\alpha}}\right)^{\mu} &= \delta \cdot \Delta P_1\\ \left(\frac{\Delta P_2/L_2^{\alpha}}{\Delta P_1/L_1^{\alpha} + \Delta P_2/L_2^{\alpha}}\right)^{\mu} &= \delta \cdot \Delta P_2.\end{aligned} \tag{14}$$

After some calculations, we obtain

$$\begin{aligned}\Delta P_1 &= \frac{1}{\delta}\left[\frac{1}{\left(1 + \left(L_1^{\alpha}/L_2^{\alpha}\right)^{1/1-\mu}\right)}\right]^{\mu}\\ \Delta P_2 &= \frac{1}{\delta}\left[\frac{1}{\left(1 + \left(L_2^{\alpha}/L_1^{\alpha}\right)^{1/1-\mu}\right)}\right]^{\mu}.\end{aligned} \tag{15}$$

Namely, there is an equilibrium point given by $(\Delta P_1, \Delta P_2)$. We perform the simulation using MATLAB by setting the parameters $\alpha = 2$, $\mu = 0.8$, $\delta = 0.3$, $L_1 = 10$ and $L_2 = 12$ following from (14), and the solutions are shown in Figure 7, where two curves intersect in a point E which superpose on the equilibrium point $(\Delta P_1, \Delta P_2)$.

We present a linear stability analysis at the equilibrium point in before parameters. The Jacobi matrix J on the right-hand side of (13) is calculated as

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix}, \tag{16}$$

where

$$J_{11} = \frac{\mu {Q_1}^{\mu-1} \Delta P_2/L_2^{\alpha}}{L_1^{\alpha}\left(\Delta P_1/L_1^{\alpha} + \Delta P_2/L_2^{\alpha}\right)^2} - \delta$$

$$J_{12} = \frac{-\mu {Q_1}^{\mu-1} \Delta P_1/L_1^{\alpha}}{L_2^{\alpha}\left(\Delta P_1/L_1^{\alpha} + \Delta P_2/L_2^{\alpha}\right)^2}$$

```
(1)  while (no receiving wireless radio)
(2)     node i sleep;
(3)  end while
(4)  wake node i;
(5)  R_iP·SA = P · ER;
(6)  if (i! = P · NA)
(7)     go on sleeping;
(8)  else
(9)     receiving packet P;
(10) end if
```

Algorithm 3: Receiving processing.

```
(1)  if (not receive ACK from next hop before deadline)
(2)     if (|Temp[0] · θ_jsd − Temp[1] · θ_jsd| ≥ π/2)
(3)        P · SA = i;
(4)        P · NA = Temp[1];
(5)        P · ER = R_ii;
(6)        send P;
(7)     else
(8)        for each node j (j ∈ N_i^L) do
(9)           calculating (d/dt)ΔP_ij following (11);
(10)          save j and (d/dt)ΔP_ij into Temp in ascending order;
(11)       end for
(12)       P · SA = i;
(13)       P · NA = Temp[0] · j;
(14)       P · ER = R_ii;
(15)       send P;
(16)    end if
(17) end if
```

Algorithm 4: Specific processing.

$$J_{21} = \frac{-\mu {Q_2}^{\mu-1} \Delta P_2 / L_2^{\alpha}}{L_1^{\alpha} \left(\Delta P_1 / L_1^{\alpha} + \Delta P_2 / L_2^{\alpha}\right)^2}$$
$$J_{22} = \frac{\mu {Q_2}^{\mu-1} \Delta P_1 / L_1^{\alpha}}{L_2^{\alpha} \left(\Delta P_1 / L_1^{\alpha} + \Delta P_2 / L_2^{\alpha}\right)^2} - \delta \tag{17}$$

and the Jacobi matrix at equilibrium point E is denoted $J(E)$. After some calculations, the following formula is obtained

$$J(E) = \begin{pmatrix} \delta\left(\mu Q_2^* - 1\right) & -\delta\mu \dfrac{L_1^{\alpha}}{L_2^{\alpha}} Q_1^* \\ -\delta\mu \dfrac{L_2^{\alpha}}{L_1^{\alpha}} Q_2^* & \delta\left(\mu Q_1^* - 1\right) \end{pmatrix} = \delta \cdot \begin{pmatrix} \mu Q_2^* - 1 & -\mu \dfrac{L_1^{\alpha}}{L_2^{\alpha}} Q_1^* \\ -\mu \dfrac{L_2^{\alpha}}{L_1^{\alpha}} Q_2^* & \mu Q_1^* - 1 \end{pmatrix}, \tag{18}$$

where Q_1^* and Q_2^* are virtual communication fluxes along the first and second wireless links at the equilibrium point E. Using the relation $Q_1^* + Q_2^* = I$, we have

$$\det J(E) = \delta\left(1 - \mu I\right), \qquad \operatorname{tr} J(E) = \delta\left(\mu I - 2\right). \tag{19}$$

Note that δ is the decay rate of ΔP_{ij} and $\delta > 0$. If we let $I = 1$, thus,

$$\begin{gathered} \det J(E) < 0 \quad \text{for } \mu > 1, \\ \det J(E) > 0 \operatorname{tr} J(E) < 0 \quad \text{for } 0 < \mu < 1. \end{gathered} \tag{20}$$

Since we set $\mu = 0.8$, this means that the equilibrium point E is stable. Therefore, the routing of WSNs will reach to equilibrium with our P-iSNH, which is very important to a routing strategy.

6. Simulation Results

We design a simulation platform using C++ to validate P-iRP. In the simulation, 441 sensors are relatively regularly

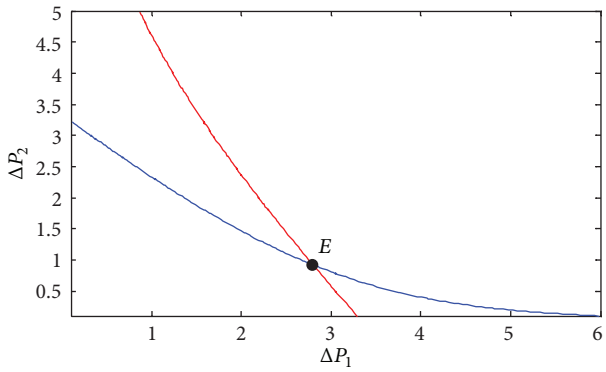


FIGURE 7: Simulation of the solution. In $\alpha = 2$, $\mu = 0.8$, $\delta = 0.3$, $L_1 = 10$, and $L_2 = 12$, two curves which come from (14) intersect in a point E, which is the sole equilibrium point.

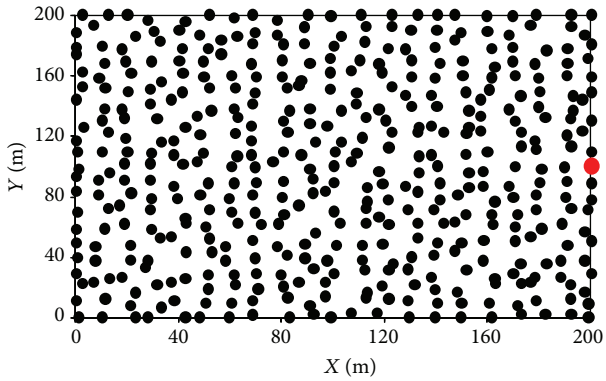


FIGURE 8: Sensor nodes deployment.

deployed in the field of 200 m × 200 m, and the sink node is deployed in the right of the field, shown in Figure 8. The sensing radius of each sensor is 30 m, the original energy of each node is 100, and the energy of sink node is inexhaustible. We suppose that the energy consumption of one transmission is 1, if the transmission distance is 20 m. Therefore, the energy consumption of one transmission of two nodes i and j is $(L_{ij}/20)^{\alpha}$, where α is set to 2.

In order to validate the energy equilibrium, we only choose the nodes in the field of [(0, 0), (1 m, 1 m)] to transmit data to the sink. If a chosen node transmits a group of data to sink, the P-iRP is used to choose next hops until the sink is found, which is called a *round*. This iterative process will halt after n rounds until WSNs break down. We run GPSR, GEAR ($k = 0.5, 0.9$) and P-iRP ($k = 0.5, 0.9$) 10 times, respectively, to acquire their average value and compare them, where we use k replace α which is used in GEAR to bring into correspondence with P-iRP. If the distance between the nodes and sink is less than 30 m, we let the nodes directly transmit data to sink to quicken convergence of P-iRP, and the energy consumption is set to 1. From Figure 8, there are 18 sensors around the sink. Therefore, the ideal number of rounds of simulation process is $1800/(L_{ij}/20)^{\alpha}$.

6.1. Energy Equilibrium of P-iRP. Figure 9 illustrates the energy distribution of GPSR, GEAR ($k = 0.5$), and P-iRP ($k = 0.5$) in different rounds. We can infer that (1) the energy distributions of GPSR are very imbalanced, (2) the energy distributions of GEAR and P-iRP are rather balanced, and (3) the energy distributions of P-iRP are more balanced than those of GEAR.

Figure 10 illustrates the lifetime of WSNs. In GPSR, the first dead node emerges in round of 192, and the WSNs break down in round of 889. In GEAR ($k = 0.5$), the first dead node emerges in round of 910, and the WSNs break down in round of 1223. In P-iRP ($k = 0.5$), the first dead node emerges in round of 1112, and the WSNs break down in round of 1396. In GEAR ($k = 0.9$), the first dead node emerges in round of 1428, and the WSNs break down in round of 1592. In P-iRP ($k = 0.9$), the first dead node emerges in round of 1542, and the WSNs break down in round of 1696. Therefore, the lifetime of GEAR ($k = 0.5$) is 48.8% longer than that of GPSR; the lifetime of P-iRP ($k = 0.5$) is 14.2% longer than that of GEAR ($k = 0.5$); and the lifetime of P-iRP ($k = 0.9$) is 6.5% longer than that of GEAR ($k = 0.9$). From Figure 9 and Figure 10, we can differ that (1) whether considering energy residue of next hops or not will impacts on the lifetime of WSNs greatly, and (2) in energy balanced WSNs, the time period is very short from emerging dead nodes to networks breaking down because all nodes reach to exhausted status of energy in the same time period.

Figure 11 illustrates the dead nodes distributions of GEAR ($k = 0.5$) and P-iRP ($k = 0.5$) in the rounds of 1380. The results show that P-iRP ($k = 0.5$) has much less dead nodes than GEAR ($k = 0.5$). We can also differ that the dead nodes of both algorithms are converged on a specific field but do not spread around the entire range of WSNs, which is useful in deploying such WSNs to prolong the lifetime.

The reasons to gain the results of Figures 9, 10, and 11 are that (1) since GPSR does not take energy into account, it utilizes frequently the "hot" nodes to result in imbalanced energy distributions, (2) since GEAR and P-iRP consider both energy and location of nodes, their energy distributions are rather balanced, and (3) since P-iRP is more elaborate in energy utilization than GEAR, the energy distributions of P-iRP are more balanced than those of GEAR.

6.2. Efficiency of P-iRP. Figure 12 illustrates the number of hops that the different algorithms need in different rounds of transmission. By calculating, the average hops of GPSR, GEAR ($k = 0.5$), P-iRP ($k = 0.5$), GEAR ($k = 0.9$), and P-iRP ($k = 0.9$) are 19.4, 24.3, 21.6, 28.9, and 26.8, respectively.

In case of $k = 0.5$, the average hops of P-iRP are 11.3% more than those of GPSR, and the hops of GEAR are 23.3% more than those of GPSR. Combined with Figure 10, the increment of average hops of 11.3% will lead to the increment of lifetime of more than 60% from GPSR to P-iRP, while the increment of average hops of 23.2% will only lead to the increment of lifetime of about 48% from GPSR to GEAR. Therefore, the P-iRP is more efficient in balance of routing efficiency and energy equilibrium than GEAR.

In case of $k = 0.9$, the average hops of P-iRP are 38.1% more than those of GPSR, and the hops of GEAR are 49.0%

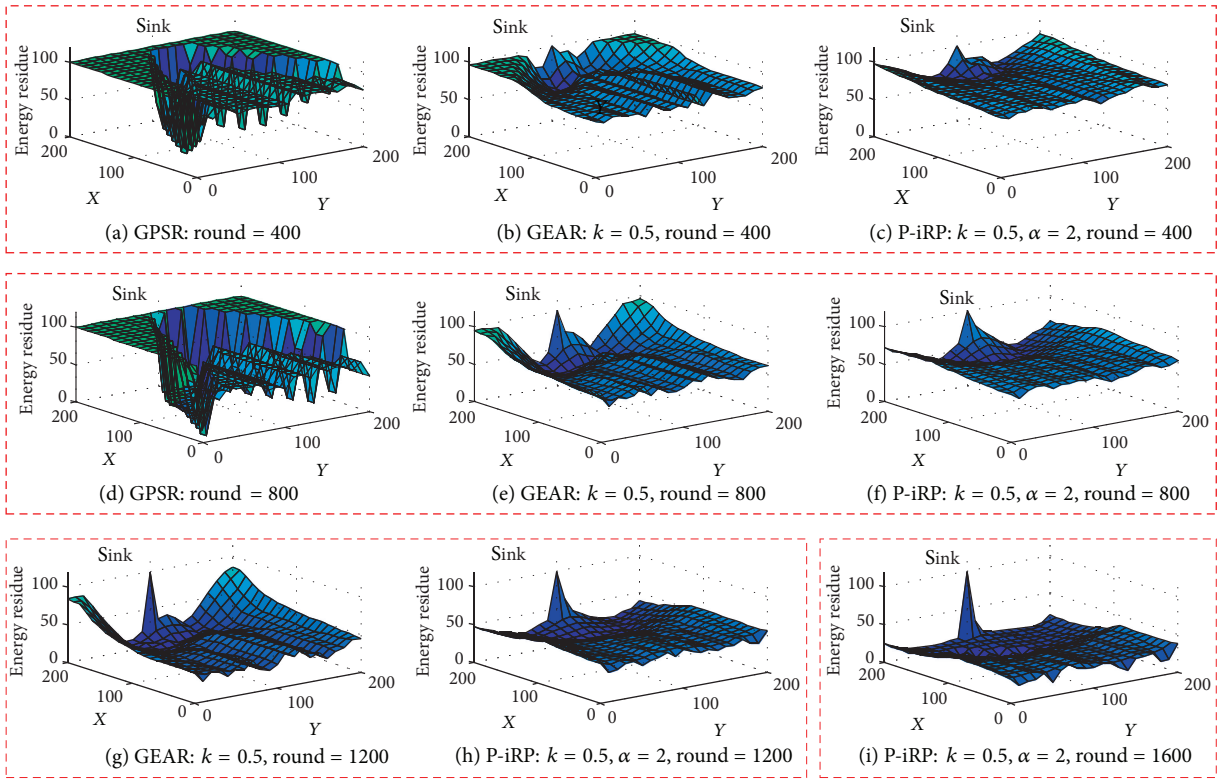


Figure 9: Hops of transmission of each round.

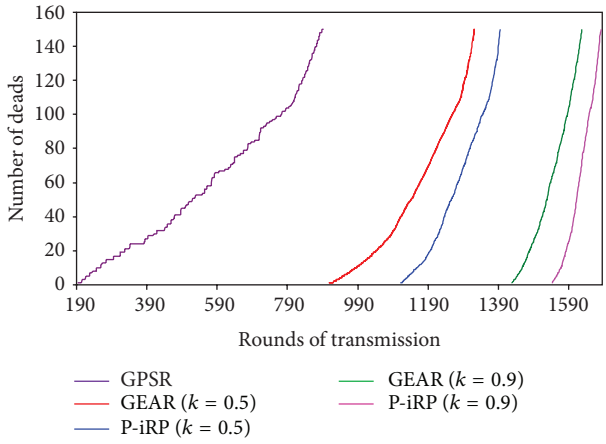


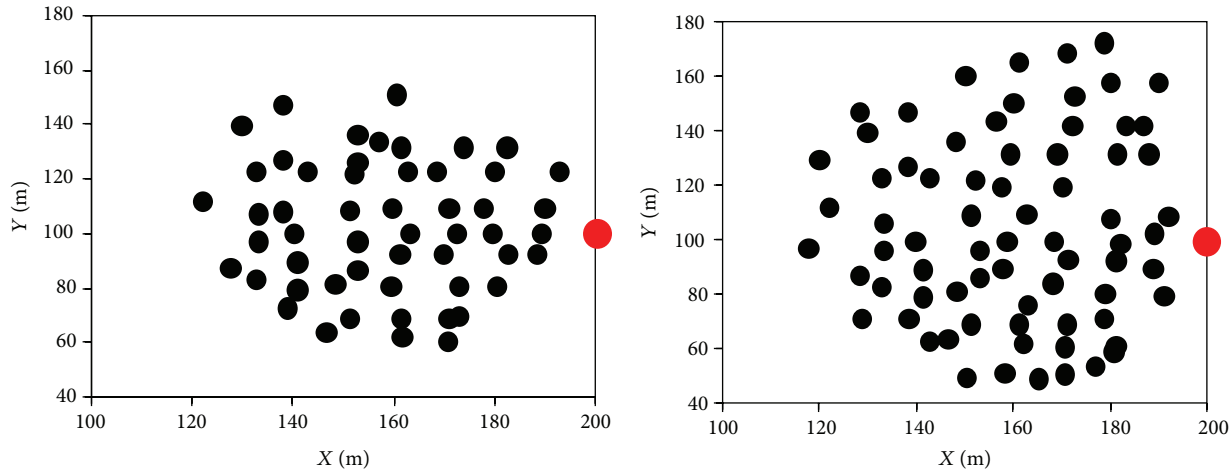
Figure 10: Lifetimes of WSNs.

more than those of GPSR. Combined with Figure 10, the increment of 11.3% of average hops will lead to the increment of about 60% of lifetime from GPSR to P-iRP (k = 0.5), while the increment of 24% of average hops will only cause the increment of 21.5% of lifetime from P-iRP (k = 0.5) to P-iRP (k = 0.9). That is to say, the larger k is, the smaller the increment of k impacts on lifetime of WSNs. Therefore, it is improper to set a larger k, so does GEAR.

In addition, Figure 12 implies that (1) GPSR can get much higher routing efficiency than those of GEAR and P-iRP at initial periods of time in WSNs' lifetime, but its efficiency decreases exponentially with the time going, and (2) the routing efficiencies of GEAR and P-iRP are almost the same at initial periods of time in WSNs' lifetime, while the difference becomes gradually large with the time going. The reasons to gain those results are that (1) since GPSR does not take energy into account, the routing efficiency will decrease exponentially after dead nodes emerge and (2) P-iRP can outperform GEAR in routing efficiency, since P-iRP considers comprehensively the distance, energy residue, and location of the next hop, other than only considering energy and location in GEAR.

7. Conclusion

The *Physarum* forages for patchily distributed food sources through accommodating its body to form networks with comparable efficiency, fault tolerance, and cost, which is the source of P-iRP's inspiration. For the proposed scenario, the P-iRP ensures the passage of data packets through one by one static sensor nodes to reach the sink. In each intermediate node, the P-iSNH is used to choose the proper next hop. Once an energy hole emerges, a specific processing will be triggered to bypass the hole. The theoretical analysis and simulation results show that the P-iRP possesses many advantages, for example, rather low algorithm complexity for P-iRP, ever-present equilibrium solution for P-iSNH, and effective trade-off between routing efficiency and energy equilibrium compared to other famous algorithms, which greatly reduces the processing delay and saves the sensors' energy and also demonstrates that the P-iRP is applicable to the proposed



(a) P-iRP, $k = 0.5$, round = 1380

(b) GEAR, $k = 0.5$, round = 1380

Figure 11: Distributions of dead sensor nodes.

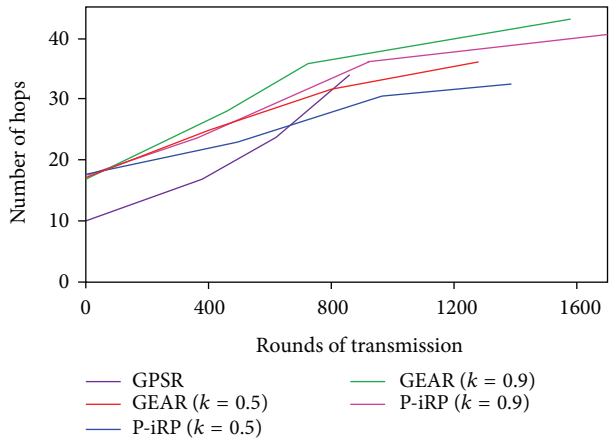


Figure 12: Hops of transmission of each round.

scenario. Furthermore, we consider that the model may also provide a useful help to develop the routing protocol in mobile ad hoc networks, which will be our future focus.

Acknowledgments

This work was partially supported by the National High-Tech Research and Development Program of China (863 Program) under Grant no. 2011AA010701, in part by the National Basic Research Program of China (973 Program) under Grant no. 2013CB329102, in part by the National Natural Science Foundation of China (NSFC) under Grant nos. 61003283, 61001122, 61232017, U1204614, 61003035, and 61142002, and in part by the Natural Science Foundation of Jiangsu Province under Grant no. BK2011171.

References

[1] S. Li, Y. Lou, and B. Liu, "Bluetooth aided mobile phone localization: a nonlinear neural circuit approach," *Transactions on Embedded Computing Systems*, 2013.

[2] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, New York, NY, USA, August 2000.

[3] F. Kuhn, R. Wattenhofer, and A. Zollinger, "An algorithmic approach to geographic routing in ad hoc and sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 51–62, 2008.

[4] K. Padmanabh, A. M. Reddy V., S. Sen, and P. Gupta, "Random Walk on Random Graph based Outlier detection in wireless sensor networks," in *Proceedings of the 3rd International Conference on Wireless Communication and Sensor Networks (WCSN '07)*, pp. 45–49, Allahabad, India, December 2007.

[5] G. Trajcevski, F. Zhou, R. Tamassia, and B. Avii, "On the construction of data aggregation tree with minimum energy cost in wireless sensor networks: NP-completeness and approximation algorithms," in *Proceedings of the INFOCOM*, pp. 2591–2595, Orlando, Fla, USA, 2012.

[6] M. H. Chaudhary and L. Vandendorpe, "Battery-aware power allocation for lifetime maximization of wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, Cape Town, South Africa, May 2010.

[7] J. Zhang, S. Ci, H. Sharif, and M. Alahmad, "Lifetime optimization for wireless sensor networks using the nonlinear battery current effect," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, Dresden, Germany, June 2009.

[8] C.-K. Chau, F. Qin, S. Sayed, M. H. Wahab, and Y. Yang, "Harnessing battery recovery effect in wireless sensor networks: experiments and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1222–1232, 2010.

[9] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Transactions on Communications*, vol. 60, no. 1, pp. 220–230, 2012.

[10] S. Bai, W. Zhang, G. Xue, J. Tang, and C. Wang, "DEAR: delay-bounded energy-constrained adaptive routing in wireless sensor networks," in *Proceedings of the INFOCOM*, pp. 1593–1601, Shanghai, China, 2012.

[11] Y. Yu, R. Govindan, and D. Estrin, "Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks," Tech. Rep. UCLACSD TR-01-0023, UCLA Computer Science Department, 2001.

[12] N. Roseveare and B. Natarajan, "Energy-aware distri*f*buted tracking in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '11)*, pp. 363–368, Cancun, Mexico, March 2011.

[13] S. Li, Z. Wang, and Y. Li, "Using Laplacian Eigenmap as heuristic information to solve nonlinear constraints defined on a graph and its application in distributed range-free localization of wireless sensor networks," *Neural Processing Letters*, vol. 37, no. 3, pp. 411–424, 2012.

[14] O. Yang and W. Heinzelman, "Sleeping multipath routing: a trade-off between reliability and lifetime in wireless sensor networks," in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference (GLOBECOM '11)*, Houston, Tex, USA, December 2011.

[15] F. Sivrikaya, T. Geithner, C. Truong, M. A. Khan, and S. Albayrak, "Stochastic routing in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications Workshops (ICC '09)*, Berlin, Germany, June 2009.

[16] M. Zhang, C. Xu, J. Guan, R. Zheng, Q. Wu, and H. Zhang, "P-iRP: physarum-inspired routing protocol for wireless sensor networks," in *Proceedings of the VTC*, Las Vegas, Nev, USA, 2013.

[17] C. Xu, T. Liu, J. Guan, H. Zhang, and G.-M. Muntean, "CMT-QA: quality-aware adaptive concurrent multipath data transfer in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, 2012.

[18] Y. Cao, C. Xu, J. Guan, F. Song, and H. Zhang, "Environment-aware CMT for efficient video delivery in wireless multimedia sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 381726, 12 pages, 2012.

[19] R. Zheng, M. Zhang, Q. Wu, S. Sun, and J. Pu, "Analysis and application of bio-inspired multi-net security model," *International Journal of Information Security*, vol. 9, no. 1, pp. 1–17, 2010.

[20] C. Xu, F. Zhao, J. Guan, H. Zhang, and G.-M. Muntean, "QoE-driven user-centric VoD services in urban multi-homed P2P-based vehicular network," *IEEE Transactions on Vehicular Technology*, 2012.

[21] S. Li, M. Q. H. Meng, W. Chen et al., "SP-NN: a novel neural network approach for path planning," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO '07)*, pp. 1355–1360, Sanya, China, December 2007.

[22] J. Rao and A. Fapojuwo, "A battery aware distributed clustering and routing protocol for wireless sensor networks," in *Proceedings of the WCNC*, pp. 1538–1543, Shanghai, China, 2012.

[23] C.-L. Wang and S.-J. Syue, "An efficient relay selection protocol for cooperative wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '09)*, Budapest, Hungary, April 2009.

[24] G. Trajcevski, F. Zhou, R. Tamassia, B. Avci, P. Scheuermann, and A. Khokhar, "Bypassing holes in sensor networks: load-balance versus latency," in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference (GLOBECOM '11)*, Houston, Tex, USA, December 2011.

[25] T. Nakagaki, H. Yamada, and Á. Tóth, "Maze-solving by an amoeboid organism," *Nature*, vol. 407, no. 6803, p. 470, 2000.

[26] A. Tero, S. Takagi, T. Saigusa et al., "Rules for biologically inspired adaptive network design," *Science*, vol. 327, no. 5964, pp. 439–442, 2010.

[27] A. Tero, R. Kobayashi, and T. Nakagaki, "A mathematical model for adaptive transport network in path finding by true slime mold," *Journal of Theoretical Biology*, vol. 244, no. 4, pp. 553–564, 2007.

[28] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the ACM Mobicom*, pp. 166–179, New York, NY, USA, 2001.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 203719, 6 pages
http://dx.doi.org/10.1155/2013/203719

Review Article

Circle Fitting Using a Virtual Source Localization Algorithm in Wireless Sensor Networks

Junli Liang,[1,2] Miaohua Zhang,[1] Xianju Zeng,[3] Kexin Zhao,[2] and Jian Li[2]

[1] *Xi'an University of Technology, Xi'an 710048, China*
[2] *University of Florida, Gainesville, FL 32611, USA*
[3] *College of Management, Shenzhen University, Shenzhen 518060, China*

Correspondence should be addressed to Junli Liang; heery_2004@hotmail.com

Received 16 August 2012; Accepted 14 January 2013

Academic Editor: Arunita Jaekel

A novel circle fitting algorithm is proposed in this paper. The key points of this paper are given as follows: (i) it formulates the circle fitting problem into the special source localization one in wireless sensor networks (WSN); (ii) the multidimensional scaling (MDS) analysis is applied to the data points, and thus the propagator-like method is proposed to represent the circle center parameters as the functions of the circle radius; (iii) the virtual source localization model can be rerepresented as special nonlinear equations of a unique variable (the circle radius) rather than the original three ones (the circle center and radius), and thus the classical fixed-point iteration algorithm is applied to determine the radius and the circle center parameters. The effectiveness of the proposed circle fitting approach is demonstrated using the simulation and experimental results.

1. Introduction

Circle fitting receives considerable attention because it plays an important role in computer vision, observational astronomy, structural geology, industry inspection, medical diagnosis, Iris recognition, military, security, and so forth [1–8]. For instance, to meet the increasing demand for manufacturing automation, the circle fitting technique is often applied to measure the diameter of the processing product in the manufacturing systems.

The fitting problem can be viewed as follows: estimate the parameters of a circle from a set of coplanar points. Several classical approaches [1–8], including the Hough transform (HT) methods [4, 5] and the least square (LS) approaches [6–8], have been developed to solve this problem. The former are actually to carry out a voting procedure in a three-dimensional (3D) Hough accumulator space, where every point represents a circle of a certain size. The corresponding coordinate of the local maxima is obtained as the estimated parameters of the circle. In comparison, the latter attempt to find the parameters of a circle by minimizing an error metric between the primitive and the data points.

In this paper, we develop a novel circle fitting approach by borrowing the idea from source localization in wireless sensor networks (WSN) [9, 10]. It is worthwhile to highlight the main contributions of this paper here.

(i) It formulates the circle fitting problem into special source localization one in WSN, where each data point should be understood as an abstract sensor node in sensor networks, and the circle center represents the localized target. However, the propagation delays are unknown, and thus the existing source localization algorithms in WSN cannot be applied to solve the special source localization problem.

(ii) The multidimensional scaling (MDS) analysis [11] is applied to the data points, and a special covariance-like matrix is constructed. Thus, we propose the propagator-like method to represent the circle center as the functions of the circle radius.

(iii) The virtual source localization model can be rerepresented as special nonlinear equations, where the radius is the unique variable rather than the original three ones (the circle center and radius), and thus the

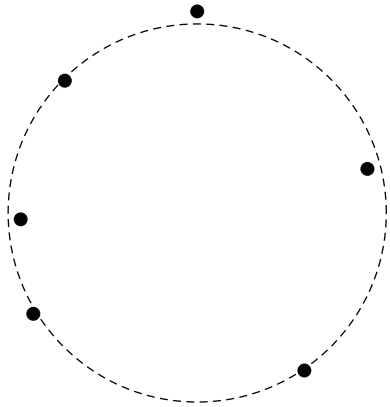


Figure 1: Circle fitting problem description.

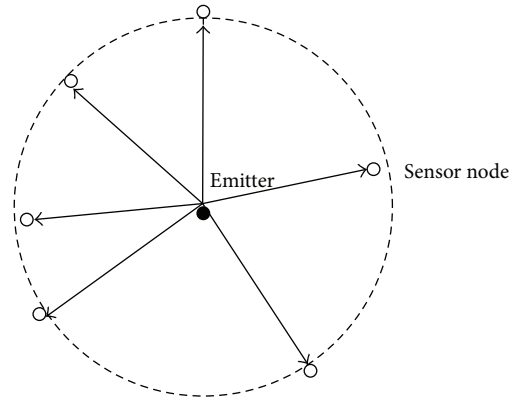


Figure 2: Virtual source localization in WSN.

classical fixed-point iteration algorithm [12] is applied to determine the radius and circle center.

The rest of this paper is organized as follows. The circle fitting problem is described in Section 2. A novel circle fitting approach is developed in Section 3. Simulated and experimental results are presented in Section 4. The paper is concluded in Section 5.

2. Problem Formulation

The equation for a circle centered at (x_0, y_0) with a radius r in (x, y) coordinates has the following form:

$$\frac{(x - x_0)^2}{r^2} + \frac{(y - y_0)^2}{r^2} = 1. \tag{1}$$

The circle fitting problem (CFP) [1–8] can be described in Figure 1, that is, given data points (x_i, y_i), $i = 1, \ldots, I$, the objective of circle fitting is to estimate circle parameters (r, x_0, y_0) that best fit to these data points. However, since in the actual application noise is introduced by some operations (e.g., the segmentation and edge detection operations in the image processing application), these obtained points are not completely precise, that is,

$$\frac{(x_i - x_0)^2}{r^2} + \frac{(y_i - y_0)^2}{r^2} = 1 + v_i, \tag{2}$$

where v_i is the introduced noise. The objective of this paper is to estimate the circle parameters (r, x_0, y_0) from the given data points (x_i, y_i), $i = 1, \ldots, I$.

3. Proposed Algorithm

In this section, we first reformulate CFP into a virtual source localization problem in wireless sensor networks (WSN) [9, 10] and then develop a novel circle fitting algorithm in this framework.

Let us review the source localization model in WSN [9, 10]:

$$\frac{1}{c}\left\|\mathbf{s} - \mathbf{x}_i\right\| = \tau_i + w_i = \widetilde{\tau}_i, \tag{3}$$

where c is the propagation speed of light; $\mathbf{s}$ is the unknown location of the emitter; $\mathbf{x}_i$ is the location of the ith sensor node in wireless sensor networks; τ_i is the (ideal) signal propagation delay from the target to the ith sensor node, but $\widetilde{\tau}_i$ is the available delay measurement, containing noise w_i. In addition, $\| \bullet \|$ denotes the Euclidean norm of vector $\bullet$. The objective of source localization in WSN is to estimate $\mathbf{s}$ from the given measurement delays $\widetilde{\tau}_i$, $i = 1, \ldots, I$.

To reformulate CFP into the source localization problem in WSN, we rewrite (2) in another form as

$$\begin{aligned} &(x_i - x_0)^2 + (y_i - y_0)^2 = r^2 + r^2 v_i \\ &\Longrightarrow \left\|\mathbf{s} - \mathbf{x}_i\right\| = r + n_i, \quad i = 1, \ldots, I, \end{aligned} \tag{4}$$

where $n_i = r(\sqrt{1 + v_i} - 1)$, $\mathbf{s} = [x_0 \; y_0]$, $\mathbf{x}_i = [x_i \; y_i]$, and $i = 1, \ldots, I$.

The source localization model in (3) is quite similar to the circle fitting model in (4), especially when $c = 1$: (i) each data point should be understood as a virtual "sensor node" in sensor networks; (ii) the circle center represents the virtual "emitter" or localized "target"; and (iii) the circle radius r is the virtual "propagation delay", which are clearly described in Figure 2.

By comparing (3) with (4), we can easily observe their differences; that is, all "propagation delays" from the "target" to "sensor nodes" $\widetilde{\tau}_i$ or τ_i are unknown. Therefore, the existing source localization algorithms cannot solve (x_0, y_0) since they require the knowledge of "propagation delays" r, $\widetilde{\tau}_i$, or τ_i.

In the rest of this section, we will develop a novel algorithm for estimating the "target position" (x_0, y_0) and the "propagation delays" r of all sensor nodes.

Let

$$d_{i,j} = \left\|\mathbf{x}_i - \mathbf{x}_j\right\|, \quad i, j = 1, \ldots, I. \tag{5}$$

And then define an $I \times I$ similarity matrix $\mathbf{B}$ [11]:

$$\mathbf{B} = \begin{bmatrix} \mathbf{s} - \mathbf{x}_1 \\ \vdots \\ \mathbf{s} - \mathbf{x}_I \end{bmatrix} \begin{bmatrix} \mathbf{s} - \mathbf{x}_1 \\ \vdots \\ \mathbf{s} - \mathbf{x}_I \end{bmatrix}^T, \tag{6}$$

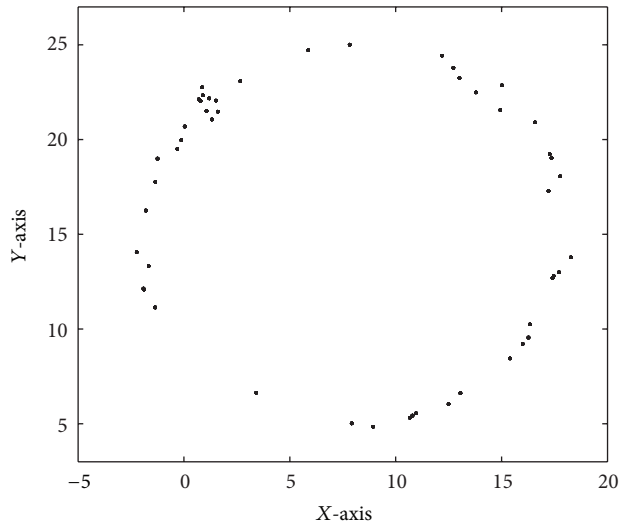


Figure 3: Data points used in the first experiment.

the (i, j)th element of which can be represented as

$$\begin{aligned}\mathbf{B}(i,j) &= 0.5\left(\left\|\mathbf{s}-\mathbf{x}_i\right\|^2+\left\|\mathbf{s}-\mathbf{x}_j\right\|^2-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2\right)\\ &= r^2-0.5d_{i,j}^2+r\left(n_i+n_j\right)+0.5\left(n_i^2+n_j^2\right).\end{aligned} \tag{7}$$

Under the ideal (without noise) case, $\mathbf{B}(i,j)=r^2-0.5d_{i,j}^2$. Note that r is unknown, and thus $\mathbf{B}$ is actually the function of r, that is, $\mathbf{B}(r)$.

Since the rank of $\begin{bmatrix}\mathbf{s}-\mathbf{x}_1\\ \vdots\\ \mathbf{s}-\mathbf{x}_I\end{bmatrix}$ equals 2, the rank of $\mathbf{B}(r)$ is also 2. From $\mathbf{B}(r)$, we introduce the following partition:

$$\mathbf{B}(r)=\left[\mathbf{B}_1(r)\quad \mathbf{B}_2(r)\right], \tag{8}$$

where $\mathbf{B}_1(r)$ and $\mathbf{B}_2(r)$ are the first two and last $(I-2)$ columns of $\mathbf{B}(r)$, respectively.

Similar to the conventional propagator method [13], we define the propagator

$$\mathbf{P}(r)=\left(\mathbf{B}_1(r)^T\mathbf{B}_1(r)\right)^{-1}\mathbf{B}_1(r)^T\mathbf{B}_2(r), \tag{9}$$

which satisfies

$$\mathbf{P}(r)=\min_{\mathbf{P}}\left\|\mathbf{B}_2(r)-\mathbf{B}_1(r)\mathbf{P}\right\|. \tag{10}$$

Let $\mathbf{x}(:,1)$ and $\mathbf{x}(:,2)$ stand for the first and second column of $[\mathbf{x}_1^T\ \mathbf{x}_2^T\ \cdots\ \mathbf{x}_I^T]^T$. Based on the propagator method [13], we have

$$\begin{aligned}\left[x_0\times 1-\mathbf{x}(:,1)\right]^T\begin{bmatrix}\mathbf{P}(r)\\ -\mathbf{I}\end{bmatrix}&=0,\\ \left[y_0\times 1-\mathbf{x}(:,2)\right]^T\begin{bmatrix}\mathbf{P}(r)\\ -\mathbf{I}\end{bmatrix}&=0,\end{aligned} \tag{11}$$

where 1 denotes the $I\times 1$ vector with all elements 1.

Table 1: Estimated results using different algorithms (Experiment 1).

Parameters	Hough	LS	Proposed
r (10)	10	9.9177	9.9204
x_0 (8)	8	8.0354	8.0332
y_0 (15)	15	14.9800	14.9703

Table 2: Estimated results using different algorithms (Experiment 2).

Parameters	Reference	Hough	LS	Proposed
$2\times r$ (mm)	45.07	45.0700	44.9838	44.9874
x_0 (pixel)	—	161	160.5635	160.5635
y_0 (pixel)	—	123	122.8443	122.8443

Table 3: Estimated results using different algorithms (Experiment 3).

Parameters	Hough	LS	Proposed
r (pixel)	23	23.2277	23.2237
x_0 (pixel)	120	119.7545	119.7542
y_0 (pixel)	156	156.4885	156.4885

From (11), we can solve x_0 and y_0 as follows:

$$\begin{aligned}\widehat{x}_0(r)&=\frac{\left(1^T\begin{bmatrix}\mathbf{P}(r)\\ -\mathbf{I}\end{bmatrix}\right)\left(\mathbf{x}(:,1)^T\begin{bmatrix}\mathbf{P}(r)\\ -\mathbf{I}\end{bmatrix}\right)^T}{\left(1^T\begin{bmatrix}\mathbf{P}(r)\\ -\mathbf{I}\end{bmatrix}\right)\left(1^T\begin{bmatrix}\mathbf{P}(r)\\ -\mathbf{I}\end{bmatrix}\right)^T},\\ \widehat{y}_0(r)&=\frac{\left(1^T\begin{bmatrix}\mathbf{P}(r)\\ -\mathbf{I}\end{bmatrix}\right)\left(\mathbf{x}(:,2)^T\begin{bmatrix}\mathbf{P}(r)\\ -\mathbf{I}\end{bmatrix}\right)^T}{\left(1^T\begin{bmatrix}\mathbf{P}(r)\\ -\mathbf{I}\end{bmatrix}\right)\left(1^T\begin{bmatrix}\mathbf{P}(r)\\ -\mathbf{I}\end{bmatrix}\right)^T}.\end{aligned} \tag{12}$$

Note that $\mathbf{P}(r)$ depends on r and thus $x_0(r)$ and $y_0(r)$ are not determined from (12) directly and can only be represented as the functions of the unknown "propagation delay" r.

Plugging (12) (i.e., $x_0(r)$ and $y_0(r)$) into (4), we can obtain

$$\left\|\mathbf{s}(r)-\mathbf{x}_i\right\|=r+n_i,\quad i=1,\ldots,I, \tag{13}$$

which implies that r is the root of the equations previously mentioned in absence of noise.

According to the fixed-point iteration theory [12], r is the fixed point of the function $\|\mathbf{s}(r)-\mathbf{x}_i\|$, that is, the value that the function $\|\mathbf{s}(r)-\mathbf{x}_i\|$ "locks onto" in the iterative process. Thus, we can compute "propagation delay" r using the following iteration procedure:

$$r(k+1)=\left\|\mathbf{s}(r(k))-\mathbf{x}_i\right\|,\quad k\geq 0,\ i=1,\ldots,I, \tag{14}$$

which are combined to yield the following iteration process:

$$r(k+1)=\frac{1}{I}\sum_{i=1}^{I}\left\|\mathbf{s}(r(k))-\mathbf{x}_i\right\|,\quad k\geq 0. \tag{15}$$

Once circle radius r is obtained from the previously mentioned iterative procedure, the circle center (x_0, y_0) can be solved from (12).

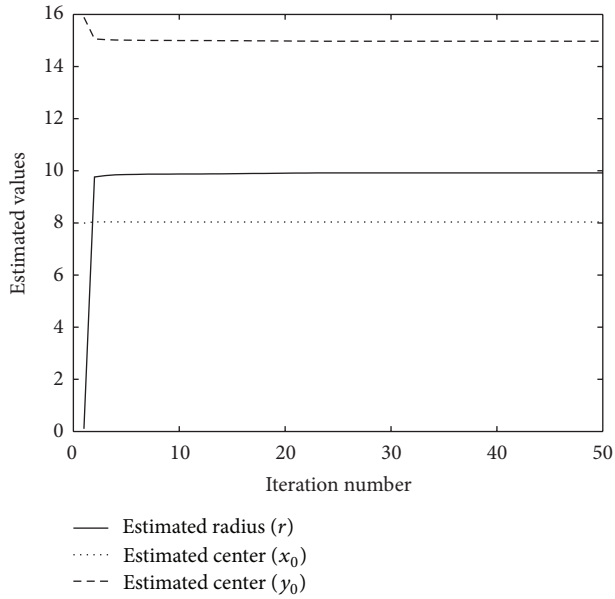


Figure 4: Estimated circle parameters versus iteration number (Experiment 1).

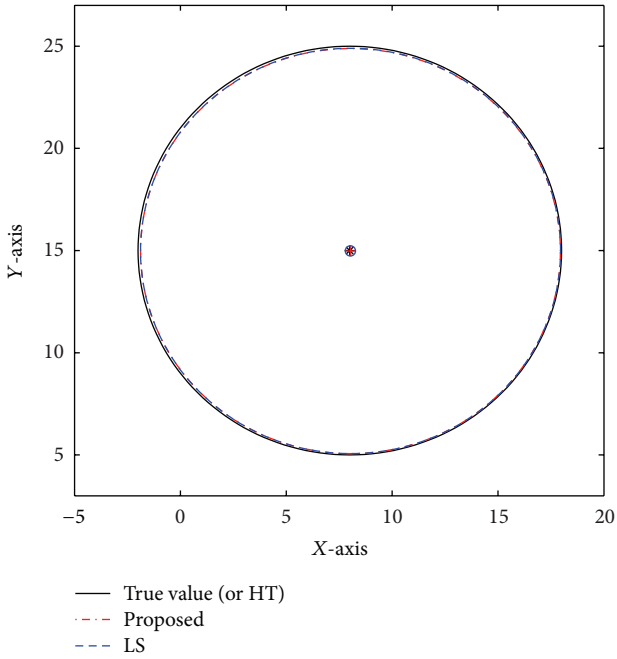


Figure 5: Fitting results using different algorithms.

4. Simulation Results

In this section, some experiments are conducted to evaluate the performance of the proposed method. For comparison, we simultaneously implement the HT method [4, 5] and the LS approach [6–8].

4.1. Experiment 1. The first experiment is implemented on $I = 50$ data points of a circle shown in Figure 3, where the

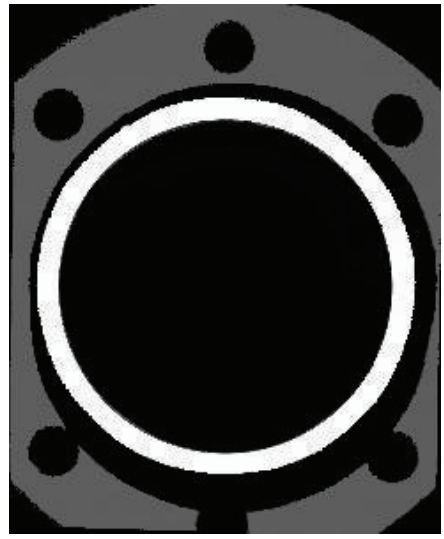


Figure 6: Experimental data used in Experiment 2.

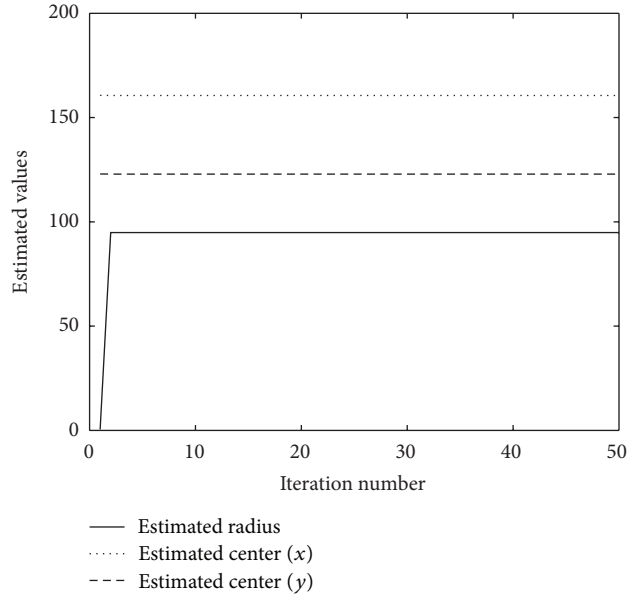


Figure 7: Estimated circle parameters versus iteration number (Experiment 2).

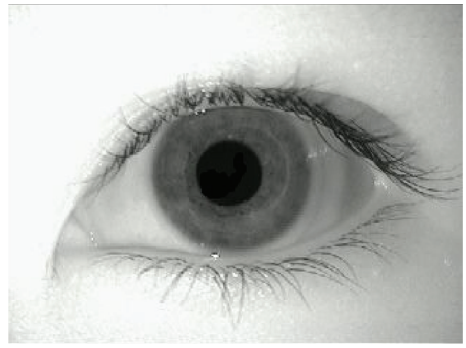


Figure 8: Iris Image used in Experiment 3.

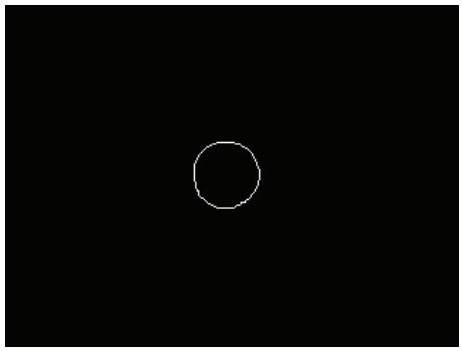


Figure 9: Edge points used in Experiment 3.

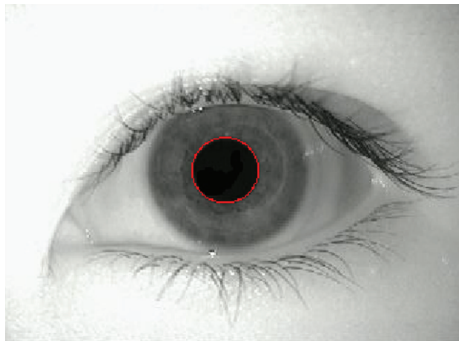


Figure 10: Fitting result using the proposed algorithm.

noise is stochastic additive white Gaussian noise with zero mean and variance 0.01, and the true circle parameters are $r = 10$, $x_0 = 8$, and $y_0 = 15$. The initial value of $r(0)$ is 0.1. Figure 4 shows the realized r, x_0, and y_0 using the proposed algorithm with 50 iterations. We can see from Figure 4 that the realization of r, x_0, and y_0 generally stays around 10, 8, and 15, respectively, after 3 iterations in this problem, which shows that the proposed algorithm converges rapidly. Table 1 gives the estimated results using the proposed algorithm, the LS method, and the Hough transform approach. We can see from Table 1 that the proposed algorithm has the approximate estimation accuracy as those of HT and LS and is slightly more accurate in estimating (r, x_0) than the LS method. Figure 5 shows the fitting results using different algorithms, which further shows that the fitting result obtained by the proposed algorithm approaches the true circle.

4.2. Experiment 2. In this experiment, the proposed algorithm is applied to the real data. Figure 6 (resolution 300×244) shows an example which computes the diameter of the bright hole after orientating the clouds of points to get the hole plane parallel to the projection plane (Available from http://www.aqsense.com/docs/docu/Compatibility.html). We can obtain the inner edge points of the hole via threshold segmentation, edge detection, and spectral clustering and then fit these points. Figure 7 shows the realized r, x_0, and y_0 using the proposed algorithm with 50 iterations. Table 2 lists the estimation results using the proposed algorithm, the HT method, and the LS approach.

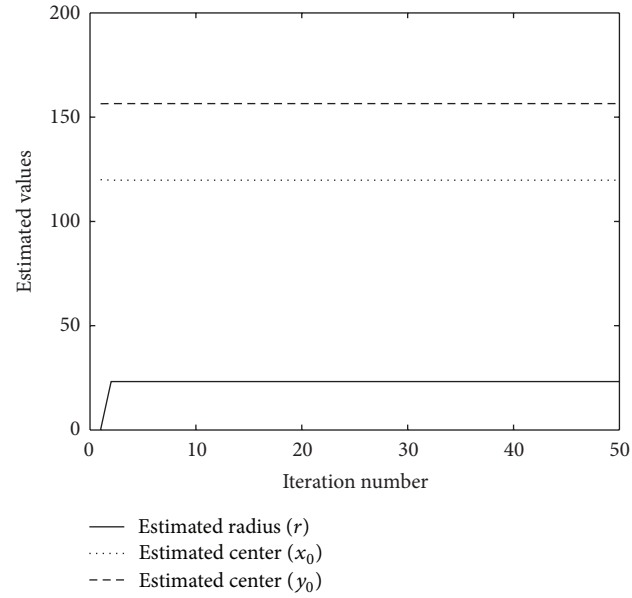


Figure 11: Estimated circle parameters versus iteration number (Experiment 3).

4.3. Experiment 3. Iris recognition is a biometric identification technique based on images of the irides of an individual's eyes. Since the Iris area lies between the pupil region (a dark ellipse with the lowest intensity) and limbus region, determining the pupil region is an important preprocessing step of Iris localization. In the third experiment, we implement the proposed algorithm on the Iris image, as shown in Figure 8. Via the thresholding segmentation and Sobel edge detection, edge points are given in Figure 9. Table 3 gives the estimation results from three different methods. Figure 10 displays the fitting results by the proposed algorithm that are marked by red points, and Figure 11 shows the realized r, x_0, and y_0 using the proposed algorithm with 50 iterations, which shows that the proposed algorithm can fit the Pupil's boundary well.

Although the HT method is of the highest estimation accuracy, it needs to be pointed out that the HT method requires (i) quantizing the three-dimensional space finely enough; otherwise the peaks in the transform plane will be broadened and (ii) the overwhelming burden of the three-dimensional search in the r, x_0, and y_0 plane.

5. Conclusion

In this paper, we propose a novel circle fitting algorithm by borrowing the idea from source localization in wireless sensor networks. Since the virtual propagation delays of all sensor nodes are unknown, the existing source localization algorithms cannot be applied. This paper formulates the virtual source localization model of three unknown parameters (r, x_0, y_0) into special nonlinear equations of a unique parameter, that is, the circle radius r, using the MDS analysis and propagator method, and then it employs the classical fixed-point iteration theory to determine the circle radius and circle center.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61172123 and 60901059, by Excellent Youth Research Star (2012KJXX-35), and educational Department Foundation of Shaanxi Province (12JK0526).

References

[1] U. M. Landau, "Estimation of a circular arc center and its radius," *Computer Vision, Graphics and Image Processing*, vol. 38, no. 3, pp. 317–326, 1987.

[2] J. F. Crawford, "A non-iterative method for fitting circular arcs to measured points," *Nuclear Instruments and Methods in Physics Research*, vol. 211, no. 1, pp. 223–225, 1983.

[3] G. Coath and P. Musumeci, "Adaptive arc fitting for ball detection in robocup," in *Proceedings of APRS Workshop on Digital Image Computing*, pp. 63–68, Brisbane, Australia, February 2003.

[4] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[5] D. J. Kerbyson and T. J. Atherton, "Circle detection using hough transform filters," in *Proceedings of the 5th International Conference on Image Processing and its Applications*, pp. 370–374, July 1995.

[6] H. Spath, "Least-square fitting by cirles," *Computing*, vol. 57, no. 2, pp. 179–185, 1996.

[7] I. D. Coope, "Circle fitting by linear and nonlinear least squares," *Journal of Optimization Theory and Applications*, vol. 76, no. 2, pp. 381–388, 1993.

[8] L. Moura and R. Kitney, "A direct method for least-squares circle fitting," *Computer Physics Communications*, vol. 64, no. 1, pp. 57–63, 1991.

[9] H. C. So and F. K. W. Chan, "A generalized subspace approach for mobile positioning with time-of-arrival measurements," *IEEE Transactions on Signal Processing*, vol. 55, no. 10, pp. 5103–5107, 2007.

[10] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1770–1778, 2008.

[11] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2001.

[12] R. L. Burden and J. D. Faires, *Numerical Analysis: Fixed-Point Iteration*, PWS Publishers, 2010.

[13] S. Marcos, A. Marsal, and M. Benidir, "The propagator method for source bearing estimation," *Signal Processing*, vol. 42, no. 2, pp. 121–138, 1995.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 208904, 9 pages
http://dx.doi.org/10.1155/2013/208904

Research Article

Indoor Mobile Localization in Wireless Sensor Network under Unknown NLOS Errors

Long Cheng,[1] Hao Wu,[2] Chengdong Wu,[1] and Yunzhou Zhang[1]

[1] *College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*
[2] *Faculty of Engineering & Information Technologies, University of Sydney, NSW 2006, Australia*

Correspondence should be addressed to Long Cheng; chenglong2000_0@yahoo.com.cn

Received 17 November 2012; Accepted 12 December 2012

Academic Editor: Shuai Li

Localization is one of the key techniques in wireless sensor network. One of the main problems in indoor mobile localization is non-line-of-sight (NLOS) propagation. And the NLOS effects will lead to a large localization error. So the NLOS problem is the biggest challenge for accurate mobile location estimation in WSN. In this paper, we propose a likelihood matrix correction based mixed Kalman and H-infinity filter (LC-MKHF) method. A likelihood matrix based correction method is firstly proposed to correct the LOS and NLOS measurements. This method does not need the prior information about the statistical properties of the NLOS errors. So it is independent of the physical measurement ways. And then a mixed Kalman and H-infinity filter method is proposed to improve the range measurement. Simulation results show that the LC-MKHF algorithm has higher estimate accuracy in comparison with no-filter, Kalman filter, and H-infinity filter methods. And it is robust to the NLOS errors.

1. Introduction

Localization is one of the most important subjects. Global position system (GPS) is the most common position system in the world. However, research studies show that the GPS performance degrades drastically when the receiver is located in large buildings. In recent years, there has been a growing research interest in wireless sensor network (WSN). Wireless sensor network that consist of thousands of low cost sensor nodes have been used in many promising applications such as health surveillance, robot navigation [1], and environmental monitoring. And localization is one of the important applications of WSN.

The classic methods to estimate the indoor location are time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA), and received signal strength (RSS). TOA method measures travel times of signals between nodes. TDOA method is located by measuring the signals' arrival time difference between anchor nodes and mobile node. It is able to achieve high ranging accuracy but requires extra hardware and consumes more energy. The angles between mobile node and a number of anchor nodes are used in the AOA method to estimate the location. RSS has established the mathematical model on the basis of path loss attenuation with distance, and it requires relatively low configuration and energy. And the data fusion technique has been utilized to improve the accuracy by combining two or more measurements such as TOA/AOA, TDOA/AOA, and TOA/RSS [2]. Accurate position estimation can be obtained by using filtering techniques if there exists direct propagation between the beacon node and the mobile node, also known as the line of sight (LOS). The complicated indoor environment causes non-line-of-sight (NLOS) situation. And the NLOS effects will lead to a large localization error. So the NLOS problem is the biggest challenge for accurate mobile location estimation in WSN.

In indoor environments, the NLOS signal propagation such as reflection and diffraction leads to an overestimation, which in turn results in erroneous measurements containing NLOS errors. There are two ways to deal with the NLOS situation: unknown parameters of NLOS errors and known parameters of NLOS errors.

For the unknown parameters of NLOS errors method, the advantage of this method is that it can be used with any of the ranging technologies and does not require prior information about statistical properties of the NLOS measurements. In

[3], a residual weighting algorithm (Rwgh) which uses the sum of squared residuals of a least squares estimation as the indicator of the accuracy of calculated node coordinates was proposed. This method applies least squares multilateration on all possible combinations of the distance measurements and then the estimated location is computed as a weighted combination of these intermediate estimates. A Gaussian mixture model (GMM) [4] is used to model the distribution of LOS errors and NLOS errors [5].

For the known parameters of NLOS errors method, the first method attempts to identify the NLOS propagation such as hypothesis test [6], likelihood ratio test [7], and statistical analysis methods [8, 9] and then it mitigates the NLOS errors. If the identification is correct, the localization accuracy can be achieved. However, the probability of wrong identification is inevitable. The second method attempts to combine all of the LOS and NLOS measurements to estimate the location of mobile node. The interacting multiple model (IMM) approach with the Kalman filtering technique is developed [10]. And the data fusion based IMM approach is investigated [11, 12]. An M-estimator [13] is employed to estimate the distribution of NLOS errors. And an IMM based cubature Kalman filter [14] is introduced to deal with the maneuvers of the target. Most of the above mentioned algorithms rely on certain NLOS error distribution which is unknown in practice. However, most of the methods focus on cellular network or ultrawideband communication system. Very few papers investigate the mobile node mobile localization in NLOS environment for WSN.

In this paper, we firstly propose a likelihood matrix based correction method. And then a mixed Kalman filter and H-infinity filter method is proposed. A 2-dimensional analysis is provided, as extension to 3 dimensions is rather straightforward. The main contribution of this paper is given as follows.

(1) The proposed likelihood matrix based correction method does not require identification of propagation state, and it is independent of the physical measurement ways.

(2) The proposed method does not require any prior information about the statistical properties of NLOS errors. Therefore, it can be widely used in the cellular network, ad hoc network, and wireless sensor network.

(3) In comparison with other methods, the proposed method is robust to the NLOS errors.

The rest of the paper is organized as follows. Section 2 provides the system model in LOS/NLOS conditions. We will introduce our proposed strategy in Section 3. Some simulation results will present in Section 4. The conclusions are given in Section 5.

2. System Model

In this section, we introduce some technical preliminaries. The assumption of this paper is as follows. Consider a range-based 2D localization problem employing N beacon nodes at known locations $Z_i = [x_i, y_i]^T$, $i = 1, \ldots, N$. The position of the mobile node at time k is $Z_u^t = [x_k^u, y_k^u]^T$, $k = 1, \ldots, t_n$. The beacon nodes send the signal and the mobile node receives it and converts it to range information. The real distance between ith the beacon node and the mobile node at time k is

$$d_k^i = \sqrt{\left(x_k^u - x_i\right)^2 + \left(y_k^u - y_i\right)^2}. \quad (1)$$

In LOS propagation conditions, the range measurement by ith beacon node at time k is modeled as follows:

$$\widehat{d}_k^i = d_k^i + n_i, \quad (2)$$

where n_i is the measurement noise modeled as zero mean white Gaussian with variance σ_i^2.

In NLOS propagation conditions, the signal does not travel in a straight line when an obstacle exists between the beacon and mobile node due to the reflection or diffusion effect. So the range measurement by ith beacon node at time k is modeled as follows:

$$\widehat{d}_k^i = d_k^i + n_i + b_{\mathrm{NLOS}}, \quad (3)$$

where n_i is the measurement noise with zero mean and σ_i^2 variance. b_{NLOS} is the NLOS errors and is assumed to be independent of the measurement noise n_i. And the NLOS error b_{NLOS} obeys Gaussian, uniform, exponential, or Delta distribution. The distribution and parameters of b_{NLOS} are different in different indoor environments and measurement methods.

3. Implementation of the Proposed Algorithm

The system flow of the proposed NLOS errors mitigation method is plotted in Figure 1. The input of this method is the measured distance $\widehat{d}_k^i$, and the output of the method is filtered distance $\widetilde{d}_k^i$. This method consists of two major stages, that is, likelihood matrix based correction and mixed Kalman and H-infinity filter. The proposed algorithm is described in the following discussion.

3.1. Likelihood Matrix Based Correction. Firstly, a grid-based likelihood matrix is established to provide the initial localization results. And then the corrected range and the factor of the mixed Kalman/H-infinity filter are introduced.

At time k, the proposed algorithm consists of the following steps.

Step 1. The area is divided into a grid Φ with $W \times W$ cells and grid resolution w, for example, a 100×100 field with $W = 5$ and a grid resolution w is equal to 20. The number of cells is a tradeoff between estimation accuracy and computational complexity. $C(i, j)$, for $i, j = 1, \ldots, W$, denotes the centers of these cells in a matrix form. And then we construct a $W \times W$ likelihood matrix V. The elements of the likelihood matrix are obtained by

$$V(m, n) = \sum_{i=1}^{N} b_i(m, n), \quad \text{for } m, n = 1, \ldots, W, \quad (4)$$

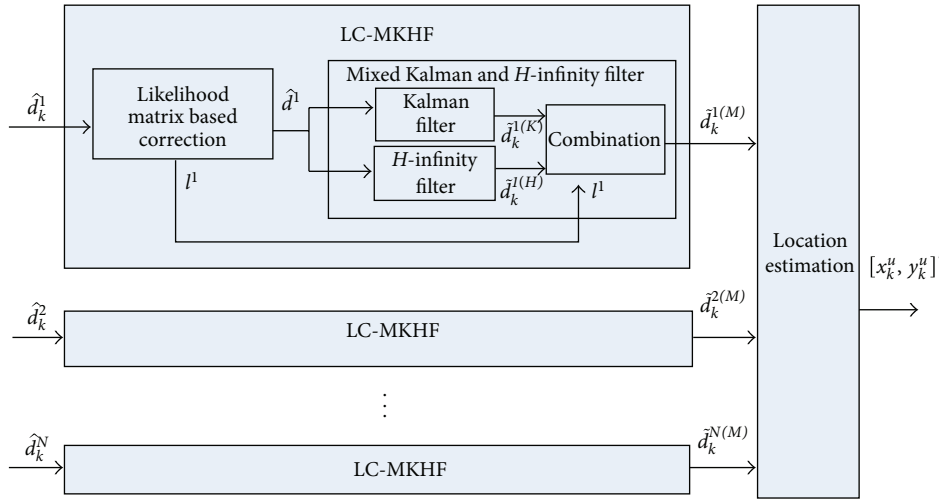


Figure 1: Architecture of the localization system.

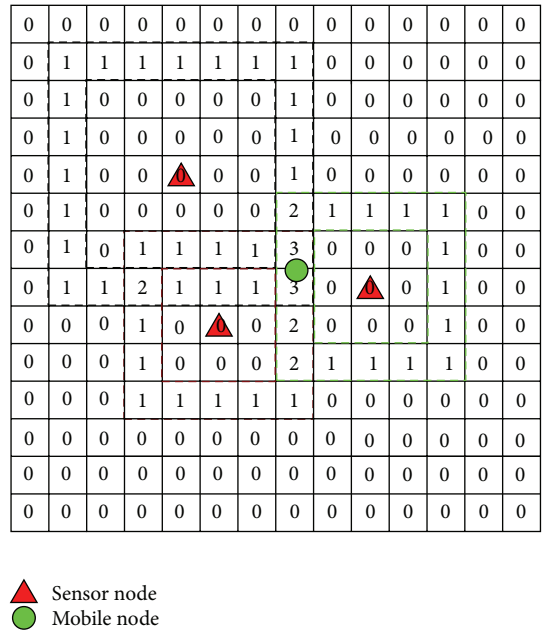


Figure 2: An example of Step 1.

where

$$b_i(m,n) = \begin{cases} 1, & \widehat{d}^i - \varepsilon \le d_{imn} \le \widehat{d}^i + \varepsilon \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

d_{imn} is the Euclidean distance between the $C(m,n)$ and ith node. $\widehat{d}^i$ is the measured distance of ith node. And we set $\varepsilon = \sigma_i$.

Let (m^*, n^*) be the element of V with the maximum value, that is, $V(m^*, n^*) \ge V(m,n)$, for $m, n = 1, \ldots, W$, then the estimated location of mobile node is $C^* = [C_1^*, \ldots, C_v^*]$. C^* is the initial results set of localization. The center of the initial results set can be expressed as $\overline{C}^* = \sum_i^v C_i^*/v$.

In order to illustrate Step 1, we provide a simple example using a square measured distance. The measured distances of sensors are 2, 2, and 3 respectively. The locations of the sensors are illustrated in Figure 2. For the construction of the likelihood matrix V, each sensor node adds a positive one contribution to the element of V that corresponds to the cells that the measured range meets the condition $\widehat{d}^i - \varepsilon \le d_{imn} \le \widehat{d}^i + \varepsilon$. On the other hand, sensor node adds zero contribution to these elements. The resulting likelihood matrix after computing these contributions of three sensor nodes is shown in Figure 2. The mobile node is correctly localized in the grid cell with the maximum value 3.

Step 2. The corrected range by ith beacon node at time k is can be expressed as

$$\widehat{d}_k^i = \left\| \overline{C}^* - Z_i \right\|. \tag{6}$$

Step 3. The factor of the mixed Kalman/H-infinity filter can be obtained as

$$l_k^i = \frac{\left\| \widehat{d}_k^i - \widehat{d}_k^i \right\|}{\sum_{i=1}^{N} \left\| \widehat{d}_k^i - \widehat{d}_k^i \right\|}. \tag{7}$$

3.2. Kalman Filter. Kalman filter is the most widely used method for tracking and mobile location estimation. And the Kalman filter is suggested as a promising method to range measurement for smoothing and mitigating Gaussian noise. A modified Kalman filter [15, 16] is employed to reduce the NLOS error in rough wireless environments. A modified extended Kalman filter [17] is proposed to jointly estimate the mobile state and the LOS/NLOS sight state based on the measurements. And then a Bayesian data fusion algorithm is applied to achieve high estimation accuracy. An extended Kalman based interacting multiple model smoother (EKF-IMM) [11] is proposed. The IMM is employed as a switch between the LOS and NLOS condition, which are considered to be a Markov process with two interactive models. And then the Kalman filter is used for nonlinear estimation. In this paper, we employ the Kalman filter to mitigate the measurement noise.

The state vector at time k is defined as $X_k = [\widehat{d}_k^i, \dot{\widehat{d}}_k^i]^T$, where $[\widehat{d}_k^i, \dot{\widehat{d}}_k^i]^T$ represents the measured distance and the corresponding velocity. Then the motion model with random acceleration can be modeled as

$$X_{k+1}^i = \Phi_k X_k^i + w_k, \tag{8}$$

where w_k is the random process noise modeled as zero mean with variance Q_k^i. The state transition matrix Φ_k is defined as $\Phi_k = \left[\begin{smallmatrix} 1 & \Delta t \\ 0 & 1 \end{smallmatrix}\right]$, and $\Delta t = t_{k+1} - t_k$ is the sampling period.

The measurement model is defined as (2) and (3) in LOS and NLOS environments, respectively. So the measurement model can be rewritten as

$$z_k^i = H_k X_k^i + v_k, \tag{9}$$

where $H_k = [1, 0]$. Most of the papers assume that b_{NLOS} is Gaussian distribution, that is, $b_{\mathrm{NLOS}} \sim N(\mu_{\mathrm{NLOS}}, \sigma_{\mathrm{NLOS}}^2)$. So, $v_k \sim N(m_k, R_k)$ can be rewritten as

$$\begin{aligned} v_k &\sim N\left(0, \sigma_i^2\right), \quad \text{LOS} \\ v_k &\sim N\left(\mu_{\mathrm{nlos}}, \sigma_i^2 + \sigma_{\mathrm{NLOS}}^2\right), \quad \text{NLOS}. \end{aligned} \tag{10}$$

Therefore, the Kalman filter cannot perform better performance when the NLOS noise is non-Gaussian distribution.

The operation of the Kalman filter can be represented by two recursive steps. The prediction step includes the following operations:

$$\begin{aligned} \widehat{X}_{k+1|k}^i &= \Phi_{k+1} \widehat{X}_{k|k}^i, \\ P_{k+1|k}^i &= \Phi_{k+1} P_{k|k}^i \Phi_{k+1}^T + Q_{k+1}. \end{aligned} \tag{11}$$

The update step includes the following operations:

$$\begin{aligned} \widetilde{y}_{k+1}^i &= z_{k+1}^i - H_{k+1} X_{k+1}^i, \\ S_{k+1} &= H_{k+1} P_{k+1|k}^i H_{k+1}^T + R_{k+1}, \\ K_{k+1} &= P_{k+1|k}^i H_{k+1}^T S_{k+1}^{-1}, \\ \widehat{X}_{k+1|k+1}^i &= \widehat{X}_{k+1|k}^i + K_{k+1} \widetilde{y}_{k+1}^i, \\ P_{k+1|k+1}^i &= P_{k+1|k}^i - K_{k+1} S_{k+1} K_{k+1}^T. \end{aligned} \tag{12}$$

The output of Kalman filter can be expressed as

$$\widetilde{d}_k^{i(K)} = D \widehat{X}_{k+1|k+1}^i, \quad D = [1, 0]. \tag{13}$$

3.3. H-Infinity Filter. Although some filtering techniques such as unscented filter [18] and particle filter [19] are superior to the Kalman filter for nonlinear and non-Gaussian noise system, the computational cost is too high that they do not suit for resource limited sensor nodes. The Kalman filter is established on the H-2 estimate criterion, and it needs to know the statistical properties of the noise. However, it is not practical especially in the complex indoor environment. H-infinity filter differs from the Kalman filter in that it does not require the knowledge of the noise properties except that the noises are assumed to have bounded power. Therefore, H-infinity filter is one of the promising methods due to its simplicity, optimality, tractability, and robustness [20, 21]. However, very few papers employ the H-infinity filter into the mobile location estimation field for WSN.

The motion model and measurement model are given by

$$\begin{aligned} X_{k+1}^i &= \Phi_k X_k^i + w_k, \\ z_k^i &= H_k X_k^i + v_k, \end{aligned} \tag{14}$$

where w_k and v_k are process noise and measurement noise, respectively. And the distribution and the statistical properties of the noises are unknown. The definition of Φ_k and H_k is the same with Kalman filter.

The linear combination of measurement and state vectors [22, 23] is

$$s_k^i = L_k X_k^i. \tag{15}$$

We assume that the initial covariance of estimation error is $P_0^i = E[(X_0^i - \widehat{X}_0^i)(X_0^i - \widehat{X}_0^i)^T]$. $\widehat{s}_k^i = F_f(z_0^i, z_1^i, \ldots, z_k^i)$ represents the estimation of s_k^i using the measurements z_k^i from 0 to time k. The filtering error is $\Delta s_k^i = \widehat{s}_k^i - s_k^i$. $T_k(F_f)$ denotes the transfer operators that map the unknown

Table 1: The default parameter values.

Parameters	Symbol	Default values
Number of beacon nodes	N	7
The standard deviation of measurement noise	σ_i	4
The NLOS errors	$b_{\text{NLOS}} \sim N(\mu_{\text{NLOS}}, \sigma^2_{\text{NLOS}})$	$\mu_{\text{NLOS}} = 4, \sigma_{\text{NLOS}} = 5$

disturbances $\{(P_0^i)^{-0.5}(X_0^i - \widehat{X}_0^i), w_k, v_k\}$ to the filtering error Δs_k^i. The H-infinity norm of the transfer operator T is defined as

$$\left\|T_k\left(F_f\right)\right\|_\infty = \frac{\sum_{i=0}^{k}\left(\Delta s_k^i\right)^T \Delta s_k^i}{\left(X_0^i - \widehat{X}_0^i\right)^T (P_0^i)^{-1}\left(X_0^i - \widehat{X}_0^i\right) + \sum_{i=0}^{k} w_k^T w_k + \sum_{i=0}^{k} v_k^T v_k}. \tag{16}$$

So the H-infinity filter can be described as follows. Given a scale $\gamma > 0$, find H-infinity estimation strategies that achieve $\|T_k(F_f)\|_\infty < \gamma$. In this paper, we set $L_k = I$. The recursive steps are as follows:

$$\begin{aligned} K_{k+1} &= \left(\frac{1}{\gamma} I + \Phi_{k+1}\left(P_{k+1|k}^i\right)^{-1} H_{k+1}^T\right) \\ &\quad \times \left(I + H_{k+1}\left(P_{k+1|k}^i\right)^{-1} H_{k+1}^T\right)^{-1}, \\ \widehat{X}_{k+1|k+1}^i &= \Phi_{k+1}\widehat{X}_{k|k}^i + K_{k+1}\left(z_{k+1}^i - H_{k+1}X_{k+1}^i\right), \\ P_{k+1|k}^i &= \left(P_{k+1|k}^i\right)^{-1} - \gamma^2 I, \\ P_{k+1|k+1}^i &= \left(\Phi_{k+1} - K_{k+1}H_{k+1}\right)\left(P_{k+1|k}^i\right)^{-1} \\ &\quad \times \left(\Phi_{k+1} - K_{k+1}H_{k+1}\right)^T + K_{k+1}K_{k+1}^T. \end{aligned} \tag{17}$$

The output of H-Infinity filter can be expressed as

$$\tilde{d}_k^{i(H)} = D\widehat{X}_{k+1|k+1}^i, \quad D = [1, 0]. \tag{18}$$

3.4. Mixed Kalman and H-Infinity Filter. In this subsection, we combine the results of Kalman and H-infinity filter. It can be expressed as

$$\tilde{d}_k^{i(M)} = l^i \cdot \tilde{d}_k^{i(H)} + \left(1 - l^i\right)\tilde{d}_k^{i(K)}. \tag{19}$$

3.5. Location Estimation. In this section, we briefly introduce maximum likelihood localization method. We set the position of beacon node as $\langle (x_1, y_1), \ldots, (x_N, y_N)\rangle$. At time k, the position of mobile node is $X = [x_k^u, y_k^u]^T$. $\tilde{d}_k^i$ is output by mixed Kalman and H-infinity filter algorithm:

$$\begin{aligned} \left(x_1 - x_k^u\right)^2 + \left(y_1 - y_k^u\right)^2 &= \left(\tilde{d}_k^1\right)^2 \\ &\vdots \\ \left(x_N - x_k^u\right)^2 + \left(y_N - y_k^u\right)^2 &= \left(\tilde{d}_k^N\right)^2. \end{aligned} \tag{20}$$

The linear equation $AX = b$ represents the above equation, where A and B are given by

$$A = 2\begin{bmatrix} (x_1 - x_2) & (y_1 - y_2) \\ (x_1 - x_3) & (y_1 - y_3) \\ \vdots & \vdots \\ (x_1 - x_N) & (y_1 - y_N) \end{bmatrix}, \tag{21}$$

$$B = \begin{bmatrix} \tilde{d}_k^2 - \tilde{d}_k^1 - \left(x_2^2 + y_2^2\right) + \left(x_1^2 + y_1^2\right) \\ \tilde{d}_k^3 - \tilde{d}_k^1 - \left(x_3^2 + y_3^2\right) + \left(x_1^2 + y_1^2\right) \\ \vdots \\ \tilde{d}_k^N - \tilde{d}_k^1 - \left(x_N^2 + y_N^2\right) + \left(x_1^2 + y_1^2\right) \end{bmatrix}. \tag{22}$$

We can obtain the coordinate matrix of the mobile node as follows:

$$X = \left(A^T A\right)^{-1} A^T B. \tag{23}$$

4. Performance Evaluation

In this section, we present simulation results for the proposed LC-MKHF algorithm for mobile location estimation in non-line-of-sight environments. As shown in Figure 3(a), we randomly deploy seven beacon nodes in the 100 m × 100 m square area and one mobile node (MN) moving in the area. We assume that the mobile node has the velocity of 1 m/s. The communication range of sensor node is 150 m. We compare the proposed method with no-filter (NF) method, Kalman filter (KF) method, and H-infinity Filter (HF) algorithm. The default parameter values in the simulation are shown in Table 1. In each simulation case, 1000 Monte Carlo runs are performed with the same parameters. The performance of the proposed algorithm is measured by average localization error (ALE):

$$\text{error} = \frac{1}{N \cdot t_n}\sum_{i=1}^{K}\sum_{k=1}^{t_n}\sqrt{\left(\widehat{x}_k - x_k^i\right)^2 + \left(\widehat{y}_k - y_k^i\right)^2}, \tag{24}$$

where $N = 1000$, $t_n = 100$, (x_k^i, y_k^i) is the true location of the mobile node, and $(\widehat{x}_i, \widehat{y}_i)$ is the estimated location.

Figure 3(a) shows the deployment of the beacon nodes and obstacles. In this paper, the beacon nodes and obstacles are randomly deployed. And Figure 3(b) shows sight state with respect to all the beacon nodes in sample points. We can see that the sight states vary with time. In Figure 3(c), we can obtain the detailed localization errors in each sample point. It can be observed that the proposed method has better performance in comparison with other methods in most of the sample points.

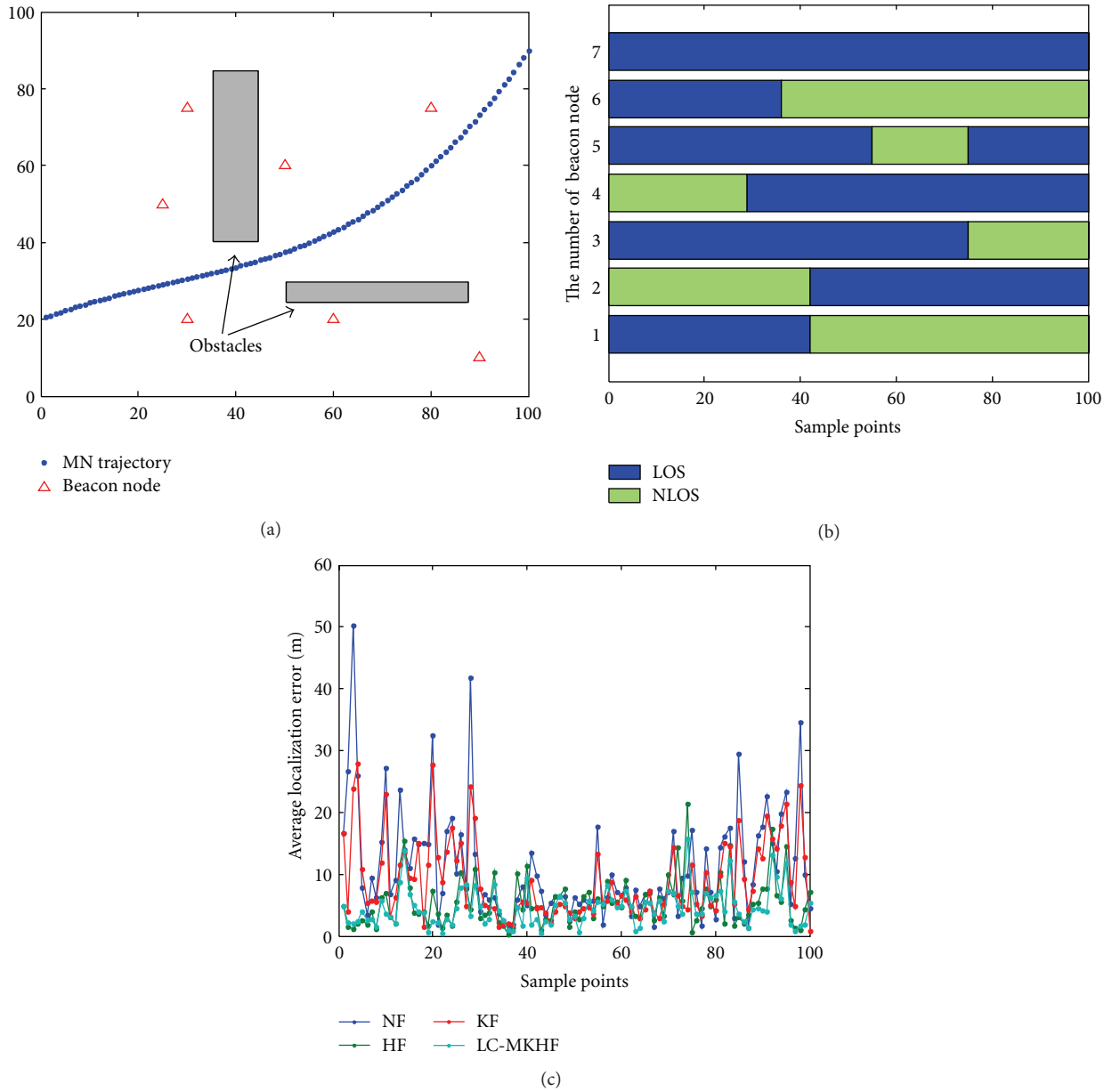


FIGURE 3: (a) The deployment of beacon nodes and obstacles. (b) The sight state of each beacon node in each sample point. (c) The average localization error in each sample point.

In the following section, we evaluate the performance of our proposed method under different environment, that is, the NLOS errors obey different distribution. We also investigate the effect of various parameters on the proposed method.

4.1. The NLOS Errors Obey Gaussian Distribution, That Is, $b_{NLOS} \sim N(\mu_{NLOS}, \sigma^2_{NLOS})$. In this subsection, we compare the proposed method with other methods when the NLOS errors obey Gaussian distribution. Figure 4(a) shows the relation between the average localization errors and the number of sensor nodes. The average localization errors of all methods decrease with the number of sensor nodes increase. And Figures 4(b) and 4(c) show the impact of mean and standard deviation of NLOS errors on the average localization errors. It can be observed that the mean and standard deviation of NLOS errors have a significant impact on KF and NF methods. The HF and LC-MKHF are robust to the parameters. In Figure 4(b), the proposed method has higher localization accuracy than HK, EK, and NF, about 39.59%, 48.94%, and 54.81%, respectively. In Figure 4(c), the proposed method has higher localization accuracy than HK, EK, and NF, about 28.32%, 47.91%, and 58.26%, respectively.

4.2. The NLOS Errors Obey Uniform Distribution, That Is, $b_{NLOS} \sim U(2, U_{\max})$. In this subsection, we assume that the

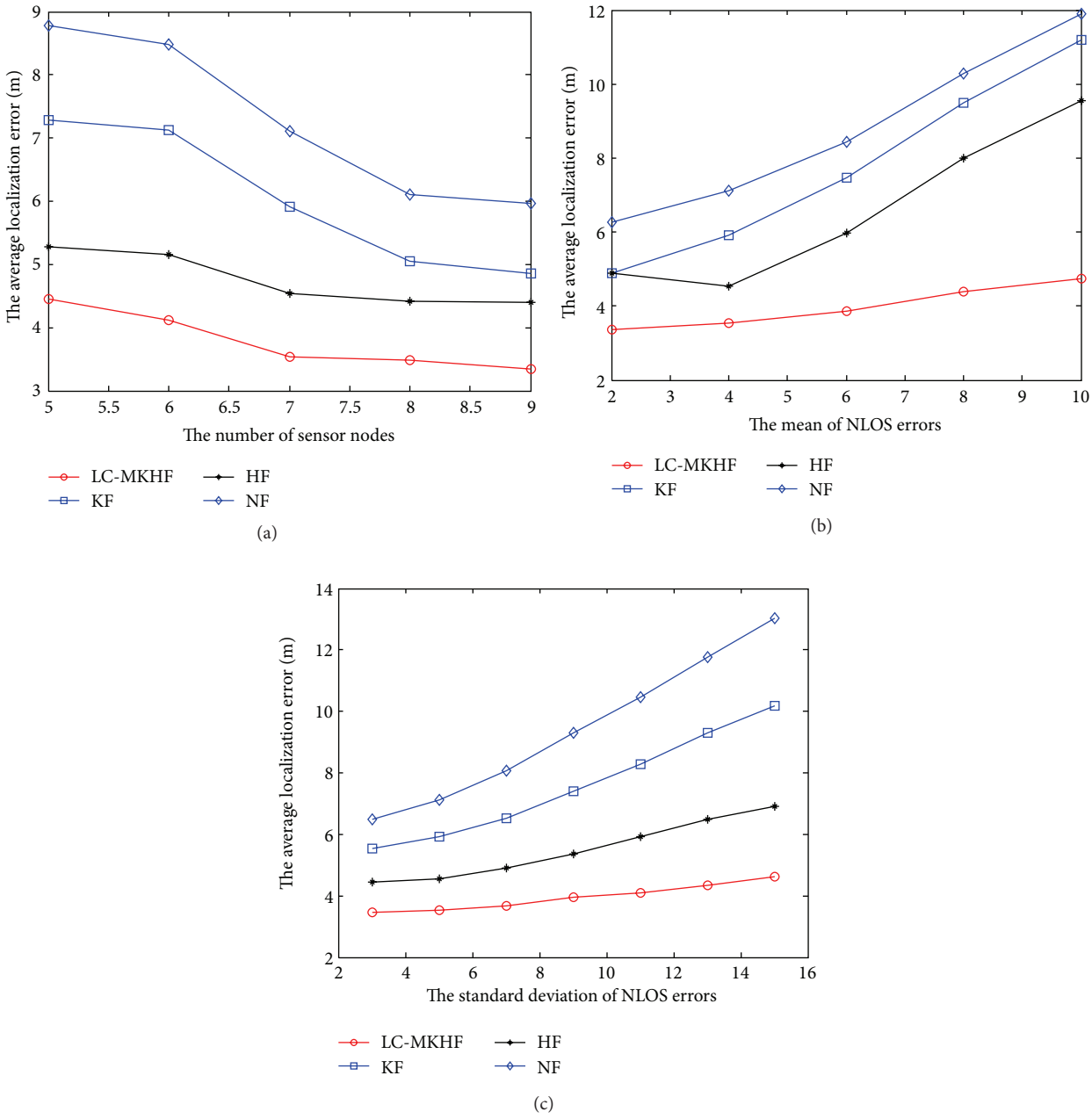


FIGURE 4: (a) The number of beacon nodes versus ALE. (b) The mean of NLOS errors versus ALE. (c) The standard deviation of NLOS errors versus ALE.

NLOS errors obey uniform distribution. The minimum and maximum values of parameters of uniform distribution are 2 and $U_{\max}$, namely, $U(2, U_{\max})$. We firstly investigate the impact of $U_{\max}$ on average localization errors. As shown in Figure 5(a), the KF has the similar performance with HF when the $U_{\max}$ is low. The NF method owns the worst performance, and our proposed method has the best localization accuracy. In Figure 5(b), we also investigate the influence of standard deviation of measurement noise when $U_{\max} = 7$. It can be observed that the proposed method has the highest localization accuracy in each case.

4.3. The NLOS Errors Obey Exponential Distribution, That Is, $b_{NLOS} \sim E(u)$. In this subsection, we study the performance of the methods when the NLOS errors obey exponential distribution. As shown in Figure 6(a), LC-MKHF performs stably when the mean parameter increases. However, the KF and NF methods increase with the mean parameter increase. The localization accuracy improved 58.75%, 52.56%, and 41.18% when comparing with NF, KF, and HF methods. And Figure 6(b) indicates that the proposed method owns the best performance under various standard deviation of measurement noise.

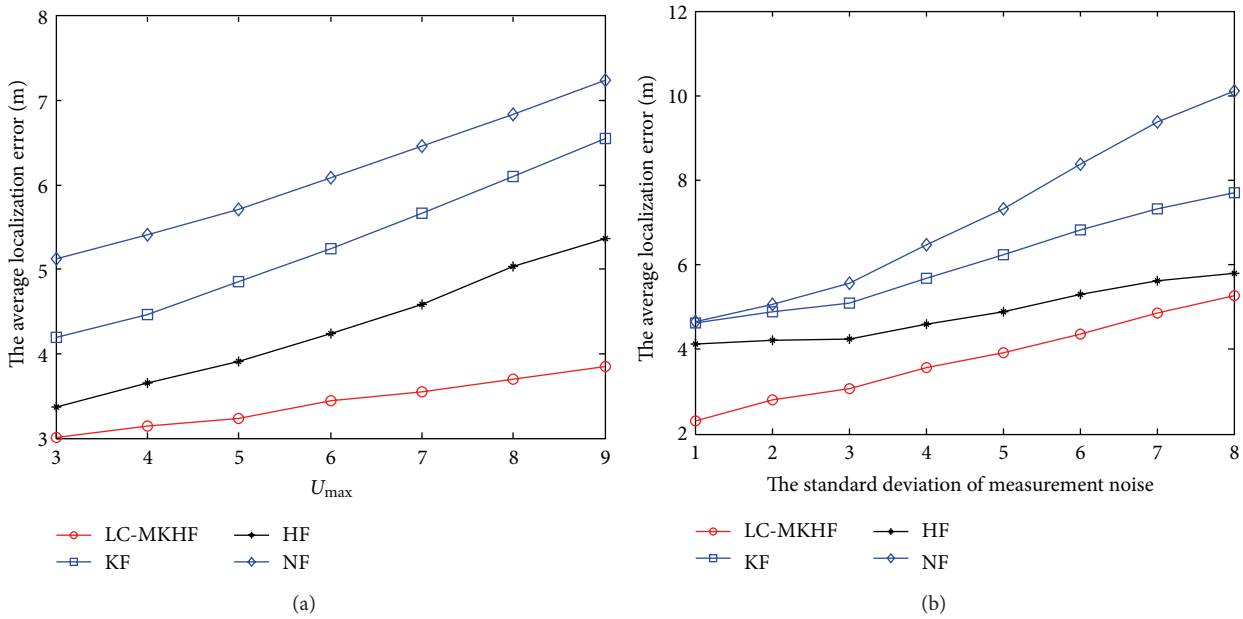


FIGURE 5: (a) The U_{max} versus ALE. (b) The standard deviation of measurement errors versus ALE.

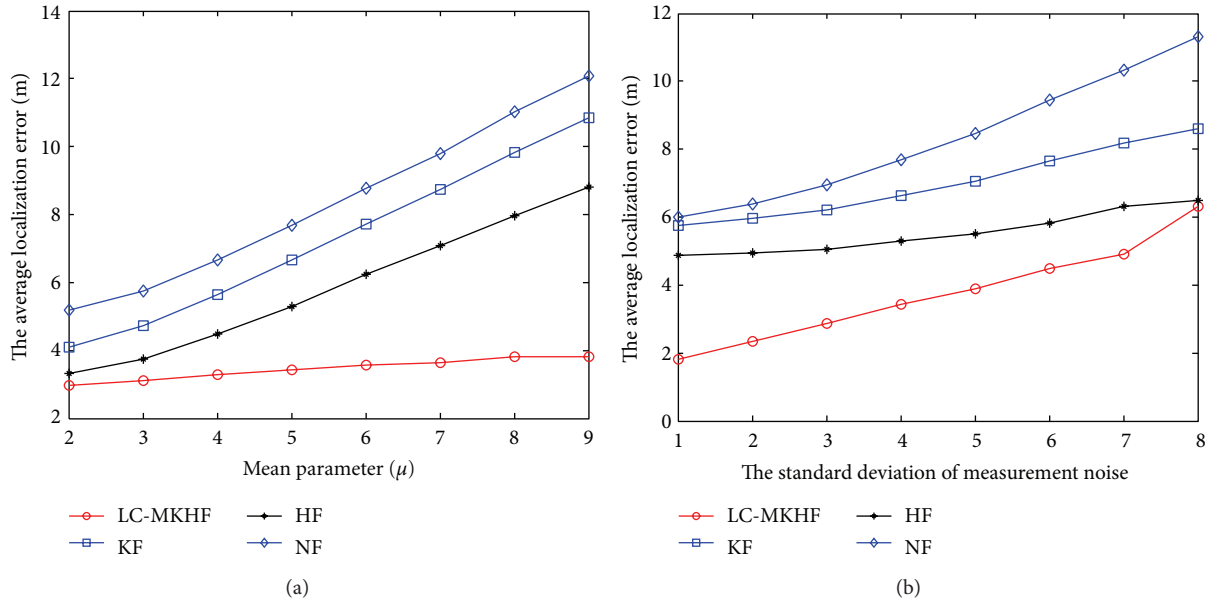


FIGURE 6: (a) The mean parameter versus ALE. (b) The standard deviation of measurement errors versus ALE.

5. Conclusion

This paper presents a novel LC-MKHF method which is robust to the NLOS errors without prior information on error model. A likelihood matrix based correction method is firstly proposed. And then a mixed Kalman and H-infinity filter is proposed to further mitigate the NLOS errors. This method does not need much parameters of the measurement model. The simulation results show that the proposed algorithm has the higher localization accuracy, and it is robust to the NLOS errors.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grants no. 61203216 and 61273078 and the Fundamental Research Fund for the

Central Universities of China (N110404030, N110804004, N110404004 and N090304003).

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, Mass, USA, 2005.

[2] W. Li and Y. Jia, "Location of mobile station with maneuvers using an IMM-based cubature Kalman filter," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4338–4348, 2012.

[3] P.-C. Chen, "A non-line-of-sight error mitigation algorithm in location estimation," in *Proceedings of the Wireless Communication and Networking Conference*, vol. 1, pp. 316–320, 1999.

[4] Q. H. Wang, I. Balasingham, M. M. Zhang, and X. Huang, "Improving RSS-based ranging in LOS-NLOS scenario using GMMs," *IEEE Communications Letters*, vol. 15, no. 10, pp. 1065–1067, 2011.

[5] C. D. Wann and H. Y. Lin, "Hybrid TOA/AOA estimation error test and non-line of sight identification In wireless location," *Wireless Communications & Mobile Computing*, vol. 9, no. 6, pp. 859–873, 2009.

[6] L. Cong and W. Zhuang, "Nonline-of-sight error mitigation in mobile location," *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, pp. 560–573, 2005.

[7] İ. Güvenç, C.-C. Chong, F. Watanabe, and H. Inamura, "NLOS identification and weighted least-squares localization for uwb systems using multipath channel statistics," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Article ID 271984, 14 pages, 2008.

[8] S. Venkatraman, J. Caffery, and H. R. You, "A novel ToA location algorithm using LoS range estimation for NLoS environments," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 5, pp. 1515–1524, 2004.

[9] S. Venkatesh and R. M. Buehrer, "Non-line-of-sight identification in ultra-wideband systems based on received signal statistics," *IET Microwaves, Antennas & Propagation*, vol. 1, no. 6, pp. 1120–1130, 2007.

[10] J. F. Liao and B. S. Chen, "Robust mobile location estimator with NLOS mitigation using interacting multiple model algorithm," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3002–3006, 2006.

[11] B. S. Chen, C. Y. Yang, F. K. Liao, and J. F. Liao, "Mobile location estimator in a rough wireless environment using extended Kalman-based IMM and data fusion," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 3, pp. 1157–1169, 2009.

[12] C. Y. Yang, B. S. Chen, and F. K. Liao, "Mobile location estimation using fuzzy-based IMM and data fusion," *IEEE Transactions on Mobile Computing*, vol. 9, no. 10, pp. 1424–1436, 2010.

[13] U. Hammes and A. M. Zoubir, "Robust MT tracking based on M-estimation and interacting multiple model algorithm," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3398–3409, 2011.

[14] W. Li and Y. Jia, "Location of mobile station with maneuvers using an IMM-based cubature Kalman filter," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4338–4348, 2012.

[15] W. Ke and L. Wu, "Mobile location with NLOS identification and mitigation based on modified Kalman filtering," *Sensors*, vol. 11, no. 2, pp. 1641–1656, 2011.

[16] M. Nájar and J. Vidal, "Kalman tracking for mobile location in NLOS situations," in *Proceedings of the 14th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '03)*, pp. 2203–2207, Beijing, China, September 2003.

[17] L. Chen and L. Wu, "Mobile positioning in mixed LOS/NLOS conditions using modified EKF banks and data fusion method," *IEICE Transactions on Communications*, vol. E92-B, no. 4, pp. 1318–1325, 2009.

[18] J. M. Huerta, J. Vidal, A. Giremus, and J.-Y. Tourneret, "Joint particle filter and UKF position tracking in severe non-line-of-sight situations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 874–888, 2009.

[19] M. . Boccadoro, G. De Angelis, and P. Valigi, "TDOA positioning in NLOS scenarios by particle filtering," *Wireless Networks*, vol. 18, no. 5, pp. 579–589, 2012.

[20] W. Wu and F. Zhang, "Robust cooperative exploration with a switching strategy," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 828–839, 2012.

[21] F. Yang, Z. Wang, S. Lauria, and X. Liu, "Mobile robot localization using robust extended H_∞ filtering," *Proceedings of the Institution of Mechanical Engineers. Part I*, vol. 223, no. 8, pp. 1067–1080, 2009.

[22] B. Hassibi, A. H. Sayed, and T. Kailath, "Linear estimation in Krein spaces—part I: theory," *IEEE Transactions on Automatic Control*, vol. 41, no. 1, pp. 18–33, 1996.

[23] B. Hassibi, A. H. Sayed, and T. Kailath, "Linear estimation in Krein spaces—part II: applications," *IEEE Transactions on Automatic Control*, vol. 41, no. 1, pp. 34–49, 1996.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 374796, 11 pages
http://dx.doi.org/10.1155/2013/374796

Research Article

Secure Routing Protocol Using Cross-Layer Design and Energy Harvesting in Wireless Sensor Networks

Nabil Ali Alrajeh,[1] Shafiullah Khan,[2] Jaime Lloret,[3] and Jonathan Loo[4]

[1] *Biomedical Technology Department, College of Applied Medical Sciences, King Saud University, Riyadh 11633, Saudi Arabia*
[2] *Institute of Information Technology, Kohat University of Science and Technology (KUST), Kohat City 26000, Pakistan*
[3] *Department of Communications, Universidad Politecnica de Valencia, Camino de Vera 46022, Valencia, Spain*
[4] *School of Engineering and Information Sciences, Middlesex University, London NW4 4BT, UK*

Correspondence should be addressed to Nabil Ali Alrajeh; nabil@ksu.edu.sa

Received 28 November 2012; Accepted 23 December 2012

Academic Editor: Shuai Li

Energy efficiency is the main concern of research community while designing routing protocols for wireless sensor networks (WSNs). This concern can be addressed by using energy-harvesting scheme in routing protocols. In this paper, we propose a secure routing protocol that is based on cross layer design and energy-harvesting mechanism. It uses a distributed cluster-based security mechanism. In the cross-layer design, parameters are exchanged between different layers to ensure efficient use of energy. Energy-harvesting system is used to extract and store energy, which is used to take decisions for the node state and thus for the routing issues. Simulation results show that our routing protocol can perform better in many scenarios and in hostile attack-prone environment.

1. Introduction

WSN is typically used to monitor environmental or geographical location for some specific purpose. WSN consists of sensor nodes that have the capability of self-configuration and its deployment in target area is so easy. WSNs have some limitations in terms of battery power, data rates, memory, and processing.

Energy efficiency is one of the most important factors in designing a WSN. As WSN is deployed in many hostile and extreme environments, it is not possible to supply energy source or recharging facility. The entire network has to perform its task on the embedded batteries. If some nodes died due to low battery power, it may result in the breakdown of entire network termed as network partitioning [1], so one of the main purposes is to enlarge the WSN lifetime [2]. Built-in power technologies such as batteries are consistently improving [3], and there are many power saving and energy saving techniques for WSNs [4]. However, most of WSNs are deployed in harsh environments in which there is a need of environmental energy harvesting. Energy harvesting is a mechanism in which sensor nodes have the ability to extract energy from environment, store it, and then use it whenever needed. In WSN more energy is used in data transmission from source to multihop away destination. This is the reason; energy-efficient routing is always desirable in such kind of networks [5]. Energy efficiency can be achieved by utilizing clustering mechanism in WSN. Clustering is a technique in which many sensor nodes are grouped together to perform a task. Cluster head is responsible for monitoring all the nodes in its own cluster. In cluster-based WSN, routing mechanism is more simple and easy as compared to noncluster WSN. Cluster head facilitates the routing protocol to reliably send data from source to destination. On the other hand, routing protocol is responsible for finding optimal route from source to destination. In classical OSI model, all the layers operate independently. In such case, routing protocol would select a path regardless of physical layer (battery power) and MAC layer (data rates) requirements. Networks having energy or bandwidth limitations must interact with upper layers

for selecting energy-efficient path. This kind of interaction between different layers is only possible using cross-layer technique.

The idea behind cross-layer information exchange [6, 7] is to optimize network usage and resources by communicating different layers. Cross-layer optimization technique can be used to make intelligent decisions about power saving, QoS routing, enhanced scheduling, and bandwidth allocation algorithms in multihop networks. The important fact of using cross-layer design is to exchange multiple parameters across the protocol stack to increase network performance and efficiency of network resources. Network resources in WSN can be threatened by many security attacks such as sleep deprivation attack, packet dropping attack, or collecting sensitive information [1, 8–10]. The attacker conducts sleep deprivation attack at physical layer while packet dropping attack at network layer. Such kind of multilayered security attacks cannot be prevented by using a security mechanism at single layer. To counter multilayer security attacks, again cross-layer security mechanism is highly desirable for detecting and responding to different attacks at different layers. One possible solution can be cross-layer secure routing.

Secure routing is highly desirable for multihop wireless networks such as WSN. Multihop wireless networks are more vulnerable to security attacks as compared to single-hop wireless networks. The reason is that most of multihop wireless networks are distributed having no centralized body. Designing an appropriate secure routing protocol for WSN is a challenging task. In WSN the ideal routing protocol should be secure and efficient in terms of energy consumption.

In this paper, we present a secure routing protocol which is based on cross-layer information exchange and energy-harvesting technique.

Our proposal is capable to consistently monitor the energy consumption and select secure and energy-efficient path from source to destination.

The rest of the paper is organized as follows. Section 2 discusses related work. Protocol design considerations and parameters are covered in Section 3. Section 4 describes the evaluation and simulation results. Section 5 concludes the paper and provides our future work.

2. Related Work

WSN has many applications such as wide area surveillance for borders security, monitoring heat, sound, and pressure in a given area [11]. Routing packets from source to destination is one of the important operations in WSN. Many Routing protocols have been proposed in the literature [12–16]. Most of these protocols are either application specific or lacking security mechanism. Research community is paying special attention to propose various security mechanisms for WSN [17–20]. Most of these security mechanisms operate and counter specific security threat. Many secure routing protocols are developed for WSN as mentioned in [21–24], which are used to address particular security concern. Furthermore, most of these proposed routing protocols are based on key management schemes to encrypt the data. Although key management scheme is efficient to protect data confidentiality, it cannot prevent data dropping or packet misdirecting kind of attacks [8]. It is also important to mention that most of these existing secure routing protocols operate without taking energy into consideration. Some researchers proposed energy-aware routing protocols for WSN [25–27]. However, most of these energy-aware routing protocols lack security mechanism. Furthermore, these proposed mechanisms have no concept of energy-harvesting mechanism in WSN.

It is important to consider energy limitations while designing any mechanism for WSN. Majority of current energy-aware routing protocols determine efficient use of energy. Such mechanisms may increase the life time of WSN, but do not offer harvesting of environmental energy to provide durable solution.

Research community is now seriously considering such mechanisms for WSN, in which environmental energy is harvested and stored so that to provide a durable source of more energy to sensor nodes especially for those sensor networks which are deployed for long-term activities. Many routing protocols have been proposed so far which are based on the concept of energy harvesting in WSN. Low latency geographic routing using energy harvesting is proposed for WSN [28]. This proposal estimates the energy consumption and the expected energy from harvesting device. The authors made a claim about reliable data delivery with low latency. However, this scheme cannot ensure reliable data delivery in case of security attack or malicious activity in WSN.

Another beaconless geographic routing based on energy-harvesting technique is proposed for WSN [29]. The main idea of this proposal is same as presented in [28] except that its nodes send data packets first instead of control packet and the nodes have no prior information of neighbors. This proposal also harvests energy from harvesting device. However, the performance is yet not known in case of mobility, multimedia traffic, and large network size. Furthermore, security concerns are not addressed in this proposal.

Adaptive opportunistic routing based on energy harvesting technique is proposed in [30]. This proposal considers grouping of nodes and estimating distance of nodes from sink. In this work, the authors assume that all the nodes have energy harvesting capability.

In [31], a routing protocol is proposed on the basis of energy transfer mechanism using electromagnetic waves. Another routing protocol based on the concept of energy harvesting is proposed in [32] for environmental monitoring of sustainable WSN. In this work, the authors equipped WSN networks with two types of node, that is, battery-power-driven nodes and energy-harvesting-driven nodes. Two types of routing are proposed for these two categories of nodes. Authors in [33] proposed a novel mechanism for transmission power control based on energy level and harvesting technique. The authors claim that the problem of unbalanced energy consumption is solved by using unbalanced energy capability. Routing protocol with hybrid energy storage system is proposed [34] to extend the network lifetime with a new cost metric. Another harvesting-aware mechanism [35] is designed for sustainable mobile sensor nodes. In this mechanism, mobile sensor nodes move to energy station

for recharging if the energy is found below threshold value. However, this mechanism is not suitable for static WSN. A detailed work is done on opportunistic routing based on ambient energy harvesting [36]. In this proposal, nodes are grouped together to improve throughput and minimize delay.

3. Proposed Routing Scheme

Data transmission from source to destination node requires some sort of routing mechanism. Typical WSN nodes sense information and forward to sink node over multi-hop intermediate nodes using routing protocol. The objective behind this work is to transmit packets along such path, which is reliable and energy efficient. We assume that each WSN node is equipped with energy harvesting system. It is capable to harvest environmental energy and convert it into electrical energy. The proposed mechanism consists of four important modules as given in Figure 1.

(i) The proposed mechanism is cluster based in which when WSN is deployed for any application, nodes form two-hop cluster for coordination. In cluster-based WSN, the optimal cluster size is two hop as presented in [37].

(ii) Energy consumption can be reduced when nodes only communicate with cluster head. So our energy model is cluster-based WSN.

(iii) Proposed mechanism is cross-layer in nature so that it can get energy parameters at network layer using cross-layer interface.

(iv) The mechanism is secure in nature especially against variety of active and passive attacks.

3.1. Cluster Formation. The first step consists of cluster formations. There are many clustering schemes [38], but we have used the following one. In the start of the network deployment, all the nodes are assumed to have equal battery power. Initially, each node broadcasts a neighbor-discovery message. All nodes in their coverage area will reply with a neighbor-discovery ack message. Thus, network links and topology are built. Then, interested cluster head nodes send a *cluster-invitation* message to all one-hop and two-hop neighbors in order to become its cluster members. One- and two-hop neighbors respond back with cluster-joining message. In cluster-joining message, the node enables hop-count field, so that when cluster head receives cluster-joining message, it can confirm that the distance of the joining member is not more than two hops. Interested cluster head nodes may be defined in advance or selected randomly by the system based on their position. Described message flow system is shown in Figure 2.

We assume that cluster head is aware of its position with respect to sink node. Such location information can be obtained using global positioning system or using built-in configuration.

3.2. Energy Model. WSN is deployed in such areas where wired network is not feasible to maintain and configure. WSN

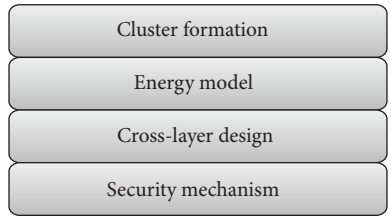


Figure 1: Modules of proposed routing mechanism.

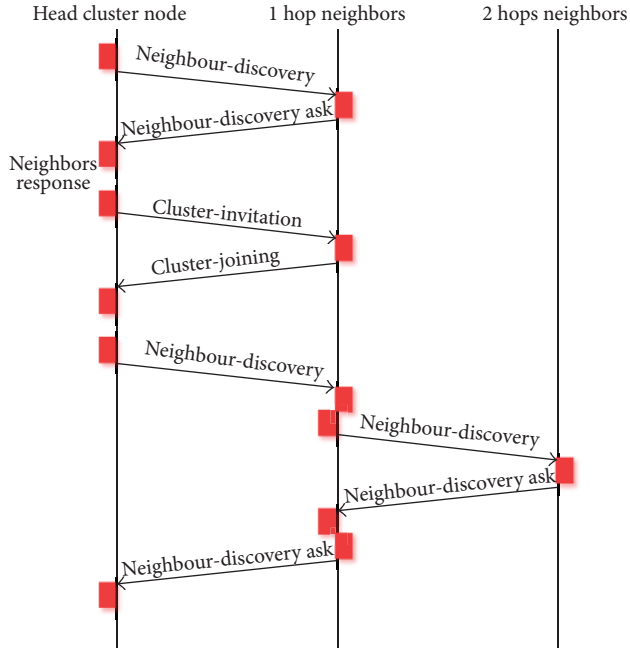


Figure 2: Message flow for cluster formation.

is used to sense information, analyzes them, and transmits to base station. WSN nodes have built-in batteries which determine the network lifetime. The battery life of nodes in WSN depends on the location and other environmental factors. A node that is located in the center of WSN has less battery life as compared to those nodes which are located at edges, because the centre nodes not only collect information around its own field but also forward data for others as well towards the base station. It is not possible to replace or recharge batteries of dying nodes. There is a need for a constant power source for WSN nodes especially for sustainable WSN. Energy harvesting is a promising technique in which sensor nodes are empowered to extract energy from environment, store it, and later on use it for performing different tasks.

Some important sources of energy harvesting are mentioned in [39–41], which are summarized below.

(i) Mechanical vibration is used to create movement which is later on converted to electrical energy using piezoelectric, electrostatic or electromagnetic schemes.

(ii) Photovoltaic cells are used to convert sun light energy into electric energy.

Some other sources of energy are radio frequency (RF) radiation and thermal energy [42]. In RF scheme, high power electromagnetic waves are directed towards sensor nodes from nearby source. In thermal energy-harvesting schemes, sensor nodes have the capability to convert heat energy to electric energy. More investigation is needed to explore all possible merits and demerits of these energy sources especially factors like environmental pollution. Furthermore, some energy harvesting schemes may perform well in one scenario but may not in other. For example, usage of photo cells to harvest solar energy may perform well in those WSN applications where nodes have more exposure to sunlight. Some other challenges are mentioned in [43] such as energy harvesting hardware and software overheads. Enabling energy harvesting in sensor nodes demands specialized hardware to harvest and store energy. Furthermore, specific software is needed to control and manage harvested energy. Such specialized demand of software and hardware will definitely increase the overall cost of WSN deployment. From the literature survey, it is observed that less attention is given to security mechanisms in WSN having energy-harvesting mechanism. Any new mechanism for WSN must consider that as WSNs are deployed in harsh areas, so only battery power may not be sufficient for medium and long-term monitoring. It is indeed necessary to harvest environmental energy so that to provide constant and durable source of energy to all nodes.

In the proposed mechanism, the energy model is considered in next step. We are considering sunlight as the source for harvesting in WSN. For perpetual operation, a duty cycle and energy harvesting mechanism using a mathematical model is present in [44]. Using this equation, we can get the power output from energy source and energy harvested. The proposed equations in [44] also estimate power consumption of a node during specific interval of time. We defined three energy ranges for every sensor node. These three energy ranges define three states of sensor node. The three states are *active state, semiactive state,* and *idle state*. The three states and their characteristics are listed in Table 1.

In active state, WSN node is actively participating in WSN operations, that is, as soon as it sense, or receives any packet, it is immediately routed to cluster head. In active state, node does not harvest environmental energy. A node remains in active state as far as its energy is greater than X. In semiactive state, node starts harvesting environmental energy. A node remains in semiactive state as far as its energy (Y) is in between X and Z.

In semiactive state, a node does not actively participate in WSN operation. It collects and stores packets and later on sends to cluster head. In semiactive state, when node is in the process of harvesting energy, it collects and stores packets. After some time, it stops harvesting process and sends a bulk of packets to cluster head and again starts harvesting energy. It is a kind of sleep and wake state. In sleep state, it only harvests energy and collects packets. When in wake state, it forwards packets to cluster head and stop energy-harvesting process.

Table 1: The energy related states of WSN node.

State	Energy range	Energy harvesting
Active	Above X	No
Semi active	Y	Yes
Idle	Below Z	Yes

In idle state, a node does not perform any operation, only harvest environmental energy. A node remains in idle state as far as its energy is below Z.

In idle state, node calculates its harvested energy after interval of time. If the energy value is greater than "X," then it switches to active state. If the harvested energy is still below "Z," it remains in idle state or otherwise switchs to semiactive state.

Given $E(\mathrm{AR})$ as the energy in active range and $E(\mathrm{SAR})$ as energy in semiactive range, the algorithm for the three states of sensor node is given in Figure 3.

Along a node lifetime, it will be in any of these three states. Now we can define t_a as the amount of time that has been in active mode, t_s as the amount of time that has been in semiactive mode, and t_i as the amount of time that has been in idle range. The node lifetime T can be expressed by

$$T = t_a + t_s + t_i. \tag{1}$$

Now, we can estimate the energy consumed along the node lifetime. It is given by the following expression:

$$E(t) = E(\mathrm{AR}) \cdot t_a + E(\mathrm{SAR}) \cdot t_s + E(\mathrm{IR}) \cdot t_i. \tag{2}$$

$E(\mathrm{AR})$ varies according the number of packets to transmit, packets to receive, acknowledgements to transmit, acknowledgements to receive, and the number of retransmissions during t_a time. Bearing in mind that in a wireless link there is a packet retransmission probability (P_s), because there can be lost or error packets, $E(\mathrm{AR})$ can be given by the following expression:

$$E(\mathrm{AR}) = \left(1 + P_s\right) \cdot \left(E_{\mathrm{TX}} + E_{\mathrm{RX}} + E_{\mathrm{TX\_ACK}} + E_{\mathrm{RX\_ACK}}\right), \tag{3}$$

where E_{TX} is the energy consumed because of the transmitted packets, E_{RX} is the energy consumed because of the received packets, $E_{\mathrm{TX\_ACK}}$ is the energy consumed because of the transmitted acknowledgement packets, and $E_{\mathrm{RX\_ACK}}$ is the energy consumed because of the received acknowledgement packets. If we take into account the following parameters and the energy model for wireless sensor nodes provided in [45], we obtain (4) to estimate $E(\mathrm{AR})$ in free space:

(i) number of packets to be transmitted (n_t),

(ii) average number of bits of each transmitted packet (x_t),

(iii) average number of acknowledgements transmitted for a packet (n_{at}),

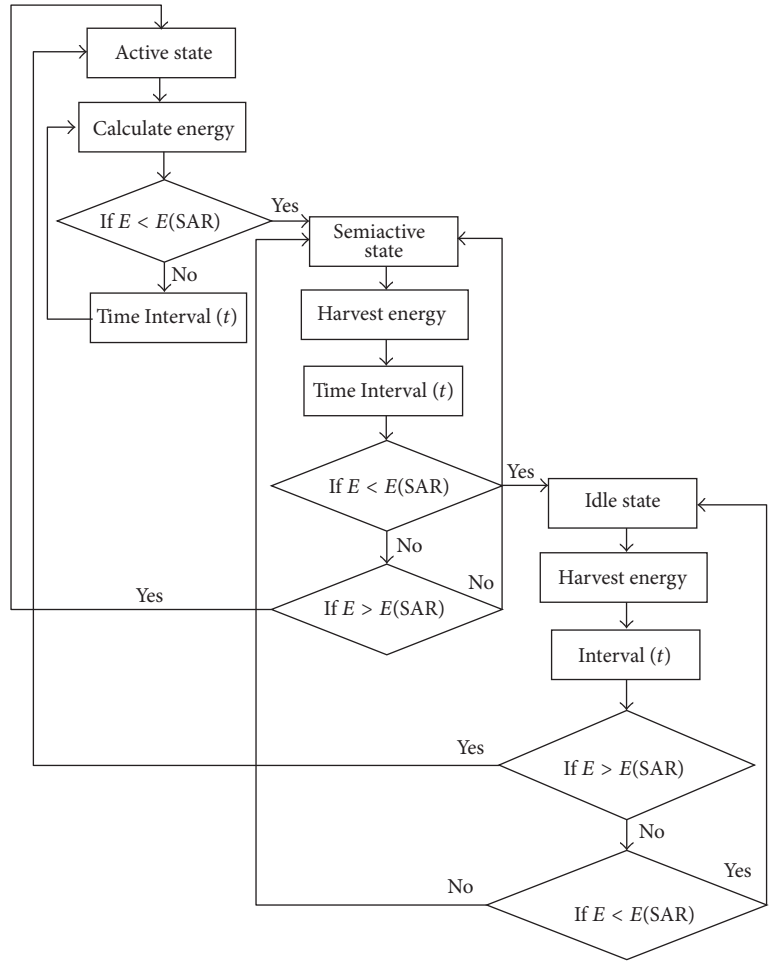


Figure 3: Algorithm for the three states of sensor node.

(iv) average number of bits of each transmitted acknowledgement packet (x_{at}),

(v) number of packets to be received (n_r),

(vi) average number of bits of each received packet (x_r),

(vii) average number of acknowledgements received for a packet (n_{ar}),

(viii) average number of bits of each received acknowledgement packet (x_{ar})

$$E(\text{AR}) = \left(1 + P_s\right) \cdot \left(E_{\text{elec}} \cdot \left(n_t \cdot x_{tr} + n_r \cdot x_{tr} + n_{\text{atr}} \cdot x_{\text{atr}} + n_{\text{atr}} \cdot x_{\text{atr}}\right) + \varepsilon_{\text{amp}} \cdot d^2 \cdot \left(n_t \cdot x_{tr} + n_{\text{atr}} \cdot x_{\text{atr}}\right)\right). \quad (4)$$

We have assumed that generally $x_t = x_r$ (we will call it x_{tr}), $x_{\text{at}} = x_{\text{ar}}$ (we will call it x_{atr}), and $n_{\text{at}} = n_{\text{ar}}$ (we will call it n_{atr}).

When any node switches to idle state, it informs its neighbors. In return, neighbor nodes start routing packets through another route.

For energy-efficient routing, all the member nodes of cluster periodically exchange route energy packets (REP). In REP, nodes communicate energy value. A node always selects that path in which the neighbors have more energy. For example, node A has three one-hop neighbors K, L, and M. Now A will select that neighbor which has more energy.

3.3. Cross-Layer Design. Interaction amongst parameters across the protocol stack is performed using methodology of cross-layer design. In proposed mechanism, the interaction between physical layer and network layer is possible due to this methodology.

REP is generated using cross-layer design. Energy is physical layer scheme, while routing is the mechanism of network layer. To bring current energy value of a node in routing packet is only possible using cross-layer design. In cross-layer design, energy value is first captured at application layer and then inserted to network packet using cross-layer interface [9]. This selection of energy efficient route helps semiactive nodes to harvest more energy and to participate less in WSN operations so that they become active soon. This kind of intelligent routing is possible with cross-layer design.

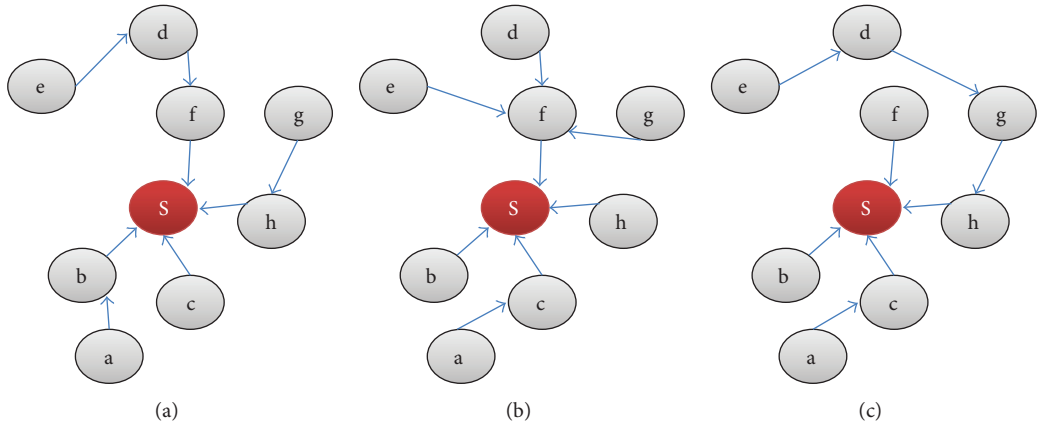


Figure 4: Energy-efficient route selection.

The energy-efficient route selection process is given in Figure 4.

In Figure 4(a), "S" is a cluster head forming a two-hop cluster of many nodes. Initially all the nodes are in active state and as soon as any node gathers some information, it transfers that information immediately to cluster head for further processing. As in active state, environmental energy is not harvested so that all the nodes are using battery power. In WSN, those nodes which are in center or having many neighbors are supposed to consume more energy as they not only gather information and transmit to cluster head but also relay data for all other neighbors. In Figure 4(a), node "b" is not only forwarding its own data to "S," but also relaying data for "a" as well. So "b" energy consumption is more as compared to "a." Here "b" cannot remain in active state for long time. When "b" energy falls in semiactive range, it will switch to semiactive state. Here "a" has an alternate path to start routing data through node "c" as shown in Figure 4(b).

Node "b" will start energy harvesting and will remain in semiactive state. Let suppose node "c" is not there and node "a" has no alternate path. In such case, node "a" will still forward data to node "b" and node "b" will relay data for node "a" after some interval of time. As in semiactive state, node "b" will harvest environmental energy for some time and will forward collected data to cluster head for some time. Let suppose node "b" is in idle state and node "a" has no alternate route to cluster head. In that case, node "a" will collect information and has to wait till it receives a control packet from node "b" about its active or semiactive state.

From Figure 4(a), suppose node "b," "h," and "d" are under heavy traffic load. After some time, these three nodes switch to semiactive state. Their neighbors will start data relaying through other alternate routes as given in Figure 4(b). In this figure, node "a" is now routing data through node "c," node "e" is communicating with cluster head through node "f", while node "g" is relaying data through "f." However, now node "f" is relaying packets for many nodes. Suppose, node "f" switches to semiactive state, then the nodes will reorganize themselves to alternate paths as shown in Figure 4(c).

As nodes periodically communicate REP packet, so all the nodes are informed of neighbors current state. When a neighbor receives REP packet and the energy value in REP packet is in semiactive or idle range, the corresponding neighbors start searching for alternate routes. This kind of mechanism ensures energy-efficient routing in cluster-based sustainable WSN. However, there is a need of some kind of security scheme to ensure reliable data forwarding,

3.4. Security Mechanism. Most of WSNs are used to sense, collect, and process sensitive information. Data confidentiality and integrity is one of the important objectives in such cases. This kind of objective can be achieved by designing some sort of security mechanism especially enabling security mechanism in routing protocol. Important requirement of any network is to ensure confidentiality, integrity, and availability [1, 8, 10]. Confidentiality ensures the secrecy of data sent from source to destination. Integrity makes sure that the destination received data in correct format and sequence without any alteration. Availability means that all the nodes and network devices are operating in harmonious mode and the network resources are available all the time. The attacker uses active or passive attacks to violate either confidentiality of sensitive data or integrity of transmitted data by altering the real information

Different kinds of active and passive attacks can bring serious disruption in overall performance of WSN. Passive attacks [10] do not harm the network or network resources; however, these attacks collect, analyze, and decode sensitive information. Active attacks [1] have the capability to drop or misdirect routing packets. To counter passive attacks and to ensure secrecy and confidentiality of data, we are using similar kind of mechanism as used in [46]. To counter packet dropping or misdirecting kinds of active attacks, we modified a bit the security mechanism proposed in [9]. The security mechanism proposed in [9] sends passive acknowledgement for every successful delivery of packet. For example, if a source node sends 100 packets to destination node through intermediate node(s), the destination node sends back 100

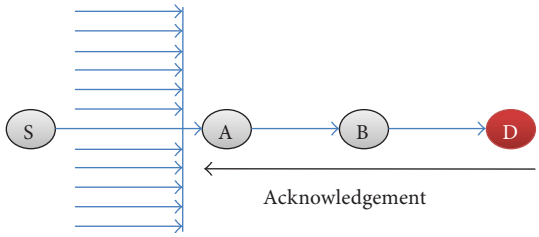


Figure 5: Proposed security mechanism.

passive acknowledgements to source node for every packet arrived. Keeping in view the limited resources and bandwidth, in WSN we cannot use this kind of heavy mechanism. The reason is that, such mechanism will greatly increase routing overheads and could create congestion. In our case, a packet counter is introduced at every node including cluster head. Suppose in Figure 4(a), node "e" forwards 300 packets to cluster head. When cluster head will not receive any further packets from node "e" till fixed interval of time, it will assume that node "e" has no more packets to send. The cluster head will send a packet count of 300 to node "e," which means that cluster head successfully received 300 packets. When node "e" receives packet count from cluster head which matches to its own packet count, it means node "d" is not malicious and all the packets are successfully relayed through node "d." Our mechanism is per session basis contrary to the per packet bases mechanism proposed in [9].

The proposed security model can counter many kinds of active attacks such as blackhole, greyhole, and wormhole [1, 8, 9]. Blackhole is a compromised node and if it is located as intermediate node between source and destination, it is used to drop all the packets passing through it. Greyhole is a less harsh version of blackhole attack. Greyhole is such a malicious node, which is used to selectively drop packets passing through it. Wormhole is basically packet misdirecting attack, in which the attacker establishes a wormhole link between two malicious nodes. The wormhole link is established using fast medium such as fiber optic. One malicious node captures packets at one end and tunnel them through wormhole link to other malicious node. The objective of this attack is to create routing overheads and congestion in network. Our security mechanism is further explained in Figure 5

In Figure 5, "S" is a source node, while "D" is cluster head acting as destination node. Node "A" and "B" are intermediate nodes which relay packets for "S" towards "D." Let us suppose, node "A" is malicious and acting as greyhole. Node "S" sent 12 packets to "A." Node "A" dropped 4 packets. Node "D" received only 8 packets. At the end of the transmission, node "D" sent an acknowledgement to "S" that 8 packets are received successfully. At this stage, node "S" assumes that the next node is malicious and dropping the packets. Now, node "S" starts searching an alternate route to node "D."

Similarly, if node "A" is acting as blackhole or greyhole, the acknowledgement at the end of the session can easily detect such packet dropping or packet misdirecting kind of malicious activities.

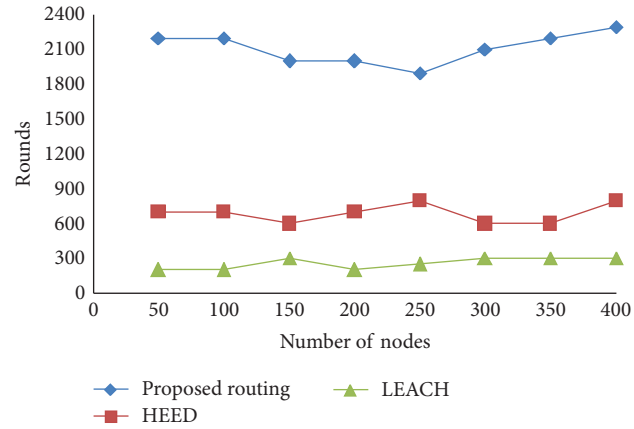


Figure 6: Network life time based on number of rounds.

On the other hand, the mechanism proposed in [9] uses every next hop passive acknowledgement. For example, node "B" sends passive acknowledgment to node "S" for every packet received. Similarly node "D" sends passive acknowledgement to node "A" for every packet received.

4. Performance Evaluation

The performance of secure routing protocol based on cross-layer design and energy harvesting technique is simulated using realistic scenarios. We simulated a WSN having 200 nodes capable of harvesting environmental energy using NS-2. These nodes are randomly deployed at 100 m × 100 m. Each data packet is of 200 bytes, while PER packet size is 40 bytes. We compared our routing mechanism with low energy adaptive clustering hierarchy (LEACH) and hybrid energy-efficient distributed (HEED) cluster-based routing protocol. Figure 6 shows the network lifetime comparison of three routing protocols based on number of rounds.

The performance of the proposed protocol is better as compared to LEACH and HEED.

The reason is that the proposed routing scheme selects energy efficient path to cluster head; furthermore, environmental energy-harvesting mechanism can create great difference in network lifetime.

In Figure 7, the remaining network energy is presented with respect to number of rounds. The total number of nodes is 400 and the network remaining energy is computed for 80 rounds. It is observed that the proposed routing mechanism is better than the rest of two. This difference is again created by the usage of energy harvesting mechanism. The proposed routing scheme is capable to balance the energy usage and harvesting. HEED performance is also satisfactory till the end of 40 rounds. The reason is that HEED is also energy-efficient routing mechanism. However, after 40 rounds, HEED gradually decreases energy value as it has no support of energy harvesting. On the other hand, LEACH performance shows gradual degradation as soon as the number of rounds increases. The reason is that LEACH is not energy efficient in nature.

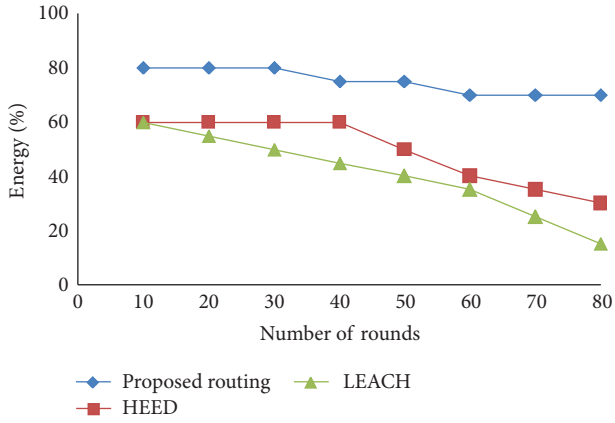


Figure 7: Remaining network energy.

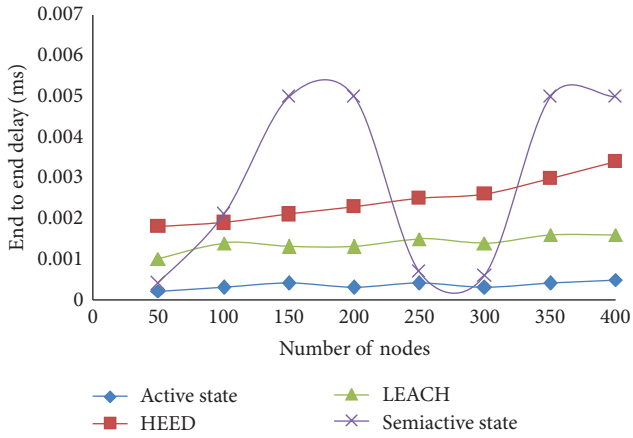


Figure 8: End to end delay.

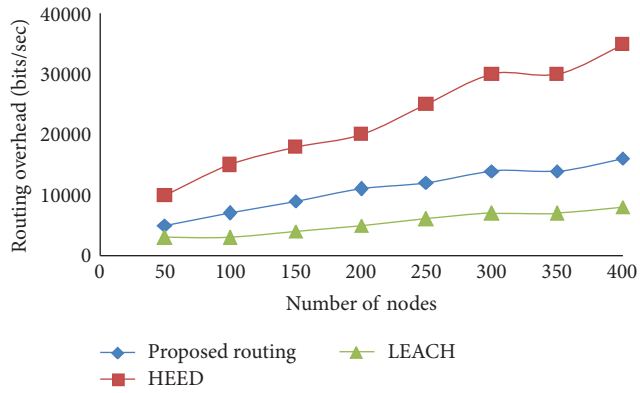


Figure 9: Routing overhead comparison.

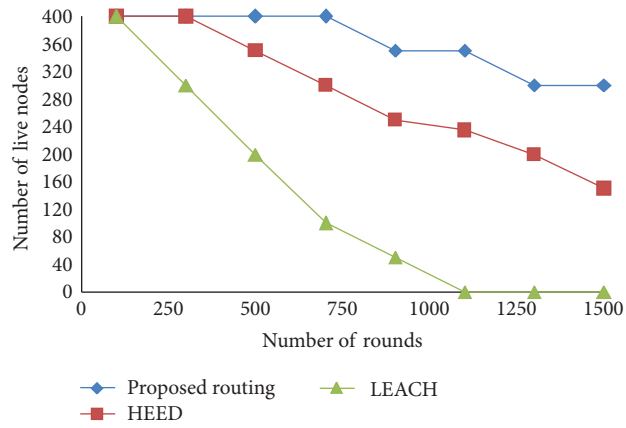


Figure 10: Number of live nodes in increased traffic scenario.

In Figure 8, end to end delay is presented, which shows some interesting results. When the proposed mechanism is in active states, it shows lowest end to end delay from source to cluster head. This is because, the proposed mechanism follows such a path which is rich in energy. However, the proposed scheme shows more end to end delay if some of its nodes are in semi-active state. The reason is that, if a node(s) is in semiactive state, it harvests energy for some time. During energy harvesting period, nodes do not forward packets or take part in communication. In this case, a neighbor has only one route to cluster head through the node in semiactive state. The node has to wait for its neighbor to harvest energy for some time and then forwards its packets through it to cluster head.

Figure 9, compares routing overheads of all three routing protocols with 400 nodes. The proposed routing scheme has more routing overheads as compared to LEACH. It is due to periodic exchange of REP packets to inform the neighbors about energy value.

Figure 10 shows a number of live nodes with respect to rounds in a network of 400 nodes. This simulation result is obtained with increased traffic from sensor nodes to destination. It is observed, that even after 1500 rounds, the number of live nodes in proposed mechanism is more than 300, whereas great performance degradation is seen in LEACH just after 200 rounds.

In Figure 11, a malicious node is introduced to observe the packet loss ratio of all the routing protocols. The malicious node is acting as a greyhole [8]. Greyhole node selectively drops packets which it receives from neighbors. We selected greyhole attack as it represents an entire class of packet dropping and packet misdirecting attacks such as blackhole, sinkhole, jellyfish, and wormhole attacks [1, 9]. This simulation is setup in many sessions. Every session is used to forward 150 packets toward cluster head. Greyhole node is introduced from second session onward (i.e., in sessions 3, 4, 5, and 6). In first two sessions, all the routing protocols successfully forward all the packets without any loss. However, when greyhole malicious node is introduced, almost half of the packets are dropped by all the routing schemes. However, our proposed mechanism adapted a new route from session 4 and onward.

Our proposed mechanism also dropped almost half of the packets in session 3. At the end of session 3, our mechanism waits for response of cluster head to receive a packet counter in which the cluster head will mention the number of packets

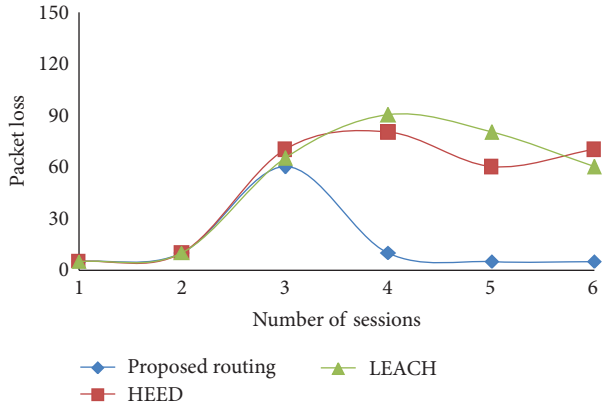


Figure 11: Packet loss in presence of malicious node.

successfully arrived. Cluster head sends packets to nodes by enabling packet counter in it. The node sees that the total number of packets sent to cluster head were 150 while only 70 packets are successfully transferred. At this stage, our routing scheme assumes that the neighbor is malicious and it is dropping packets. That is why in session 4 and onward, packet loss in our routing mechanism is negligible due to selection of alternate path. On the other hand, HEED and LEACH cannot distinguish malicious node in their way.

5. Conclusion

Research community is trying to explore different possibilities to enable energy harvesting in WSN. In this way, the lifetime of sustainable WSN can be increased to a great extent to achieve all goals of sensors deployment. In this paper, we presented in detail a secure routing protocol for WSN, which is based on cross-layer design and energy-harvesting technique. We use a cluster-based approach to group together nodes of two-hop neighbors. Initially all the nodes are in active state, in which nodes actively participate in WSN operations. However, as long as the energy value of sensor node decreases, it switches to semiactive state. In semiactive state, nodes are in wake and sleep conditions. In wake position, nodes take part in network operations, while in sleep position, nodes only harvest environmental energy. In idle state, nodes only harvest energy till it switches back either to active or semiactive states. When compared to other cluster-based routing protocols such as HEED and LEACH, our proposed routing scheme shows better performance in terms of network lifetime, number of live nodes, remaining network energy, and the presence of malicious node.

Some packet loss is observed in our mechanism especially in session 3 as shown in Figure 11. We cannot use per packet acknowledgement as it may result in high routing overheads. We are planning to devise a mechanism in which such packet loss could be reduced. Our future work is to design such distributed algorithm, which is capable of operating in both cluster and non-cluster-based WSN. Furthermore, such mechanism is also desirable since it enables sensor nodes to harvest environmental energy as well as participate in network operations simultaneously. Security mechanism can be improved by using lightweight hash function mechanism or advanced cryptographic scheme to handle active and passive attacks. An interesting technique of artificial neural network (ANN) can be considered to locate those nodes having less remaining energy.

Acknowledgments

The authors extend their appreciation to the Research Centre, College of Applied Medical Sciences, and the Deanship of Scientific Research at King Saud University for funding this research.

References

[1] S. Khan, N. Mast, and J. Loo, "Denial of service attacks and mitigation techniques in IEEE 802.11 Wireless mesh networks," *Information*, vol. 12, pp. 1–8, 2009.

[2] M. Segal, "Improving lifetime of wireless sensor networks," *Network Protocols and Algorithms*, vol. 1, no. 2, pp. 48–60, 2009.

[3] J. M. Gilbert and F. Balouchi, "Comparison of energy harvesting systems for wireless sensor networks," *International Journal of Automation and Computing*, vol. 5, no. 4, pp. 334–347, 2008.

[4] S. Sendra, J. Lloret, M. Garcia, and J. F. Toledo, "Power saving and energy optimization techniques for Wireless Sensor Networks," *Journal of Communications*, vol. 6, no. 6, pp. 439–459, 2011.

[5] A. H. Mohsin, K. Abu Bakar, A. Adekiigbe, and K. Z. Ghafoor, "A survey of energy-aware routing protocols in Mobile Adhoc networks: trends and challenges," *Network Protocols and Algorithms*, vol. 4, no. 2, pp. 82–107, 2012.

[6] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson, "Cross-layer design for wireless networks," *IEEE Communications Magazine*, vol. 41, no. 10, pp. 74–80, 2003.

[7] V. Srivastava and M. Motani, "Cross-layer design: a survey and the road ahead," *IEEE Communications Magazine*, vol. 43, no. 12, pp. 112–119, 2005.

[8] S. Khan, K. K. Loo, N. Mast, and T. Naeem, "SRPM: secure routing protocol for IEEE 802.11 infrastructure based wireless mesh networks," *Journal of Network and Systems Management*, vol. 18, no. 2, pp. 190–209, 2010.

[9] S. Khan and J. Loo, "Cross layer secure and resource-aware on-demand routing protocol for hybrid wireless mesh networks," *Wireless Personal Communications*, vol. 62, pp. 201–214, 2010.

[10] S. Khan, N. Mast, J. Loo, and A. Silahuddin, "Passive security threats and consequences in IEEE 802. 11 wireless mesh networks," *International Journal of Digital Content Technology and Its Applications (JDCTA)*, vol. 2, pp. 4–8, 2008.

[11] M. Frederickson, A publication of the National Electronics Manufacturing Center of Excellence, 2005.

[12] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[13] S. Singh, M. Singh, and D. Singh, "Routing protocols in wireless sensor networks, A survey," *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 1, pp. 25–34, 2010.

[14] A. Popescu, G. Tudorache, B. Peng, and A. Kemp, "Surveying position based routing protocols for wireless sensor and Ad-hoc networks," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 4, no. 7, pp. 41–67, 2012.

[15] O. Fdili, Y. Fakhri, and D. Aboutajdine, "Impact of queue buffer size awareness on single and multi service real-time routing protocols for WSNs," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 4, no. 2, pp. 104–111, 2012.

[16] M. Hussaini, H. Bello-Salau, A. Salami, F. Anwar, A. Abdalla, and M. Islam, "Enhanced clustering routing protocol for power-efficient gathering in wireless sensor network," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 4, no. 12, pp. 18–28, 2012.

[17] A. Kellner, O. Alfandi, and D. Hogrefe, "A survey on measures for secure routing in wireless sensor networks," *International Journal of Sensor Networks and Data Communications*, vol. 1, Article ID 235548, pp. 1–17, 2012.

[18] J. Sen, "A survey on wireless sensor network security," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 1, pp. 55–78, 2009.

[19] K. Xing, "Attacks and countermeasures in sensor networks, a survey," *Springer Network Security*, vol. 7, pp. 534–548, 2005.

[20] V. Kesavan and S. Radhakrishnan, "Multiple secret keys based security for wireless sensor networks," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 4, no. 1, pp. 68–76, 2012.

[21] M. Azeem, K. Khan, and A. Pramod, "Security architecture framework and secure routing protocols in wireless sensor networks-survey," *International Journal of Computer Science & Engineering Survey (IJCSES)*, vol. 2, pp. 189–204, 2011.

[22] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[23] B. Kur, *Secure routing protocols for wireless sensor networks [M.S. thesis]*, Masaryk University Faculty of Informatics, Brno, Czech Republic, 2008.

[24] P. Samundiswary, D. Sathian, and P. Dananjayan, "Secured greedy perimeter stateless routing for wireless sensor networks," *International Journal of Ad Hoc, Sensor & Ubiquitous Computing (IJASUC)*, vol. 1, pp. 9–20, 2010.

[25] M. Liu, J. Cao, G. Chen, and X. Wang, "An energy-aware routing protocol in wireless sensor networks," *Sensors*, vol. 9, no. 1, pp. 445–462, 2009.

[26] M. Younus, A. A. Minhas, M. Y. Javed, and A. Naseer, "EEAR: efficient energy aware routing in wireless sensor networks," in *Proceedings of the 7th International Conference on ICT and Knowledge Engineering (ICTKE '09)*, pp. 57–62, December 2009.

[27] S. Singh, M. Singh, and D. Singh, "A survey of energy-efficient hierarchical cluster-based routing in wireless sensor networks," *International Journal of Advanced Networking and Applications*, vol. 2, pp. 570–580, 2010.

[28] D. Noh, I. Yoon, and H. Shin, "Low-latency geographic routing for asynchronous energy-harvesting WSNs," *Journal of Networks*, vol. 3, no. 1, pp. 78–85, 2008.

[29] O. Jumira, R. Wolhuter, and S. Zeadally, "Energy-efficient beaconless geographic routing in energy harvested wireless sensor networks," *Concurrency and Computation*, vol. 25, no. 1, pp. 58–84, 2013.

[30] Z. Eu and H. Tan, "Adaptive opportunistic routing protocol for energy harvesting wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '12)*, pp. 318–322, June 2012.

[31] R. Doost, K. R. Chowdhury, and M. Di Felice, "Routing and link layer protocol design for sensor networks with wireless energy transfer," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, December 2010.

[32] K. Takahashi, M. Bandai, H. Tan, W. Seah, and T. Watanabe, Least Impact Routing towards Sustainable Sensor Networks Enhanced by Energy Harvesting. White Paper published by Victoria University of Wellington, 2010.

[33] G. Dai, J. Qiu, P. Liu, B. Lin, and S. Zhang, "Remaining energy-level-based transmission power control for energy-harvesting WSNs," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 934240, 12 pages, 2012.

[34] N. Pais, "Cost-benefit aware routing protocol for wireless sensor networks with hybrid energy storage system," *Journal of Green Engineering*, vol. 11, pp. 189–208, 2011.

[35] S. Kim, C. Won, J. Lee, S. Kwon, and Y. Park, "Harvesting aware system for sustainable mobile sensor networks," *International Journal of Hybrid Information Technology*, vol. 5, pp. 199–206, 2012.

[36] Z. A. Eu, H. P. Tan, and W. K. G. Seah, "Opportunistic routing in wireless sensor networks powered by ambient energy harvesting," *Computer Networks*, vol. 54, no. 17, pp. 2943–2966, 2010.

[37] A. Förster, A. Förster, and A. L. Murphy, "Optimal cluster sizes for wireless sensor networks: an experimental analysis," in *Ad Hoc Networks*, vol. 28 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 49–63, 2010.

[38] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[39] S. Chalasani and J. M. Conrad, "A survey of energy harvesting sources for embedded systems," in *Proceedings of the IEEE Conference of Southeastcon*, pp. 442–447, April 2008.

[40] B. Atwood, B. Warneke, and K. S. J. Pister, "Smart dust mote forerunners," in *Proceedings of the 14th IEEE International Conference on Micro Electro Mechanical Systems (MEMS '01)*, pp. 357–360, January 2001.

[41] G. Park, T. Rosing, M. D. Todd, C. R. Farrar, and W. Hodgkiss, "Energy harvesting for structural health monitoring sensor networks," *Journal of Infrastructure Systems*, vol. 14, no. 1, pp. 64–79, 2008.

[42] L. Mateu and F. Moll, "Review of energy harvesting techniques and applications for microelectronics," in *Proceedings of the SPIE Microtechnologies for the New Millennium*, pp. 359–373, May 2005.

[43] C. Moser, *Power management in energy harvesting embedded systems. Doctor of Sciences dissertation [Ph.D. thesis]*, Swiss Federal Institute of Technology Zurich, Zurich, Switzerland, 2009.

[44] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 1, pp. 1–35, 2007.

[45] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual*

Hawaii International Conference on System Siences (HICSS-33), p. 223, Maui, Hawaii, USA, January 2000.

[46] M. Ba, I. Niang, B. Gueye, and T. Noel, "A deterministic key management scheme for securing cluster-based sensors networks," in *Proceedings of the IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC '10)*, pp. 422–427, December 2010.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 145702, 9 pages
doi:10.1155/2012/145702

Research Article

MDS-Based Wormhole Detection Using Local Topology in Wireless Sensor Networks

Xiaopei Lu, Dezun Dong, and Xiangke Liao

College of Computer Science, National University of Defense Technology, Hunan 410073, China

Correspondence should be addressed to Xiaopei Lu, luxp02@gmail.com

Received 28 September 2012; Accepted 27 November 2012

Academic Editor: Shuai Li

Wormhole attack is a severe threat to wireless sensor networks (WSNs), which has received considerable attentions in the literature. However, most of the previous approaches either require special hardware devices or depend on rigorous assumptions on the network settings, which greatly limit their applicability. In this work, we attempt to relax the limitations in prior work, and propose a novel approach to detect wormhole attacks by only local topology information in WSNs. The basic idea is as follows. Each node locally collects its neighborhood information and reconstructs the neighborhood subgraph by multidimensional scaling (MDS). Potential wormhole nodes are detected by validating the legality of the reconstruction. Then, a refinement process is introduced to filter the suspect nodes and to remove false positives. Our approach solely relies on the local connectivity information and is extremely simple and lightweight, which makes it applicable in practical systems. Extensive simulations are conducted, and the results demonstrate the effectiveness and superior performance of our approach.

1. Introduction

Wormhole attack is a severe threat to wireless networks, which has attracted considerable attentions since it was introduced in previous works [1]. Recently, wormhole attack has become a more critical problem, especially in large-scale WSNs [2]. In a wormhole attack, the adversary places two radio transceivers, which are connected through high-speed channel. Each transceiver, captures signals in the network and delivers them to the other end. These signals are replayed, respectively, at the two ends. Then, two distant sensor nodes that are, respectively, around these two transceivers will consider each other as a close neighbor. By building these tunnels, wormhole attacker can fundamentally change the network connectivity, create a set of shortcut paths, attract a large amount of network traffic, and launch many kinds of attacks, such as selectively dropping or modifying packets and breaking the order of packets. Moreover, by attracting network traffic and collecting and analyzing network data, the attacker can perform many other more aggressive and severe attacks, such as denial of service attacks, network hijacking, and man-in-the-middle attacks. Since wormhole attacks are independent of the MAC layer protocol and immune to the cryptographic techniques, most of traditional security mechanisms are vulnerable to them.

To address wormhole attack in WSNs, a number of countermeasures have been proposed in the literature. Those solutions are respectively based on catching different symptoms of wormhole attack. However, most of them have various limitations, for example, requiring additional hardware devices, depending on special assumptions on the network settings. For instance, a number of methods are based on additional hardware devices, such as GPS [3], special radio frequency (RF) hardware [4], and directional antennas [5], which all significantly increase the hardware cost of the systems. Another kind of solutions depends on special assumptions on the network, such as global tight clock synchronization [6], special guarding nodes [7, 8], and attack-free initial networks [9, 10], which all greatly limit their applicability. In order to relax these limitations, a number of topology-based solutions are proposed [11–16]. These methods can detect wormholes by capturing various

symptoms on the network topology, by only exploring the network topology information. However, most of them still have various limitations, for example, centralized algorithms, requiring unit disk graph (UDG) model or relatively high node density, high false positive rate, and so forth. To sum up, wormhole attack has not been well addressed presently, especially in large-scale practical systems.

In this work, we propose a purely new topology-based wormhole detection approach in WSNs. We basically focus on exploring the abnormal structures introduced by wormhole attacks to the network topology. Each node v locally collects its k-hop neighborhood information and obtains the neighborhood subgraph. Then, we construct an estimation distance matrix that consists of the shortest distances (i.e., hop counts) of all node pairs in this subgraph. Next, the estimation distance matrix is used to reconstruct the subgraph and embed it on a plane by *multidimensional scaling (MDS)*, during which each node will be assigned a virtual position (i.e., node coordinates). The basic idea of our wormhole detection approach is based on an important observation as follows. If node v is a normal node, the layout of the MDS would well accord with the estimation distances, which means the distortion factor of the reconstruction would be relatively small. Otherwise, if node v is a wormhole node, its neighborhood subgraph cannot be smoothly embedded on a plane or at least would produce a great distortion factor. Based on this observation, we can detect potential wormhole nodes by validating whether the distortion factor of each node exceeds a threshold. Finally, we propose a simple but novel necessary condition for wormhole links and utilize it to filter the suspect nodes in a *refinement process*. Then, all wormhole nodes and wormhole links can be explicitly identified, with almost no false positives. Figure 1 briefly illustrates the detection results by our approach and the state-of-the-art methods. Black points in the gray regions denote real wormhole nodes, and circles denote detected wormhole nodes by wormhole detection algorithms. The given network graphs in Figures 1(a)–1(d), respectively, present the detection results by MDS-VOW method [12], LCT method [16], and our approach. We can see that MDS-VOW method can hardly work on this kind of wormhole attack, LCT method can detect all wormhole nodes, but with many false positives, and our approach can effectively detect all wormhole nodes with no false positives.

The main contributions of this work are as follows. Our approach does not require any additional hardware devices, but only needs each node to locally collect its k-hop neighborhood information. The algorithm is very simple and the overhead is extremely low, which makes it very applicable in practical WSNs. Moreover, not only can our approach identify all wormhole nodes and wormhole links, but also it produces very few false positives (almost no false positives according to extensive simulations).

The rest of this paper is organized as follows. We discuss related works in Section 2 and introduce the problem formulation in Section 3. Section 4 presents our detection approach in details. We evaluate this design through extensive simulations in Section 5 and conclude this work in Section 6.

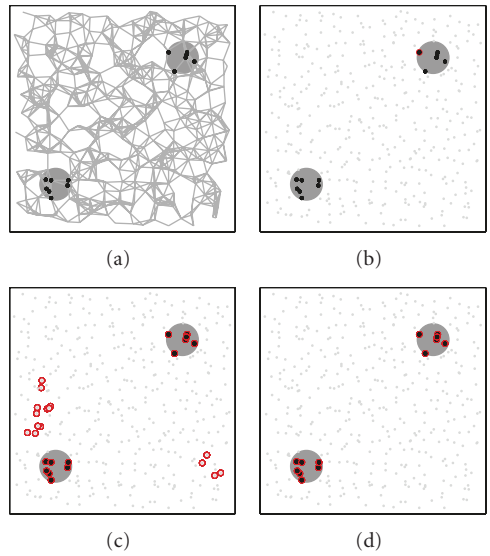


Figure 1: An illustration of wormhole detection results by different approaches. Gray areas denote the impact range of wormhole antennas. Black points in the gray areas denote real wormhole nodes that are directly affected by wormhole antennas. Circles denote detected wormhole nodes by respective detection approaches. (a) The original sensor network. 400 nodes are deployed over a square region. The average node degree is 7.5. Edges connecting wormhole nodes at different ends are omitted. (b) Detection results of MDS-VOW. Most of wormhole nodes are not detected. (c) Detection results of LCT. A number of false positives are produced. (d) Detection result of our approach.

2. Related Work

2.1. Wormhole Detection. A number of countermeasures have been proposed in the literatures. Existing methods are all based on capturing various symptoms induced by wormholes. In this section, we review and analyze the prior work.

The first line of existing solutions is based on the distance or timing analysis of data transmissions. Some methods attempt to detect wormhole attacks by validating the legality of packet traverse distance or time. By appending the location or time information of the sending nodes in each packet, they verify whether the hop-by-hop transmission is physically possible and accordingly detect the wormholes. However, such methods require the preknowledge of node locations by special hardware devices such as GPS [3, 6] or depend on the assumption of accurate globally synchronized clocks to capture the packet propagation time [4, 6]. These methods significantly increase the hardware cost of sensor nodes, and it is unclear whether these techniques would be effective in resource constraint WSNs.

Another line of existing solutions uses special communication devices. Some methods provide physical layer mechanism by using special radio frequency hardware to perform authentications in packet modulation and demodulation [6]. Hu and Evans [5] propose to adopt directional antennas to find and prevent infeasible communication links. The requirement of special hardware devices limits the applicability of these methods.

The third line of existing solutions is based on the discovery and maintenance of node neighborhood. For instance, LiteWorp [7] assumes that the network is attack-free before a time point, and each node collects its 2-hop neighbors. Then, LiteWorp selects a number of guard nodes to detect wormhole channels by overhearing the infeasible transmissions among those nonneighboring nodes. MobiWorp [8] is further proposed to complement LiteWorp by introducing some location-aware mobile nodes. Obviously, the assumption of attack-free environment significantly limits the applicability of these methods.

The forth line of existing solutions detects wormhole attacks by observing the symptom of traffic flow mismatch based on statistical analysis on the network traffic. For instance, Buttyan et al. [9] propose to detect wormhole attacks by capturing the abnormal increase of neighbor number and the decrease of the shortest path lengths that are induced by wormhole channels. This method is centralized because the base station needs to detect wormhole attacks by hypothesis testing based on the prestatistics of normal networks. Another statistical approach [10] is based on the observation that the wormhole links are selected for routing with abnormally high frequency. They identify wormhole links by comparing them with normal network statistics.However, these methods all require the prestatistics of normal network (i.e., attack-free environments).

The last line of existing solutions, which our approach would belong to, is based on the network topology. Wormhole attacks drastically change the network connectivity by introducing fake links among nodes near wormhole antennas, which will result in various abnormal symptoms to the network topology. Lazos et al. [11] present a graph-based framework to tackle wormhole attacks. They assume that a number of guard nodes that have extraordinary communication range exist in the network. The direct communication links between guard nodes and regular nodes would form special geometric structures, and the presence of wormholes would break these structures. Wang and Bhargava propose MDS-VOW [12] to reconstruct the whole network using MDS technique and detect wormhole links by capturing the abnormal features of the "network layout" introduced by wormholes. However, this method is centralized, and it can only work for special cases with only one infected node at both ends of the wormhole attack. In [13], the authors propose a wormhole detection approach with only local connectivity information. In networks with UDG model, their approach can accurately detect wormholes by looking for "forbidden substructures" that should not be present in a normal connectivity graph. However, it is inaccurate under non-UDG graph. Dong et al. [14] propose a distributed connectivity-based wormhole detection method. Each node collects its k-hop neighborhood and checks whether the boundary of its k-hop neighborhood subgraph has one or two circles. Its basic idea is based on the observation that the neighborhood that encloses a wormhole link will have two cycles and single cycle otherwise. However, Wormcircle requires relatively high node density to ensure that boundary detection algorithm works well. In another work [15], they propose to leverage global topological properties to detect wormhole attacks. They consider a legitimate multihop wireless network deployed on the surface of a geometric terrain as a 2-manifold surface of genus 0. Wormholes would introduce singularities or higher genus into the network topology. Ban et al. [16] propose local connectivity test (LCT) to identify wormhole attacks. Their basic idea is that removing the wormhole would disconnect its neighborhood from two components. Their algorithm works well in relatively dense and regular networks but results in many false positives in sparse or random networks.

To sum up, the wormhole attack problem has not been perfectly addressed presently. Existing solutions have various limitations, which make them lack applicability in practical WSNs. In this work, we attempt to propose a new wormhole detection approach to relax the limitations in prior work.

2.2. MDS and Its Applications. Multidimensional scaling was originally a method for visualizing dissimilarity data, which was developed from the behavioral and social sciences for studying the structure of objects. MDS takes a dissimilarity matrix among objects as input and produces a layout of the objects in a low-dimensional space as output. Its basic goal is to create a configuration of objects in a low-dimensional space (e.g., one, two, or three dimensions), and the distances between object pairs are close to the original dissimilarities.

Recently, MDS was applied in WSNs for solving the localization problem. As a fundamental problem in wireless networks, localization problem has been widely studied [17–22]. Shang et al. [17] propose a MDS-based localization algorithm that only relies on mere connectivity information and well tolerates measurement error. Ji and Zha [18] propose a distributed MDS-based sensor localization mechanism that presents a multivariate optimization-based iterative algorithm to calculate the positions of the sensors. In this work, we apply MDS to reconstruct the neighborhood subgraph of each node in WSNs. The input is the distance matrix of all node pairs, and the output is a set of virtual positions of all nodes. The virtual positions are used to calculate a virtual distance matrix of all node pairs. Then, the dissimilarity of these two distance matrices is utilized to evaluate the legality of the reconstruction.

3. Problem Formulation

3.1. Network Model. In our model, a WSN consists of a set of sensor nodes deployed over a plane region. Each node has a unique identity (ID). Nodes are only capable of communicating with other nodes in their proximity. We use G to denote the communication graph, where vertices and edges depict the nodes and communication links, respectively. We do not require the sensor nodes to be equipped with any special hardware, or achieve accurate globally synchronized clocks. Moreover, we do not place any restrictions on the network settings or topology, for example, static or dynamic nodes, node density, communication model, the uniformity of deployment, attack-free initial environment, and so forth. We set an assumption to the network as follows.

Each vertex v in the network G is capable of collecting its k-hop neighbor information. We use $N_G^k(v)$ to denote the neighbors of vertex v that are away from v within k hops in G. Let X be a vertex set in G, and let $G(X)$ be the vertex-induced subgraph by X, which consists of vertexes in X and edges among them. The k-hop neighborhood subgraph of vertex v is denoted by $\Gamma_G^k(v) = G(N_G^k(v) \cup v)$. This assumption is common in the literatures and is realistic in practical WSNs. It is worth noting that k would be a relatively small value, for example, $k = 2$ is sufficient for our algorithm, which makes our approach extremely lightweight.

3.2. Threat Model. In this work, wormhole attacks are defined based on the minimum capabilities required by the attacker to perform these attacks. In particular, the attacker does not need to compromise any node or have any knowledge of the network protocol used. Wormhole endpoints deployed by the adversary do not have valid network identities and do not become part of the network. We assume that in the network exist mechanisms that authenticate legitimate nodes and establish secure links between authenticated nodes. Although wormhole attacks impact neighboring discovery mechanisms in the physical or link layer greatly, transmitted data over encrypted network protocols remain transparent and unobservable to the wormhole attacker. These assumptions are common in prior work [3–6, 12, 13].

Then, we set an assumption on the threat model as follows. Each wormhole link e in network G is long enough to well separate nodes at the two ends of it. We denote nodes at the two ends of e by $V_1(e)$ and $V_2(e)$ and denote the shortest distance between $V_1(e)$ and $V_2(e)$ by $d_G(V_1(e), V_2(e))$. Then, we assume that $d_G(V_1(e), V_2(e)) > 2k$, where $2k$ presents the length of the wormhole attack, that is, the shortest distance between nodes at the two ends of the wormhole without wormhole links. The length of the wormhole determines the threat level of the wormhole attack. Longer wormholes are more dangerous because they have larger impact range and longer impact distance. For a short wormhole attack, its impact on the network connectivity would be negligible since only a small fraction of nodes are affected.

4. Local MDS-Based Wormhole Detection

In this section, we present the analysis and design details of our MDS-based wormhole detection approach.

4.1. Overview of Our Approach. Wormhole attacks introduce essential changes to the network topology. In order to detect wormhole attacks by only topology information, we have to capture the typical topological characteristics of wormhole links. The main idea of our detection approach is based on an observation as follows.

Each node v in the network G collects its k-hop neighborhood information, in particular, $k = 2$. The shortest distances (i.e., hop count) between all node pairs in the neighborhood subgraph $\Gamma_G^k(v)$ are used to construct an estimation distance matrix. Then, the distance matrix is used to reconstruct the subgraph by applying MDS on the subgraph and embedding it on a plane. There would be two conditions. First, if v is a normal node, the reconstructed subgraph would be relatively approximating to the original network. Thus, the embedded distance between each node pair would be relatively close to their estimation distance. Otherwise, if v is a wormhole node, its 2-hop neighborhood subgraph would contain all the wormhole nodes. Topologically, each wormhole node would connect with all nodes at the other end. Therefore, if we still constrainedly embed the subgraph on a plane, the distance constraints cannot be well maintained during the reconstruction. Based on this observation, we let all nodes in the network perform local MDS-based reconstruction and detect potential wormhole nodes according to the legality of their reconstructions. Additionally, we introduce a simple and effective necessary condition of wormholes to filter the suspect nodes detected by the previous process. Through this refinement process, we can remove most of false positives and identify all wormhole links.

As discussed previously, our detection approach mainly includes two components: (1) performing local MDS-based reconstruction and (2) performing refinement process. The first component obtains a number of suspect wormhole nodes. The second component filters the suspect nodes and presents the final detection results. We, respectively, describe these two components in detail as follows.

4.2. Local MDS-Based Reconstruction. For ease of representation, we divide this component into three subprocesses, as described hereinafter.

4.2.1. Distance Estimation. For an arbitrary node v in network G, it first collects its k-hop neighborhood information and obtains its k-hop neighborhood subgraph $\Gamma_G^k(v)$. Next, a classical shortest-path algorithm, for example, Dijkstra's shortest path algorithm, is applied to calculate the shortest distances between all node pairs in $\Gamma_G^k(v)$. Then, the shortest distance matrix $M[\Gamma_G^k(v)]$ is constructed, which is an $n \times n$ matrix (n denotes the number of nodes). Each element in $M[\Gamma_G^k(v)]$ is utilized as the estimation distance between each node pair.

4.2.2. Network Reconstruction. Using the shortest distance matrix $M[\Gamma_G^k(v)]$ as input parameter, we apply MDS to reconstruct the k-hop neighborhood subgraph of v. We denote the reconstructed network by $\overline{\Gamma}_G^k(v)$, in which each node would be assigned a virtual position (i.e., node coordinations). Then, the Euclidian distance between each node pair is calculated in $\overline{\Gamma}_G^k(v)$, and a virtual distance matrix $M[\overline{\Gamma}_G^k(v)]$ is produced.

4.2.3. Wormhole Judgement. Then, we describe how to decide whether a node is a wormhole node candidate by its reconstructed neighborhood subgraph. First, the distortion factor of the MDS reconstruction is calculated for each node v. The distortion factor is defined as follows.

Definition 1 (distortion factor). The distortion factor $\lambda(v)$ is defined as the root mean square error (RMSE) between the shortest distance matrix $M[\Gamma_G^k(v)]$ and the reconstructed virtual distance matrix $M[\overline{\Gamma}_G^k(v)]$, that is, $\lambda(v) = \sqrt{(1/(n \times n)) \sum_{i=1,j=1}^{n} (M[\overline{\Gamma}_G^k(v)](i,j) - M[\Gamma_G^k(v)](i,j))^2}$.

As discussed previously, each node produces large distortion factor if it is a wormhole node and little distortion factor otherwise. Based on this observation, we set a predefined threshold and label nodes that produce distortion factors above this threshold as suspect wormhole nodes. In our experiment, we set the threshold to be the median value of the distortion factors of all nodes in G, that is, $\lambda_{\text{threshold}} = (\lambda_{\max} + \lambda_{\min})/2$ and $\lambda_{\max} = \max\{\lambda(v) : v \in V(G)\}$, $\lambda_{\min} = \min\{\lambda(v) : v \in V(G)\}$, respectively.

Then, we present an efficient way to generate the threshold and distribute it to all nodes. Each node floods a message that contains its distortion factor and records the maximum and minimum values of all distortion factors in all flooding messages it receives. Each node only relays messages that contain a new maximum or minimum value. Thus, only two messages that, respectively, contain the globally maximum and minimum values would be flooded to the whole network. After the flooding is finished, each node calculates the threshold from the maximum and minimum values it records and compares it with its own distortion factor. If its distortion factor exceeds the threshold, it is labeled as a suspect wormhole node and normal node otherwise.

After the implement of this component, a number of suspect wormhole nodes are produced.

4.3. Performing Refinement Process. There is still an issue to be addressed. Some normal nodes may be wrongly labeled as suspect wormhole nodes, and false positives will be introduced. Too many false positives would result in normal links being removed and consequentially degrade the network capacity. In order to address this issue, we introduce this refinement process to filter the suspect nodes and remove false positives. By fully investigating the topology changes introduced by wormholes, we are able to capture some typical topological characteristics of wormhole links. Let X and Y denote two sets that, respectively, contain wormhole nodes at the two ends of a wormhole in network G; let $X \times Y$ denote the edge set between an arbitrary node pair $x \in X$ and $y \in Y$. Then, we present Theorem 2.

Theorem 2. *Given a network graph G and two wormhole node sets X and Y, the following two conditions hold.*

(1) *The subgraph G' that contains node set $X \cup Y$ and edge set $X \times Y$ is a complete bipartite subgraph of G.*

(2) *In the subgraph G'', which is constructed by removing all edges in $X \times Y$ from G, the k-hop neighbor sets of an arbitrary vertex pair $x \in X$ and $y \in Y$ have no common elements, that is, $N_{G''}^k(x) \cap N_{G''}^k(y) = \emptyset$.*

Proof. We first prove condition 1. Because X and Y, respectively, contain and only contain nodes at the two ends of a wormhole, each node v at one end is given the illusion that all nodes at the other end are its direct neighbors. Thus, there will be an edge between v and each node at the other end. According to the construction of G', it will obviously be a complete bipartite subgraph of G.

We then prove condition 2. If there are two nodes $x \in X$ and $y \in Y$ and $N_{G''}^k(x) \cap N_{G''}^k(y) \neq \emptyset$, the shortest distance between x and y must be less than $2k$, that is, $d(x,y) < 2k$. Consequentially, the shortest distance between node sets X and Y would be less than $2k$, that is, $d_{G''}(X,Y) < 2k$, which will contradict with our assumption in the threat model. □

Theorem 2 is a necessary condition of wormholes and is utilized to filter suspect wormhole nodes. First, all connected components are found in these suspect nodes. We denote the set of such connected components by $\mathcal{C}$. Isolated nodes can be certainly excluded. Next, all maximal complete bipartite subgraphs (MCBSs) are found in these connected components. In order to improve the detection rate, we expand each connected component by adding all 1-hop neighbors of the nodes in the component into this component. By doing this, all wormhole nodes can be included in the component if at least one wormhole node at both ends of the wormhole is suspect node. The algorithm in [23] that finds the maximal complete bipartite subgraphs in any graph is applied on each $C \in \mathcal{C}$. Let $\mathcal{B}$ be the set of maximal complete bipartite subgraphs generated by this algorithm, and let $B = (X, Y)$ be an element in $\mathcal{B}$, where X and Y are the two partitions of the bipartite graph. Then, condition 2 in Theorem 2 is applied on each $B \in \mathcal{B}$. If $N_{G''}^k(X) \cap N_{G''}^k(Y) = \emptyset$, all nodes in B will be labeled as final wormhole nodes. Otherwise, they are excluded. Till now, the final detection results are produced.

Moreover, our ultimate goal of detecting wormhole attacks is to neutralize them without breaking regular network functions. In particular, we want to eliminate the high volume of traffic passing through the wormhole links that create the wormhole effect with keeping the sensing and computational capabilities of the nodes. After detecting all wormhole nodes, this can be easily done by removing edges $X \times Y$ in each bipartite subgraph $B \in \mathcal{B}$.

4.4. Algorithm and Discussion. We present Algorithm 1 that describes our wormhole detection approach. Then, several parameters that may influence the performance of our algorithm are discussed as follows.

First, we discuss the influence of k. In our simulations, k is set to be small constant $k = 2$. The reasons are twofold. First, small k introduces low communication overhead of each node for collecting its k-hop neighborhood information. Second, if v is a wormhole node, its 2-hop neighbors would cover all wormhole nodes. Therefore, setting $k = 2$ is sufficient for capturing the abnormal embedding chrematistics induced by this wormhole. Actually, setting k to be a larger value is even adverse to the detection, because larger k induces larger subgraph, which will reduce the proportion of wormhole nodes in the subgraph and accordingly degrade the distinguishability of wormhole nodes.

Then, we discuss the influence of $\lambda_{\text{threshold}}$. The selection of the threshold dramatically impacts the detection accuracy

Input:
A network graph $G(V, E)$.
Output:
A set of complete bipartite graphs $\mathcal{B}$.
(1) **for** each $v \in V$ **do**
(2) Collect k-hop neighborhood subgraph $\Gamma_G^k(v)$.
(3) Calculate the shortest distance matrix $M[\Gamma_G^k(v)]$.
(4) Reconstruct the subgraph by MDS.
(5) Calculate the virtual distance matrix $M[\overline{\Gamma}_G^k(v)]$.
(6) Calculate the distortion factor $\lambda(v)$.
(7) Flood $\lambda(v)$ to the network.
(8) Calculate the threshold $\lambda_{\text{threshold}}$.
(9) **if** $\lambda_v > \lambda_{\text{threshold}}$ **then**
(10) Add v to the suspect node set S.
(11) **end if**
(12) **end for**
(13) Find all connected components $\mathcal{C}$ from S.
(14) **for** each $C \in \mathcal{C}$ **do**
(15) Find each MCBS B from C.
(16) Add B to the MCBS set $\mathcal{B}$.
(17) **end for**
(18) **for** each $B = \{X, Y\}$ in $\mathcal{B}$ **do**
(19) **if** $N_{G''}^k(X) \cap N_{G''}^k(Y) = \varnothing$ **then**
(20) Remove edges $X \times Y$.
(21) **else**
(22) Remove B from $\mathcal{B}$.
(23) **end if**
(24) **end for**

Algorithm 1: Our wormhole detection algorithm.

of our approach. In particular, lower threshold guarantees to catch all wormhole nodes, but causes more false positives, which will increase the workload of refinement process. Otherwise, a higher threshold induces fewer false positives but may produce false negatives. Comparatively, we are more concerned with detecting all wormhole nodes. Therefore, our approach will be on the aggressive side and select a relatively lower threshold. In our simulations, the threshold is set to be the median value of all distortion factors. Moreover, it is also a concerning issue, which makes the generation and distribution of the threshold easier.

5. Evaluation

In this section, we conduct extensive simulations to evaluate the effectiveness and performance of our design and compare it with the state-of-the-art methods.

5.1. Simulation Setup

5.1.1. Node Deployment. Two node deployment models are used: perturbed grid and random deployment. Perturbed grid model is adopted [24] to approximate manual deployments of nodes, in which all nodes are placed on an $m \times n$ grid and perturbed around their initial positions with a perturbed ratio p. Let each cell in the grid be a square with edge length d. Then, the node with coordinate (x, y) will be randomly placed in the region $[x - pd, x + pd] \times [y - pd, y + pd]$. In random deployment model, each node is assigned a coordinate randomly drawn from the network field.

5.1.2. Communication Model. Although our approach does not require specific communication models, both UDG and quasi-UDG models are adopted to build the networks. In the UDG model, there is a link between nodes u and v if and only if their distance is no larger than R, where R is the communication radius. In quasi-UDG model, nodes u and v have a link if their distance is no larger than ρR and have a link with probability q if their distance is within $[\rho R, R]$, where $0 < \rho < 1$.

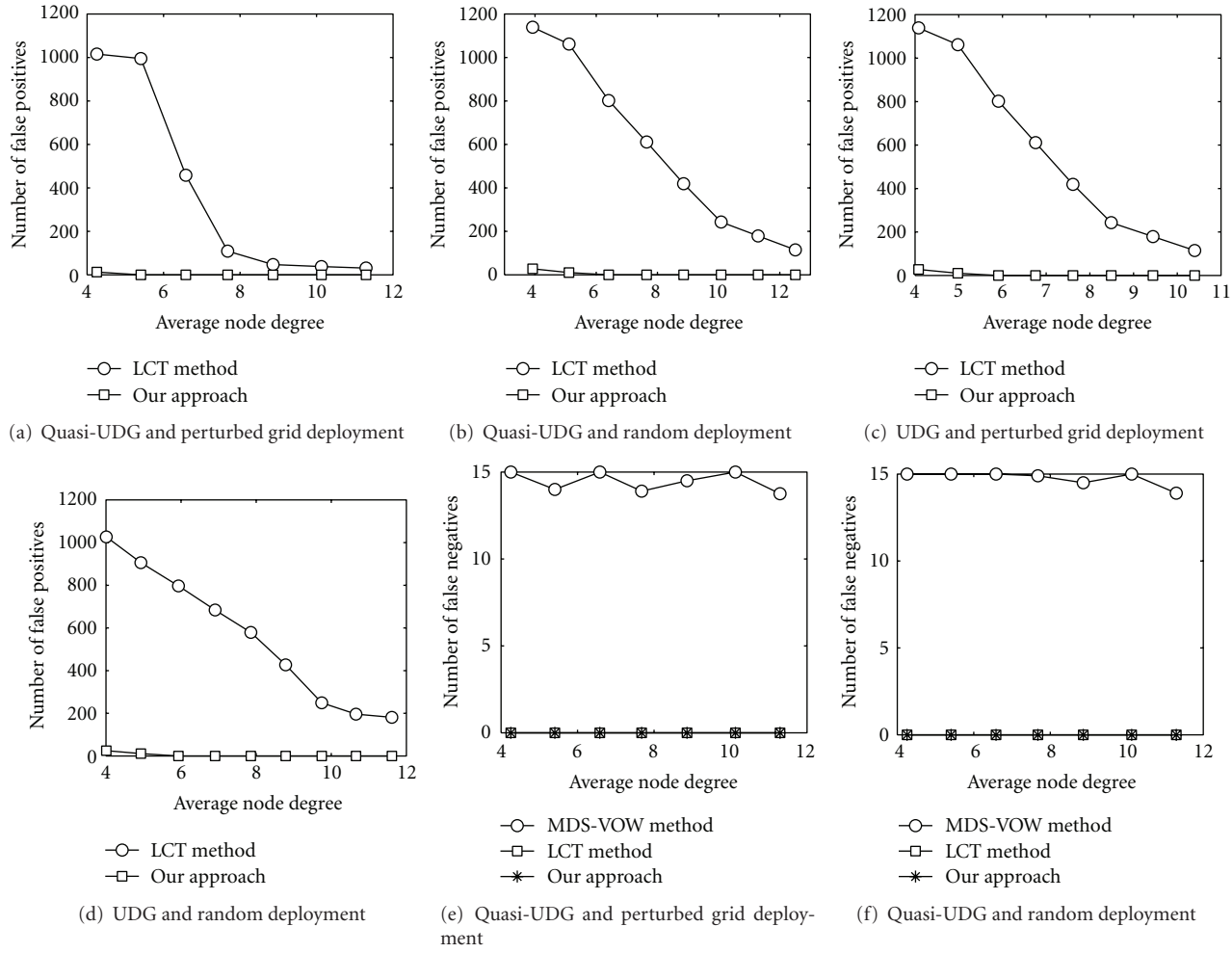
5.1.3. Wormhole Position. The wormhole position is a crucial factor for wormhole detection, because it could impact the significance of wormhole symptoms. Especially when multiple wormholes exist in the network, their relative position will dramatically influence the wormhole detection. In the simulations, our approach is evaluated for detecting wormholes placed at different positions of the network. Moreover, multiple wormholes with different relative positions are also evaluated.

5.2. Simulation Results. In this subsection, we present the results of the simulations under various network settings and compare them with the state-of-the-art MDS-VOW [12] and LCT [16] methods.

The basic network contains 1600 nodes deployed over a square region. In all simulations, $p = 2$ for perturbed grid model, and $\rho = 0.75$ for quasi-UDG model. The average node degree varies from 4 to 13. A set of wormhole nodes are placed at the diagonal of the network. The average number of wormhole nodes is 15. We require all algorithms to detect wormholes that are not shorter than 8 hops, that is, the shortest distance between nodes at the two ends of the wormhole is not less than 8. All simulations take 100 runs with random network generation and present the average results.

First, four sets of simulations are conducted to evaluate the number of false positives of our approach. Each set of simulations adopts different node deployments and communication models. The results are, respectively, presented in Figures 2(a)–2(d). From the results, we can obtain several observations as follows.

5.2.1. Influence of Node Density. The results in Figure 2 indicate that the number of false positives decreases for all approaches as the node degree increases. And our approach always greatly outperforms LCT method. However, when the degree is very low, there are still some false positives. The reason is analyzed as follows. In extremely sparse networks, there would be some special cases called bridge links, as shown in Figure 3. Although it is a normal link in the network, it topologically accords with the property of wormhole links. Some of these links may be wrongly labeled as wormhole candidates in MDS-based reconstruction and cannot be filtered by the refinement process.



(a) Quasi-UDG and perturbed grid deployment
(b) Quasi-UDG and random deployment
(c) UDG and perturbed grid deployment
(d) UDG and random deployment
(e) Quasi-UDG and perturbed grid deployment
(f) Quasi-UDG and random deployment

Figure 2: Simulation results. 1600 nodes are deployed over a square region. The average node degree varies from 4 to 13. In all simulations, $p = 2$ for perturbed grid model, and $\rho = 0.75$ for quasi-UDG model. A wormhole is launched at the diagonal of the network. The average number of wormhole nodes is 15. (a)–(d) evaluate the number of false positives under various network settings. (e)-(f) evaluate number of false negatives.

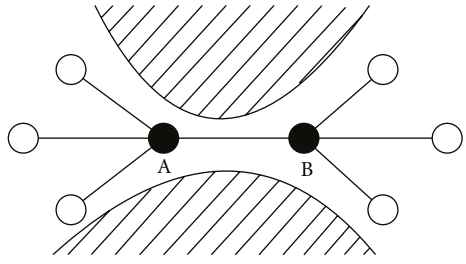


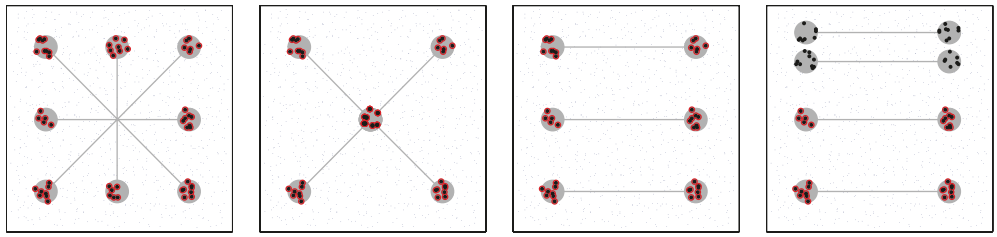
Figure 3: An example of bridge link. Link between nodes A and B may be aggressively labeled as wormhole link. The hatched areas denote holes of network deployment.

5.2.2. Influence of Node Deployment. It is shown in Figure 2 that our approach always produces few false positives for both perturbed grid distribution and random distribution. LCT produces fewer false positives for perturbed grid model than random deployment model. The reason is that perturbed grid model produces more regular networks.

5.2.3. Influence of Communication Model. Figure 2 demonstrates that our approach is not clearly influenced by the communication model. And it also demonstrates that our approach always induces much fewer false positives under both UDG and quasi-UDG models.

Then, we evaluate the number of false negatives of our approach, as shown in Figures 2(e) and 2(f). The results show that our approach can always detect all wormhole attacks. More results are constant under UDG model and are omitted here. The MDS-VOW method cannot even detect any wormholes because it does not work for the general wormhole model.

More simulations are conducted by placing wormholes at different positions in the network. The results are constant and are omitted due to the space limit. To sum up, our approach still works well in sparse and irregular networks



(a) (b) (c) (d)

Figure 4: Detection results for multiple wormholes. 900 nodes are deployed over a square region. Perturbed grid deployment with $p = 1.5$ and quasi-UDG with $\rho = 0.75$ are adopted to generate the networks. The average node degree is 7.5. Multiple wormholes are placed at different positions in the network.

and is not clearly influenced by the communication model. Moreover, our approach produces few false positives. It is worth noting that LCT method can obtain better results by increasing the shortest length of wormholes required to be detected. However, that will greatly restrict its applicability and increase the communication and computation cost.

5.3. Multiple Wormholes. In this subsection, our approach is evaluated for detecting multiple wormholes.

When the distance between two different wormholes is long enough, they will not affect each other. Thus, our approach can well detect all wormhole nodes, as shown in Figures 4(a)–4(c). Otherwise, if multiple wormholes are close, they may interfere with each other, which makes the detection more difficult. Particularly, if the distances of both ends of the wormholes are relatively short, as shown in Figure 4(d), our approach fails to detect the wormholes. The reason is as follows. When both ends of two wormholes are very close to each other, wormhole nodes at different ends are connected by short paths through wormhole links in the adjacent wormhole. Therefore, these nodes would be filtered during the refinement process. Actually, to the best of our knowledge, this situation cannot be solved by any purely topology-based detection methods.

6. Conclusions

As a severe threat to WSNs, wormhole attack has received considerable attentions during the past decade. However, most of existing countermeasures lack applicability for requiring special hardware devices or depending on rigorous assumptions on the network. In this work, we fundamentally analyze the essential wormhole symptoms by topological methodology and propose a local MDS-based wormhole detection approach. Our approach does not depend on any hardware requirements and is extremely simple and lightweight, which make it quite feasible in practical WSNs. Extensive simulations are conducted, and the results show that our approach can effectively identify all wormhole nodes for a large class of network instances.

Acknowledgments

The first author is supported by the National Natural Science Foundation of China (NSFC) under Grants no. 60903224 and no. 61202484. D. Dong is supported by NSFC under Grants no. 61272482 and no. 61170261.

References

[1] K. Sanzgiri, B. Dahill, B. Levine, and F. Belding-Royer, "A secure routing protocol for Ad Hoc networks," in *Proceedings of the IEEE International Conference on Network Protocols (IEEE ICNP '02)*, 2002.

[2] X. Mao, X. Miao, Y. He, X.-Y. Li, and Y. Liu, "CitySee: urban CO_2 monitoring with sensors," in *Proceedings of the 32nd IEEE International Conference on Computer Communications (IEEE INFOCOM '12)*, 2012.

[3] W. Wang, B. Bhargava, Y. Lu, and X. Wu, "Defending against wormhole attacks in mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 6, no. 4, pp. 483–503, 2006.

[4] S. Capkun, L. Buttyan, and J. P. Hubaux, "Sector: secure tracking of node encounters in multi-hop wireless networks," in *Proceedings of the ACM Workshop on Security in Ad Hoc and Sensor Networks (ACM SASN '03)*, 2003.

[5] L. Hu and D. Evans, "Using directional antennas to prevent wormhole attacks," in *Proceedings of the Network and Distributed System Security Symposium Conference (NDSS '04)*, 2004.

[6] Y. C. Hu, A. Perrig, and D. B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (IEEE INFOCOM '03)*, pp. 1976–1986, April 2003.

[7] I. Khalil, S. Bagchi, and N. B. Shroff, "LITE WORP: a lightweight countermeasure for the wormhole attack in multihop wireless networks (DSN '05)," in *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 612–621, July 2005.

[8] I. Khalil, S. Bagchi, and N. B. Shroff, "MOBIWORP: mitigation of the wormhole attack in mobile multihop wireless networks," in *Proceedings of the Securecomm and Workshops (SECURECOMM '06)*, September 2006.

[9] L. Buttyan, L. Dora, and I. Vajda, "Statistical wormhole detection in sensor networks," in *Proceedings of the Security and Privacy in Ad-hoc and Sensor Networks (IEEE ESAS '05)*, vol. 3813, pp. 128–141, 2005.

[10] N. Song, L. Qian, and X. Li, "Wormhole attacks detection in wireless ad hoc networks: a statistical analysis approach," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, April 2005.

[11] L. Lazos, R. Poovendran, C. Meadows, P. Syverson, and L. W. Chang, "Preventing wormhole attacks on wireless ad hoc networks: a graph theoretic approach," in *Proceedings of the IEEE Wireless Communications and Networking Conference, Broadband Wirelss for the Masses—Ready for Take-off (WCNC '05)*, pp. 1193–1199, March 2005.

[12] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," in *Proceedings of the ACM Workshop on Wireless Security (WiSe '04)*, pp. 51–60, October 2004.

[13] R. Maheshwari, J. Gao, and S. R. Das, "Detecting wormhole attacks in wireless networks using connectivity information," in *Proceedings of the 26th IEEE International Conference on Computer Communications (IEEE INFOCOM '07)*, pp. 107–115, May 2007.

[14] D. Dong, M. Li, Y. Liu, and X. Liao, "WormCircle: connectivity-based wormhole detection in wireless ad hoc and sensor networks," in *Proceedings of the 15th International Conference on Parallel and Distributed Systems (ICPADS '09)*, pp. 72–79, December 2009.

[15] D. Dong, M. Li, Y. Liu, X. Y. Li, and X. Liao, "Topological detection on wormholes in wireless ad hoc and sensor networks," in *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP '09)*, pp. 314–323, October 2009.

[16] X. Ban, R. Sarkar, and J. Gao, "Local connectivity tests to identify wormholes in wireless networks," Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc '11), 2011.

[17] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proceedings of the PROCEEDINGS OF The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '03)*, pp. 201–212, June 2003.

[18] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proceedings of the IEEE Computer and Communications Societies (IEEE INFOCOM '04)*, pp. 2652–2661, March 2004.

[19] S. Li and F. Qin, "A dynamic neuralnetwork approach for solving nonlinear inequalities defined on a graph and Its application to distributed, routing-free, range-free localization of WSNs," *Neurocomputing*. In press.

[20] S. Li, Y. Lou, and B. Liu, "Bluetooth aided mobile phone localization: a nonlinear neural circuit approach," *Transactions on Embedded Computing Systems*. In press.

[21] S. Li, B. Liu, B. Chen, and Y. Luo, "Neural network based mobile phone localization using bluetooth connectivity," *Neural Computing and Applications*. In press.

[22] S. Li, Z. Wang, and Y. Li, "Using laplacian eigenmap as heuristic information to solve nonlinear constraints defined on a graph and its application in distributed range-free localization of wireless sensor networks," *Neural Processing Letters*. In press.

[23] D. Eppstein, "Arboricity and bipartite subgraph listing algorithms," *Information Processing Letters*, vol. 51, no. 4, pp. 207–211, 1994.

[24] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 247–260, 2006.