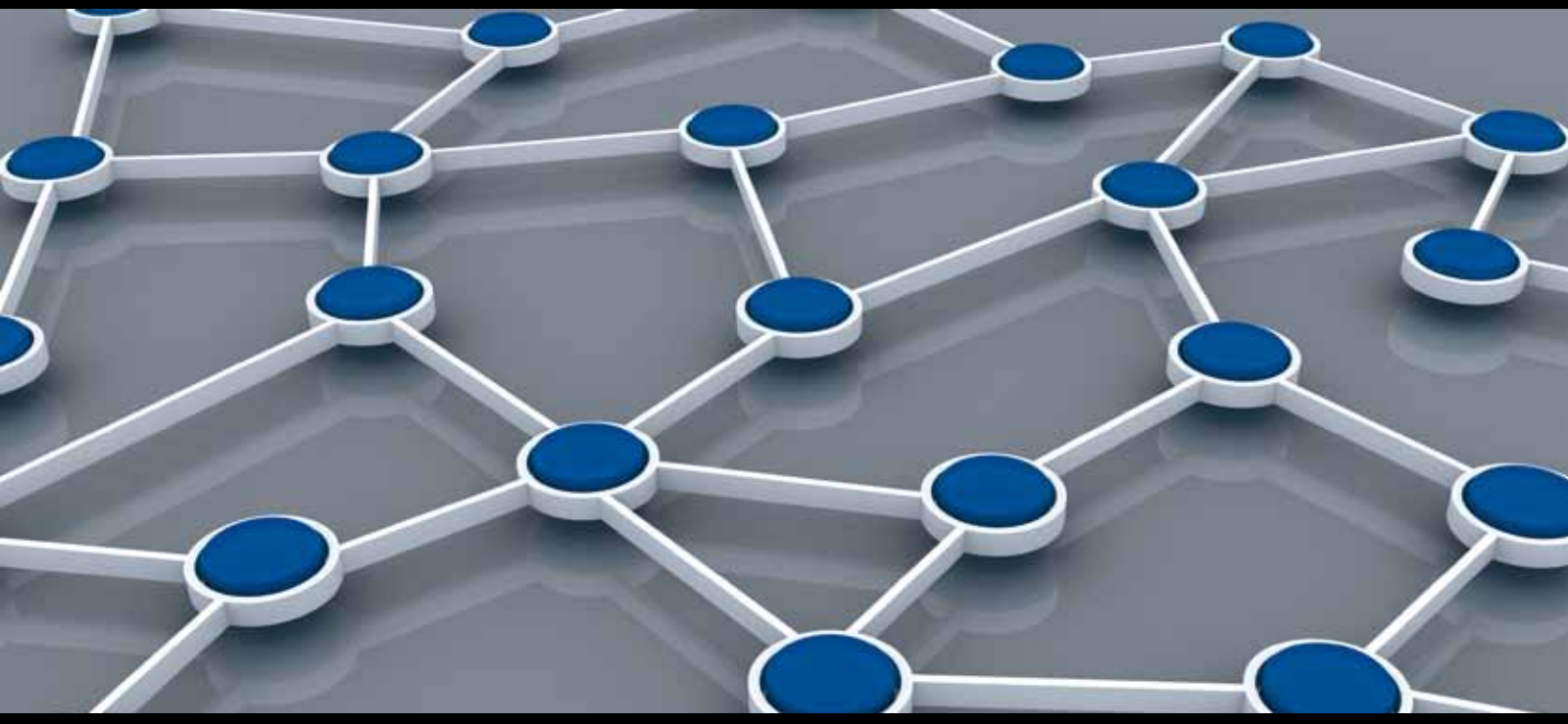
International Journal of Distributed Sensor Networks

# Algorithm and Theory for Robust Wireless Sensor Networks

Guest Editors: Chang Wu Yu, Qin Xin, Naveen Chilamkurti, and Shengming Jiang



# Algorithm and Theory for Robust Wireless Sensor Networks

International Journal of Distributed Sensor Networks

# Algorithm and Theory for Robust Wireless Sensor Networks

Guest Editors: Chang Wu Yu, Qin Xin, Naveen Chilamkurti, and Shengming Jiang

# Editorial Board

# Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 370512, 1 page
http://dx.doi.org/10.1155/2014/370512

*Editorial*

# Algorithm and Theory for Robust Wireless Sensor Networks

**Chang Wu Yu,[1] Qin Xin,[2] Naveen Chilamkurti,[3] and Shengming Jiang[4]**

[1] *Department of Computer Science and Information Engineering, Chung Hua University, Taiwan*
[2] *Faculty of Science and Technology, University of the Faroe Islands, Faroe Islands*
[3] *Department of Computer Science and Computer Engineering, La Trobe University, Victoria, Australia*
[4] *College of Information Engineering, Shanghai Maritime University, China*

Correspondence should be addressed to Chang Wu Yu; james.cwyu@gmail.com

Received 13 April 2014; Accepted 13 April 2014; Published 28 April 2014



Wireless sensor networks raise a number of interesting and undiscovered algorithmic issues, but traditional techniques are not sufficient to solve these problems in the right way. This is specifically due to constrained energy and computation capability, nondeterministic sensor failures, channel impairments, node mobility, hostile and distrusted environments, and even external attackers. In all these respects, wireless sensor networks exhibit substantial vulnerability when compared to other networks.

It is challenging to design a robust wireless sensor network by devising novel algorithms or developing new theories whilst introducing minimal communication overhead and energy consumption. For example, sensor nodes are often deployed in outdoor or hazardous environments. Power outage of sensors can lead to node failure and cause many serious problems. Recently, an aggressive approach has been developed that wirelessly recharges the sensor nodes to increase the robustness of the sensor networks. The charging tasks can be performed by single or multiple mobile machines. However, the algorithmic and theoretical issues in the wireless charging problem for WSNs were not fully explored.

The main focus of this special issue is devoted to a deeper understanding of the algorithms and theories which are developed to build up a robust wireless sensor network. Moreover, it is to push the theoretical and practical research forward for a deeper understanding in the fundamental algorithm, modeling, and analysis techniques of robust wireless sensor networks.

Totally, we have received 50 submissions coming from different countries all around the globe in response to call for paper. Each accepted article has been reviewed by at least three reviewers. In the end, 17 articles are revised and selected for publishing in this special issue. We believe that accepted papers provide a good balance of the application of algorithms and theories to different networking problems for robust wireless sensor networks.

## Acknowledgments

We would like to thank all authors who have provided their remarkable contributions to this special issue. We also appreciate the outstanding review work performed by the referees of this special issue for providing valuable comments to the authors. Without their full supports, this special issue would not be so successful.

*Chang Wu Yu*
*Qin Xin*
*Naveen Chilamkurti*
*Shengming Jiang*

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 393802, 9 pages
http://dx.doi.org/10.1155/2014/393802

*Research Article*

# Robust Estimation Fusion in Wireless Senor Networks with Outliers and Correlated Noises

**Yan Zhou,[1,2] Dongli Wang,[1] Tingrui Pei,[1] and Shujuan Tian[1]**

[1] *College of Information Engineering, Xiangtan University, Xiangtan 411105, China*
[2] *Key Laboratory of Intelligent Computing & Information Processing of MOE, Xiangtan University, Xiangtan 411105, China*

Correspondence should be addressed to Yan Zhou; yanzhou@xtu.edu.cn

Received 29 May 2013; Revised 25 October 2013; Accepted 28 October 2013; Published 2 April 2014

Academic Editor: Shengming Jiang



This paper addresses the problem of estimation fusion in a distributed wireless sensor network (WSN) under the following conditions: (i) sensor noises are contaminated by outliers or gross errors; (ii) process noise and sensor noises are correlated; (iii) cross-correlation among local estimates is unknown. First, to attack the correlation and outliers, a correlated robust Kalman filtering (co$R^2$KF) scheme with weighted matrices on innovation sequences is introduced as local estimator. It is shown that the proposed co$R^2$KF takes both conventional Kalman filter and robust Kalman filter as a special case. Then, a novel version of our internal ellipsoid approximation fusion (IEAF) is used in the fusion center to handle the unknown cross-correlation of local estimates. The explicit solution to both fusion estimate and corresponding covariance is given. Finally, to demonstrate robustness of the proposed co$R^2$KF and the effectiveness of IEAF strategy, a simulation example of tracking a target moving on noisy circular trajectories is included.

## 1. Introduction

Estimation fusion, or data fusion for estimation, has widespread applications in many practical situations that data from multiple sources are involved, for example, guidance, defense, robotics, integrated navigation, target tracking, and GPS positioning [1]. Combining the results of multiple sensors can provide more accurate estimation than using a single sensor [2]. There are two basic fusion architectures [1, 3]: centralized and distributed (referred to as measurement fusion and estimate fusion in target tracking, resp.), depending on whether raw data are sent to the fusion center or not. Both architectures have pros and cons in terms of optimality, channel requirements, requirements, reliability, information sharing, and so forth. For the distributed fusion, it has been realized for many years that local estimates (track) have correlated errors thereafter the original work of [3]. How to counter this cross-correlation has been a central topic in distributed fusion. One problem with the Kalman Filtering is that it requires either that the measurements are independent or that the cross-covariance is known [4, 5]. As is well known, the independent assumption can be relaxed in the case of correlated data, if the cross-covariance information is available. The optimal KF-based approach that the KF maintains cross-covariance information between updates is proposed considering the correlation among the local estimates [3, 6–8].

A common simplification is to assume the cross-covariance to be zero; that is, the measurements are independent. However, the simplification may cause that the KF produces nonconservative covariance. This leads to an artificially high confidence value, which can lead to filter divergence [9]. Although under the assumption that the cross-correlation is known, the optimal KF-based approach scales quadratically with the number of updates. This makes optimal KF-based approach impractical [10]. Several ingenious techniques, such as the tracklets technique [11], covariance intersection [12], and those based on the information graph [13], have been developed. Unfortunately, if the cross-correlation information is missing or incomplete, the Kalman filter cannot be applied. In such situations, to allow the use of the Kalman filter, the independence of the sources is

often assumed and the correlation is simply ignored in the fusion process, for example, the simple convex combination (SCC) method [14]. This makes the filter over optimistic in its estimation, which may lead to divergence [7]. Recently proposed covariance intersection (CI) filtering [8, 10] is based on convex combination of information matrices, that is, inverse covariance matrices and the corresponding information states. The algorithm provides a general framework for information fusion with incomplete knowledge about the signal sources since it yields consistent estimates for any degree of cross-correlation. Since covariance intersection filtering requires optimization of a nonlinear cost function and instead of underestimation of the actual covariance matrix, the covariance intersection method overestimates it, which obviously results in a significant decrease in performance. To avoid both the inconsistency of the basic convex combination and the lack of performance of the covariance intersection method, internal ellipsoid approximation fusion (IEAF) has been proposed [15]. For this approach, the largest volume ellipsoid within the intersection of two ellipsoids can be computed by internal ellipsoid approximation.

On the other hand, the distribution of noise arising in application deviates frequently from assumed Gaussian model, often being characterized by skewed (asymmetric) or heavier tails generating the outlier. A sufficiently far away located outlier can completely cause the least squares estimator or the Kalman filter to break down [16, 17]. This will degrade the fusion performance greatly. Therefore it is of practical interest to consider filters which are robust to perform fairly well in non-Gaussian environment especially in the presence of outliers, and some results have been obtained during the last decade. Robust statistical procedures provide formal methods to spot the outlying data points and reduce their influence. Most of the contributions in this area have been directed toward censoring data; namely, if an observation differs sufficiently from its predicted value, then it is discarded. For example, an $M$-estimate filter for robust adaptive filtering in impulse noise is proposed in [18]; a recursive adaptive algorithm and a robust threshold estimation method are derived employing an $M$-estimate cost function. Djurovic and Kovacevic established the equivalence between the Kalman filter algorithm and a particular least squares regression problem. Based on the equivalence it solved the robust estimation with unknown noise statistics with the help of $M$-estimate method, and the equivalence between the Kalman filter and proposed technique is established [19]. Recently, in [20] a deemphasis weighting approach is used to suppress the effect of outliers in background samples during the formation of a sample covariance matrix. When outliers are present when comparing the results from processing simulated and real coherent radar data using the proposed approach with results using no outlier suppression and censored sample matrix inversion pruning methods, the deemphasis techniques are shown to produce the most robust diction performance. Very recently, by reformulating the traditional Kalman filter into a least square form, a novel version of RKF has been proposed using $L_1$-regularized optimization [21]. To the best of the authors' knowledge, however, there is little result discussing how to eliminate or reduce the influence of outliers on the fusion performance, which remains a challenging problem so far.

In this paper, the problem of estimation fusion in a distributed architecture under the following conditions is addressed: (i) process noise and sensor noises are correlated; (ii) sensor noises are contaminated by outliers or gross errors; (iii) cross-correlation among local estimates is unknown. First, to attack the correlation and outliers, a novel robust Kalman filtering (co$R^2$KF) scheme with weighted matrices of innovation sequences is introduced as local estimator. It is shown that the proposed co$R^2$KF takes both conventional Kalman filter and robust Kalman filter as a special case. Then, a novel version of our internal ellipsoid approximation fusion (IEAF) is used in the fusion center to handle the unknown cross-correlation of local estimates.

## 2. Problem Statement and Lemmas

Consider the discrete linear stochastic system with $N$ sensors in the network:

$$x(t+1) = Fx(t) + G\omega(t), \tag{1}$$

$$y_i(t) = H_i x(t) + \upsilon_i(t) + z_i(t), \quad i = 1, 2, 3, \ldots, N, \tag{2}$$

where $x(t) \in \mathfrak{R}^n$ is the state vector, $y_i(t) \in \mathfrak{R}^{m_i}$ is the $i$th measurement in the sampling period $tT$; $\omega(t) \in \mathfrak{R}^p$ is the disturbance input or system noise with zero mean and variance matrix $Q$; $e_i(t) = [\upsilon_i(t) + z_i(t)] \in \mathfrak{R}^{m_i}$, $i = 1, 2, 3, \ldots, N$, are the outlier-contaminated measuring noise vectors. The matrices $F$, $G$, and $H_i$ are known real constant matrices with appropriate dimensions.

*Assumption 1.* $\omega(t)$ and $\upsilon_i(t)$, $i = 1, 2, 3, \ldots, N$, are correlated white noises with zero mean and

$$\begin{aligned} E\left(\omega_i(t)\upsilon_j^T(k)\right) &= S_{ij}(t)\delta_{tk}, \quad \forall t, k, \text{ if } i \neq j, \\ E\left\{\begin{bmatrix}\omega(t)\\ \upsilon_i(t)\end{bmatrix}\begin{bmatrix}\omega^T(k) & \upsilon_i^T(k)\end{bmatrix}\right\} &= \begin{bmatrix}Q(t) & S_i(t)\\ S_i^T(t) & R_i(t)\end{bmatrix}\delta_{tk}, \end{aligned} \tag{3}$$

where $E$ is the expectation, the superscript $T$ denotes the transpose, and $\delta_{tk}$ is the Kronecker delta function.

*Remark 2.* Our interest in this paper also lies in the outlier measurement which means mismatch in measurement noise model. In order to simulate these unmodeled measurement uncertainty, a non-Gaussian error term $z_i(t)$ is included. Commonly, the outlier-corrupted measuring noise can be represented by the Turkey's gross error model, a contaminated Gaussian model. Mathematically, $e_i(t) = \upsilon_i(t) + z_i(t)$ are with non-Gaussian density function $f(e)$ described by

$$\mathcal{F}_i = (1-\alpha)\mathcal{G}_i + \alpha\Delta\mathcal{G}_i, \tag{4}$$

where $\mathcal{G}_i$ is the zero-mean Gaussian density and $\Delta\mathcal{G}_i$ is some unknown symmetric function representing the impulsive part of the noise density or outliers.

*Remark 3.* The problem of outliers is of practical importance in a target tracking system using multiple sensors (radar

or infrared) [21, 22] and communication applications where non-Gaussian (heavy-tailed) noise occurs, such as in underwater acoustics, and satellite communications through the ionosphere [23]. It is commonly used to describe unmodeled measurement uncertainties that relate to the sensor failure, spikes, or jamming.

*Assumption 4.* The initial state $x(0)$ is independent of $\omega(t)$ and $e_i(t)$, $i = 1, 2, 3, \ldots, N$, and

$$Ex(0) = x_0, \qquad E\left[(x(0) - x_0)(x(0) - x_0)^T\right] = P_0. \quad (5)$$

The problem is to find the estimation fusion $\widehat{x}_0(t)$ of the state $x(t)$ in terms of local robust Kalman filter based on the measurement $(y_i(t), \ldots, y_i(1))$, $i = 1, 2, 3, \ldots, N$. The estimation fusion should have the desirable properties of efficiency and robustness; that is, it (i) yields the estimation fusion with a high accuracy for normal distributed observation while keeping the solution in high efficiency; (ii) reduces the bad effect of moderate errors on filtering and fusion in the way of weighting innovation; (iii) is robust in the sense that outliers and correlation do not affect the solution by setting the weighted matrix of innovation to be zeroes and further suppress the influence of outliers to the performance of fusion.

We start with some definitions and lemmas.

*Definition 5.* Ellipsoid $\varepsilon(x_0, P)$ in $R^n$ with center $x_0$ and shape matrix $P$ is the set

$$\varepsilon(x_0, P) = \left\{x \in R^n \mid (x - x_0)^T P^{-1} (x - x_0) \le 1\right\}, \quad (6)$$

where $P > 0$ might be standing for the covariance matrix of the estimation or fusion error.

**Lemma 6.** *Under the Assumptions 1 and 4, for ith sensor subsystem of the system* (1)-(2) *without outlier (i.e., $z_i(t) = 0$), the local optimal Kalman filters [24] one has:*

$$\widehat{x}_i(t+1 \mid t+1) = \widehat{x}_i(t+1 \mid t) + K_i(t+1)\,\varepsilon_i(t+1), \quad (7)$$

$$\widehat{x}_i(t+1 \mid t) = \Phi_i(t)\,\widehat{x}_i(t \mid t) + \Theta_i(t)\,y_i(t), \quad (8)$$

$$\varepsilon_i(t+1) = y_i(t+1) - H_i \widehat{x}_i(t+1 \mid t), \quad (9)$$

$$K_i(t+1) = P_i(t+1 \mid t) H_i^T \left[H_i P_i(t+1 \mid t) H_i^T + R_i\right]^{-1}, \quad (10)$$

$$P_i(t+1 \mid t) = \Phi_i(t) P_i(t \mid t) \Phi_i^T(t) + G\left[Q - S_i(t) R_i^{-1}(t) S_i^T(t)\right] G^T, \quad (11)$$

$$P_i(t+1 \mid t+1) = \left[I_n - K_i(t+1) H_i\right] P_i(t+1 \mid t), \quad (12)$$

$$\widehat{x}_i(0 \mid 0) = x_0, \qquad P_i(0 \mid 0) = P_0, \quad (13)$$

*where $\Phi_i(t) = F(t) - \Theta_i(t) H_i(t)$, $\Theta_i(t) = G(t) S_i(t) R_i^{-1}$, and $I_n$ stands for a n-dimensional unit diagonal matrix. $P_i(t+1 \mid t)$ and $P_i(t \mid t)$ are the one-step prediction and filtering error variance, respectively, and $K_i(t)$ is the filtering gain matrix. As described in Section 3, the innovation $\varepsilon_i(t)$ plays an important role in robust Kalman filter recursive process.*

*Remark 7.* If the process noise and measuring noises are not correlated, that is, $S_i(t) = 0$, it is obvious that $\Theta_i(t) = 0$, $\Phi_i(t) = F(t)$, and (11) reduced to $P_i(t+1 \mid t) = F(t) P_i(t \mid t) F^T(t) + GQG^T$. This makes the correlated Kalman filter recursions in Lemma 6 taking the traditional Kalman filter in case of uncorrelated noises as a special case.

**Lemma 8.** *Checking whether two ellipsoids $\varepsilon(x_{0_1}, P_1)$ and $\varepsilon(x_{0_2}, P_2)$ have nonempty intersection can be cast as to the following Quadratic Programming (QP) problem with quadratic constraints [25, 26]:*

$$\beta_1 = \min_{\langle x, P_2^{-1} x\rangle = 1} \left\langle x, P_1^{-1} x\right\rangle = \min_{x^T P_2^{-1} x = 1} x^T P_1^{-1} x,$$
$$\beta_2 = \min_{\langle x, P_1^{-1} x\rangle = 1} \left\langle x, P_2^{-1} x\right\rangle = \min_{x^T P_1^{-1} x = 1} x^T P_2^{-1} x, \quad (14)$$

*where $\beta_1$ and $\beta_2$ are invariant with respect to affine coordinate transformation and describe the position of ellipsoids $\varepsilon(x_{0_1}, P_1)$, $\varepsilon(x_{0_2}, P_2)$ with respect to each other:*

(i) *if $\beta_1 \ge 1$, $\beta_2 \ge 1$, then $\varepsilon(x_{0_1}, P_1) \cap \varepsilon(x_{0_2}, P_2) = \varphi$,*

(ii) *if $\beta_1 \ge 1$, $\beta_2 \le 1$, then $\varepsilon(x_{0_1}, P_1) \subseteq \varepsilon(x_{0_2}, P_2)$,*

(iii) *if $\beta_1 \le 1$, $\beta_2 \ge 1$, then $\varepsilon(x_{0_1}, P_1) \supseteq \varepsilon(x_{0_2}, P_2)$,*

(iv) *if $\beta_1 < 1$, $\beta_2 < 1$, then $\varepsilon(x_{0_1}, P_1) \cap \varepsilon(x_{0_2}, P_2) \ne \varphi$, and $\varepsilon(x_{0_1}, P_1) \not\subset \varepsilon(x_{0_2}, P_2)$, and $\varepsilon(x_{0_2}, P_2) \not\subset \varepsilon(x_{0_1}, P_1)$.*

## 3. An Improved Robust Kalman Filter: coR$^2$KF

From (7) to (13), we can see that in ($t$+1)th sampling period, $\widehat{x}_i(t+1 \mid t+1)$ is corrected by the linear combination of $\varepsilon_i(t+1)$. Therefore, if the measurements $y_i(t+1)$ are contaminated by outliers, $\varepsilon_i(t+1)$ will correct $\widehat{x}_i(t+1 \mid t+1)$ in a wrong way, which should make traditional Kalman filter degrade or even divergent.

In another point of view, the conventional Kalman filter can also be formulated as a solution to a particular weighted least squares problem [21]. Unfortunately, it is not robust because extreme outliers with arbitrarily large residuals can have an infinitely large influence on the resulting estimate. To handle this, the M-estimator, one of the most sophisticated approaches among the robust statistics approaches, is proposed [16]. Further, the M-estimator has an advantage of less computational effort as it can be computed by a standard least squares algorithm with minor modifications [27].

M-estimators attempt to suppress the influence of outliers by replacing the square of the residuals with a less rapidly increasing loss function, which is

$$J = \sum_{j=1}^{m_i} \rho\left(y_{ij}(t) - h_{ij} x(t)\right) = \sum_{j=1}^{m_i} \rho\left(\varepsilon_{ij}(t)\right), \quad (15)$$

where $y_{ij}(t)$, $\varepsilon_{ij}(t)$, and $h_{ij}$ stand for the $j$th row of $y_i(t)$, $\varepsilon_i(t)$, and $H_i$, respectively (cf. (9)). $\rho(\cdot)$ is a scalar robust

convex function that has to cut off the outliers. Particularly, if one chooses $\rho(\cdot)$ to be a quadratic function, the estimator according to (15) reduces to the least squares estimator or Kalman filter solution (7) [21, 27].

Equating the first partial derivatives with respect to the state to be estimated $x(t)$ leads to the following relation:

$$\sum_{j=1}^{m_i} \psi\left(y_{ij}(t) - h_{ij}\hat{x}(t)\right) h_{ij} = \sum_{j=1}^{m_i} \psi\left(\varepsilon_{ij}(t)\right) h_{ij} = 0. \tag{16}$$

The score function $\rho(\cdot)$ is usually nonnegative and symmetric, and $\psi(\cdot)$, the derivative of $\rho(\cdot)$, is often called the influence (score) function, since it describes the influence of the measurement errors on the solutions.

Now, (16) can be rewritten as

$$\sum_{j=1}^{m_i} h_{ij}\varepsilon_{ij} \frac{\psi\left(\varepsilon_{ij}\right)}{\varepsilon_{ij}} = 0. \tag{17}$$

Letting $d(\varepsilon_{ij}) = \psi(\varepsilon_{ij})/\varepsilon_{ij}$, then (17) can be reformulated as the matrix form

$$H_i^T D_i(\varepsilon)\,\varepsilon_i = 0, \tag{18}$$

where $D_i(\varepsilon) = \mathrm{diag}\{d(\varepsilon_{i1}), d(\varepsilon_{i2}), \ldots, d(\varepsilon_{in})\}$.

In the light of the above comparison and analysis of conventional Kalman filtering and M-estimator, the proposed co$R^2$KF is given in Theorem 9 as follows.

**Theorem 9.** *Under Assumptions 1 and 4, the ith sensor subsystem of the system* (1)-(2)*, the co$R^2$KF one has:*

$$\hat{x}_i(t+1 \mid t+1) = \hat{x}_i(t+1 \mid t) + K_i(t+1)\,D_i(t)\,\varepsilon_i(t+1), \tag{19}$$

$$K_i(t+1) = P_i(t+1 \mid t)\,H_i^T \times \left[H_i P_i(t+1 \mid t)\,H_i^T + D_i(t)\,R_i D_i^T(t)\right]^{-1}, \tag{20}$$

$$P_i(t+1 \mid t+1) = \left[I_n - K_i(t+1)\,D_i(t)\,H_i\right] \times P_i(t+1 \mid t) \cdot \left[I_n - K_i(t+1)\,D_i(t)\,H_i\right]^T + K_i(t+1)\,D_i(t)\,R_i D_i^T K_i^T(t+1). \tag{21}$$

*Other recursive steps are just the same as* (8), (9), (11)*, and* (13) *in Lemma 6.*

*Proof.* The formula (19) can be derived from above directly, and the covariance of weighted innovation is

$$E\left[\left(D_i(t)\,\varepsilon_i(t+1)\right)\left(D_i(t)\,\varepsilon_i(t+1)\right)^T\right] = D_i(t)\,R_i D_i^T(t) \tag{22}$$

from which we have the robust Kalman gain matrix as (20).

Substituting (19) into the filtering error equation

$$\begin{aligned}\tilde{x}_i(t+1 \mid t+1) &= x_i(t+1) - \hat{x}_i(t+1 \mid t+1) \\ &= \left[I_n - K_i(t+1)\,D_i(t)\,H_i\right]\tilde{x}_i(t+1 \mid t) \\ &\quad - K_i(t+1)\,D_i(t)\,v_i(t),\end{aligned} \tag{23}$$

where $\tilde{x}_i(t+1 \mid t)$ is the one-step prediction residual, and after mathematical manipulation, the robust filter covariance can be computed as

$$\begin{aligned}&P_i(t+1 \mid t+1) \\ &= E\left[\tilde{x}_i(t+1 \mid t+1)\,\tilde{x}_i^T(t+1 \mid t+1)\right] \\ &= \left[I_n - K_i(t+1)\,D_i(t)\,H_i\right]P_i(t+1 \mid t) \\ &\quad \times \left[I_n - K_i(t+1)\,D_i(t)\,H_i\right]^T \\ &\quad + K_i(t+1)\,D_i(t)\,R_i D_i^T K_i^T(t+1).\end{aligned} \tag{24}$$

This completes the proof. □

*Remark 10.* $\rho(\cdot)$ is a robust $M$-estimate function for suppressing the outliers and is important for the estimation performance. Different $\rho(\cdot)$ will result in different $M$-estimate and fusion performance. Say, for a given density $f$, the choice $\rho(v) = -\log f(v)$ yields the maximum likelihood estimator. Several robust cost functions have been used in the robust statistics setting, such as Huber's robust cost function, Andrews' method, Vapnik's loss function, or the biweight approach. Here, we propose a more general $M$-estimate function generated from Huber's robust cost function:

$$\rho\left(\varepsilon_{ij}(t)\right) = \begin{cases} \dfrac{\varepsilon_{ij}^2(t)}{2}, & \text{for } \left|\varepsilon_{ij}(t)\right| \le a, \\ a\left|\varepsilon_{ij}(t)\right| - \dfrac{a^2}{2}, & \text{for } a < \left|\varepsilon_{ij}(t)\right| \le b, \\ ab - \dfrac{a^2}{2}, & \text{for } \left|\varepsilon_{ij}(t)\right| > b, \end{cases} \tag{25}$$

where $a$ and $b$ have to be chosen to provide the desired efficiency at the Gaussian model while possess robustness at the non-Gaussian model. Usually, they are chose empirically [18, 22]. For simplicity, we let $a = 3\sqrt{R_i^{j,j}(t)}$, $b = 5\sqrt{R_i^{j,j}(t)}$ in this paper, where $R_i^{j,j}(t)$ stands for the $(j, j)$ entry of the covariance matrix $R_i(t)$.

*Remark 11.* It can be seen that $\rho(\cdot)$ is an even real-valued function and it is quadratic when $\varepsilon_{ij}(t)$ is smaller than $a$, which is just the same as the maximum likelihood (ML) cost function and keeps the efficiency of the $M$-estimate. For larger values of $\varepsilon_{ij}(t)$ in the interval $[a, b]$, the function is linear and increase more slowly than ML. For residuals

greater than $b$, the function is equal to a constant. Based on (25), the weighted matrix of innovations can be formulated as

$$d\left(\varepsilon_{ij}(t)\right) = \begin{cases} 1, & \text{for } \left|\varepsilon_{ij}(t)\right| \le a, \\ \dfrac{a}{\left|\varepsilon_{ij}(t)\right|} & \text{for } a < \left|\varepsilon_{ij}(t)\right| \le b, \\ 0, & \text{for } \left|\varepsilon_{ij}(t)\right| > b. \end{cases} \tag{26}$$

The three different intervals of $D_i(\cdot)$ serve to deal with different kinds of residuals. In order to keep the accuracy and efficiency, when $|\varepsilon_{ij}(t)| \le a$, $D_i(\cdot)$ are set to be 1; when sampling from the moderate innovations, $D_i(\cdot)$ are decreased with the residuals while sampling from a heavy-tailed distribution or outliers, the weighted matrix is set to be zeroes.

*Remark 12.* If $\omega(t)$ and $v_i(t)$, $i = 1, 2, 3, \ldots, N$, are correlated white Gaussian noises with variance matrices $Q$ and $R_i$, respectively, we can see that $D_i = I_n$ from Remark 11. In this case, the proposed co$R^2$KF reduced to the correlated Kalman filter formulated in Lemma 6, which in turn takes the traditional Kalman filter with uncorrelated noises as a special case.

## 4. Robust Internal Ellipsoid Approximation Fusion

Once the local estimation is obtained by the subsystems, we are facing the problem of how to fuse the estimation in a right way in the higher level, that is, a cluster head or a sink node.

In this paper, our internal ellipsoid approximation fusion (IEAF) method [15] is adopted to fuse local estimate. For convenience, the IEAF is reformulated in the following theorem.

**Theorem 13.** *Given two ellipsoids $\varepsilon(x_{0_1}, P_1)$ and $\varepsilon(x_{0_2}, P_2)$, and define parameterized family of internal ellipsoids $\varepsilon(x_0^-, P^-)$ with*

$$x_0^- = \left(\omega_1 P_1^{-1} + \omega_2 P_2^{-1}\right)^{-1}\left(\omega_1 P_1^{-1} x_1 + \omega_2 P_2^{-1} x_2\right), \tag{27}$$

$$\begin{aligned} P^- = \Big(1 &- \omega_1 x_1^T P_1^{-1} x_1 - \omega_2 x_2^T P_2^{-1} x_2 \\ &+ x_0^{-T}\left(P^-\right)^{-1} x_0^-\Big) \cdot \left(\omega_1 P_1^{-1} + \omega_2 P_2^{-1}\right)^{-1}. \end{aligned} \tag{28}$$

*The best internal ellipsoid $\varepsilon(\widehat{x}_0^-, \widehat{P}^-)$ in class* (27)-(28)*, namely, such that*

$$\varepsilon\left(x_0^-, P^-\right) \subseteq \varepsilon\left(\widehat{x}_0^-, \widehat{P}^-\right) \subseteq \varepsilon\left(x_{0_1}, P_1\right) \cap \varepsilon\left(x_{0_2}, P_2\right), \tag{29}$$

*for all $0 \le \omega_1, \omega_2 \le 1$, is specified by the parameters*

$$\begin{aligned} \widehat{\omega}_1 &= \frac{1 - \min\left(1, \beta_2\right)}{1 - \min\left(1, \beta_1\right) \cdot \min\left(1, \beta_2\right)}, \\ \widehat{\omega}_2 &= \frac{1 - \min\left(1, \beta_1\right)}{1 - \min\left(1, \beta_1\right) \cdot \min\left(1, \beta_2\right)}, \end{aligned} \tag{30}$$

*where $\beta_1$ and $\beta_2$ are the parameters determined in* (14).

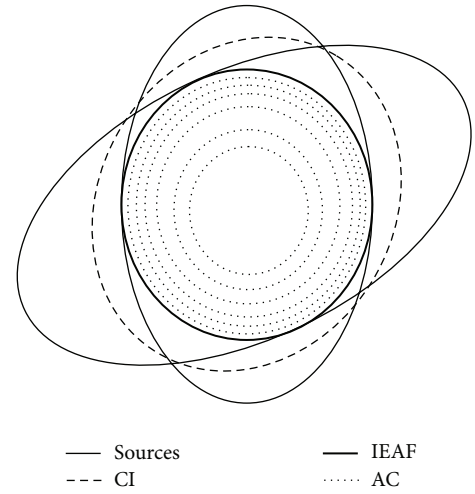



Figure 1: Comparison among the proposed IEAF and CI, and actual covariance (AC).

*Proof.* The proof is omitted. Readers are referred to [15] for a strict proof. □

*Remark 14.* After the parameters $\beta_1$ and $\beta_2$ are determined by Lemma 8, the center and shape parameters of the best internal ellipsoid $\varepsilon(\widehat{x}_0^-, \widehat{P}^-)$ can be calculated by (27)-(28) and (30).

*Remark 15.* A graphic demonstration of the relation between the correlated source ellipsoids and the fused one by IEAF and CI are shown in Figure 1. It can be seen that the actual covariance ellipsoid always lies within the region defined by the intersection of the covariance ellipsoids of the fused sources regardless the degree of correlation between these sources. The CI, on the contrary, results in a covariance matrix that will always be greater than the actual one. This makes the CI some degree of conservativeness and a suboptimal estimate. The proposed IEAF slightly underestimates the intersection region, instead of overestimating this region like CI. In fact, the IEAF computes the largest ellipsoid contained within the intersection region and results in an increased performance. Also the consistency of the IEAF can be observed from Figure 1 directly.

Note that the center of internal approximation ellipsoid can be easily calculated from (27). Hence, the only difficulty remained is to solve (28) for the shape matrix. In this paper, we derive the explicit solution to (28) as explained in the following theorem in light of the symmetric positive-define property of $P^-$, $P_1$, and $P_2$.

**Theorem 16.** *Given two local estimations $x_{0_1}$ and $x_{0_2}$ with the error covariance matrices $P_1$ and $P_2$, respectively, according to the internal ellipsoid approximation method, the fusion estimation and its covariance are*

$$\widehat{x}_0 = \left(\omega_1 P_1^{-1} + \omega_2 P_2^{-1}\right)^{-1}\left(\omega_1 P_1^{-1} x_1 + \omega_2 P_2^{-1} x_2\right), \tag{31}$$

$$P_0 = \widehat{x}_0\left(\omega_1 P_1^{-1} + \omega_2 P_2^{-1}\right)^{-1} Y^+, \tag{32}$$

*where*

$$\begin{aligned} Y = \text{ unvec } & \left\{\left(-\widehat{x}_0^T \otimes \widehat{x}_0^T\right)^+ \right. \\ & \times \text{vec}\left[\left(1-\omega_1 x_1^T P_1^{-1} x_1 - \omega_2 x_2^T P_2^{-1} x_2\right)\right. \\ & \cdot \left(\omega_1 P_1^{-1} + \omega_2 P_2^{-1}\right)^{-1} \\ & - \left(\omega_1 P_1^{-1} + \omega_2 P_2^{-1}\right)^{-T} \\ & \left.\left. \cdot \left(1-\omega_1 x_1^T P_1^{-1} x_1 - \omega_2 x_2^T P_2^{-1} x_2\right)\right]\right\} \end{aligned} \tag{33}$$

*and* $0 \le \omega_1, \omega_2 \le 1$ *can be calculated using Lagrange multiplier method from Lemma 8 and* (30).

*Proof.* Note that all the parameters in (28) are given except $P^-$. Letting

$$\begin{aligned} A &= 1 - \omega_1 x_1^T P_1^{-1} x_1 - \omega_2 x_2^T P_2^{-1} x_2, \\ B &= x_0^-, \qquad C = \left(\omega_1 P_1^{-1} + \omega_2 P_2^{-1}\right)^{-1}, \end{aligned} \tag{34}$$

and $X = P^-$, now we are in the position to calculated $X$ from the following equation:

$$X = \left(A + B^T X^{-1} B\right) C. \tag{35}$$

Noting that $C$ and $X$ are given symmetric positive-define matrices, we have

$$0 = X - X^T = \left(AC + B^T X^{-1} BC\right) - \left(C^T B^T X^{-T} B + C^T A^T\right); \tag{36}$$

that is, $AC - C^T A^T = (-B^T) X^{-1} BC + C^T B^T X^{-T} B$.

Letting $Y = X^{-1} BC$ and using the properties of Kronecker product, we have

$$\begin{aligned} \text{vec}\left(AC - C^T A^T\right) &= \text{vec}\left(-B^T Y\right) + \text{vec}\left(Y^T B\right) \\ &= \left(-I \otimes B^T\right) \text{vec}\,(Y) + \left(B^T \otimes I\right) \text{vec}\,(Y) \\ &= \left(-B^T \otimes B^T\right) \text{vec}\,(Y), \end{aligned} \tag{37}$$

where vec(·) represents vectorization function which is column stacking of the matrix.

Then $Y$ and $X$ can be derived as follows, respectively, as the same of (35)–(37):

$$\begin{aligned} X &= BC\left(\text{unvec}\left(\left(-B^T \otimes B^T\right)^+ \text{vec}\left(AC - C^T A^T\right)\right)\right)^+, \\ Y &= \text{unvec}\left(\left(-B^T \otimes B^T\right)^+ \text{vec}\left(AC - C^T A^T\right)\right), \end{aligned} \tag{38}$$

where unvec(·) transfers an vector to corresponding matrix. □

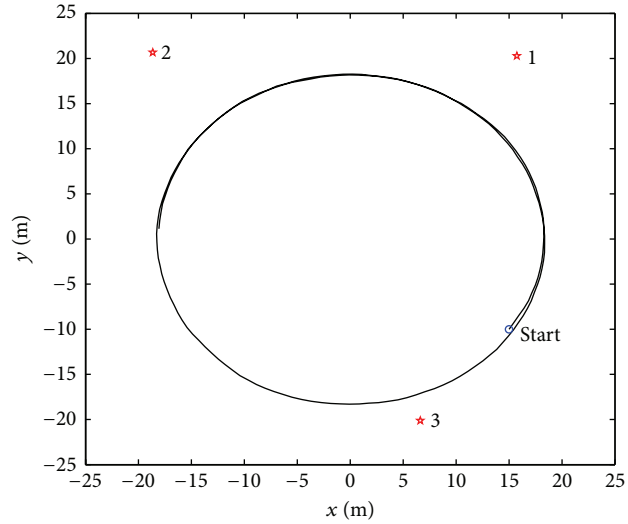



Figure 2: A particular setup of the simulation scenarios.

**Corollary 17.** (i) *If* $\beta_1 \ge 1$, $\beta_2 \le 1$, *then* $\widehat{\omega}_1 = 1$, $\widehat{\omega}_2 = 0$; *one has* $\varepsilon(\widehat{x}_0^-, \widehat{P}^-) = \varepsilon(x_{0_1}, P_1)$;

(ii) *if* $\beta_1 \le 1$, $\beta_2 \ge 1$, *then* $\widehat{\omega}_1 = 0$, $\widehat{\omega}_2 = 1$; *one has* $\varepsilon(\widehat{x}_0^-, \widehat{P}^-) = \varepsilon(x_{0_2}, P_2)$.

*Proof.* Here, we just prove (i) while the statement (ii) can be easily derived in a similar way. If $\beta_1 \ge 1$, $\beta_2 \le 1$, then from (30) we can obtain $\widehat{\omega}_1 = 1$, $\widehat{\omega}_2 = 0$. Furthermore, from Lemma 8 (ii), we have $\varepsilon(x_{0_1}, P_1) \subseteq \varepsilon(x_{0_2}, P_2)$, which means the approximated largest ellipsoid is $\varepsilon(x_{0_1}, P_1)$. In other words, we have the fused ellipsoid $\varepsilon(\widehat{x}_0^-, \widehat{P}^-) = \varepsilon(x_{0_1}, P_1)$. □

## 5. A Simulation Example

To evaluate the robustness of the proposed coR$^2$KF and effectiveness of the IEAF approach, a simulation example of tracking a target moving on noisy circular trajectories is given in this section. The objectives of the simulation examples are twofold: (a) to verify the robustness of the proposed coR$^2$KF and (b) to demonstrate the performance superiority of the IEAF method.

Suppose $N$ sensors are randomly deployed in the ROI, which is 50 m × 50 m with the coordinate from (−25, −25) to (25, 25). The layout of the simulation scenarios is illustrated in Figure 2, where a five-point star stands for the location of a sensor.

Consider a target with the following dynamics:

$$\dot{x} = F_c x + G_c \omega, \tag{39}$$

where $x = [p_x, p_y]^T$ denotes the state variable with $p_x, p_y$ stands for the target position on $x$- and $y$-direction, respectively, $F_c = \begin{bmatrix} 0 & -2; & 2 & 0 \end{bmatrix}$, and $G_c = [1, 1]^T$. We use the discrete-time model of the target with parameters:

$$F = I_2 + TF_c + \frac{T^2}{2} F_c^2 + \frac{T^3}{6} F_c^3, \qquad G = TG_c. \tag{40}$$

Table 1: RMSE ($/10^{-2}$ m) performance for single sensor.

| | | coR$^2$ KF | Traditional correlated KF |
|---|---|---|---|
| $\alpha = 0$ (no outliers) | / | 2.27 | 2.25 |
| $\alpha = 0.05$ | $k = 100$ | 2.29 | 2.64 |
| | $k = 500$ | 2.31 | 4.61 |
| | $k = 1000$ | 2.33 | 4.69 |
| $\alpha = 0.1$ | $k = 100$ | 2.46 | 4.01 |
| | $k = 500$ | 2.56 | 4.56 |
| | $k = 1000$ | 2.69 | 4.99 |
| $\alpha = 0.2$ | $k = 100$ | 2.66 | 4.88 |
| | $k = 500$ | 2.72 | 5.20 |
| | $k = 1000$ | 2.97 | 6.04 |

The step-size is $T = 0.025$. Each sensor makes noisy measurements of the position of the target according to (2) with $H = I_2$. We add outliers into $v_i(t)$, $i = 1, 2$, all the simulation period. The outlier-corrupted noise $e_i(t)$ is with the contaminated Gaussian density function $\mathcal{F}_i = (1 - \alpha)N(0, \sigma_i^2) + \alpha N(0, k\sigma_i^2)$, where $\sigma_i$ is set as 5. Moreover, $v_i(t)$ are correlated with the process noise $\omega(t)$, which is standard Gaussian white noise. The correlation coefficient $S_i(t)$ is supposed to be $[2,\ 2]$, $[2,\ 1]$, and $[1,\ 2]$, respectively. Our goal is to find the estimation fusion based on the local robust Kalman filters suppressing the effect of outliers on the estimation performance in case of cross-correlation among local estimates is unknown. We use the Root of Mean Square Error (RMSE), that is, RMSE $= \sqrt{\sum_{i=1}^{M} [\tilde{x}_k(i)^T \tilde{x}_k(i)]/M}$, as a performance criterion, where $M$ is the number of Monte Carlo runs and $\tilde{x}_k(i)$ represents the estimation error.

*Scenario 1* (one sensor case, Robustness of coR$^2$KF). To verify the robustness of proposed coR$^2$KF under condition of both outlier-corrupted measuring noises and process-measuring noises correlation, we first consider that just a single sensor is used to track the target. For the coR$^2$KF, we set $a = 3\sigma_i$ and $b = 5\sigma_i$ in (26). The RMSE over 500 Monte Carlo runs with different PDF function is shown in Table 1.

From Table 1, we can see in case of no outlier presented that the proposed coR$^2$KF performs a little poorer than traditional KF. This is because coR$^2$KF has deweighted the elements of matrix $D$ in case of larger innovations. However, when outliers present, the adaptive weight of matrix $D$ according to different innovation make the proposed coR$^2$KF more robust than traditional KF. For example, in case of $\alpha = 0.2$ and $k = 1000$, the degradation of coR$^2$KF and traditional KF are, respectively, $(2.97 - 2.27)/2.27 = 30.84\%$ and $(6.04-2.25)/2.25 = 168.44\%$. Therefore, the performance discrepancy of coR$^2$KF with different $\alpha$ and $k$ is not very large, which demonstrates the robustness of proposed coR$^2$KF.

*Scenario 2* (multiple sensors case, advantage of IEAF). In order to demonstrate the performance superiority of the IEAF method, we then consider multiple-sensor tracking of the target in a distributed way. Suppose that there are 3 sensors randomly deployed in the ROI as shown in Figure 2, while other parameters are the same as in Scenario 1.

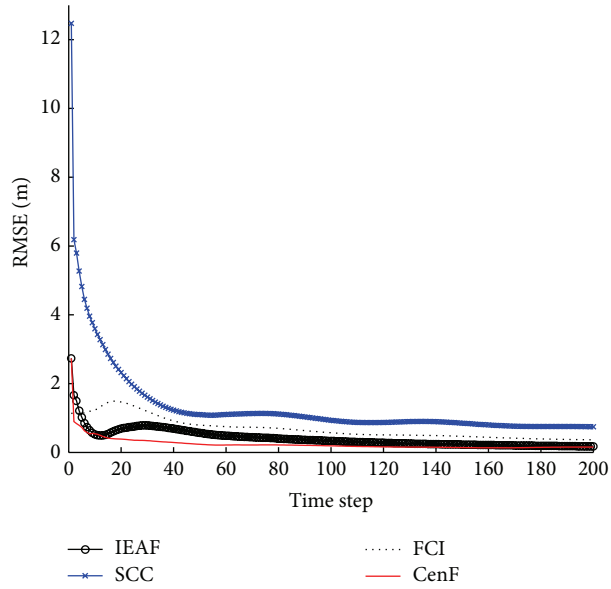



Figure 3: Comparison of RMSE among SCC, FCI, IEAF, and the centralized fusion.

In fact, the performance of IEAF has been shown to be less conservative than CI; both are not necessarily to have the knowledge of cross-correlation [15]. Also, this can be observed from the results of this scenario, where the proposed coR$^2$KF is adopted as the local estimator. The simulation is performed over 500 Monte Carlo runs each with 200 time steps. In this scenario, only the parameters $\alpha = 0.2$ and $k = 1000$ are considered. In Table 2, the performance of the fast covariance intersection (FCI) [28], the simple convex combination (SCC), and the proposed IEAF are compared with the centralized fusion, which is optimal by directly fusing the measurements from local systems.

Obviously, using coR$^2$KF as local estimators, proposed IEAF and the FCI achieve almost identical performance; both perform very closely to the optimal centralized fusion by coR$^2$KF. Besides, both IEAF and FCI are much superior to the SCC approach, which ignore the cross-correlation among local estimates. To demonstrate the results visually and clearly, the RMSE of four fusion methods mentioned above versus time sampling periods is shown in Figure 3. Moreover, the real track along $x$-direction, measurement of each sensor and fused estimate by coR$^2$KF-IEAF are shown in Figure 4. The results along $y$-axis are similar, which are omitted for space reason. From Figure 4, we can see that the measurement of each sensor fluctuates seriously as the outliers appear. However, the outliers are suppressed by the coR$^2$KF, as can be seen from Table 2 and both figures. In Table 2, we also give the results on average computational complexity over 500 Monte Carlo runs for whole 200 simulation steps. It is obvious that SCC has the smallest complexity since no cross-correlation needed. Our proposed IEAF has almost identical computational burden as FCI approach. This

Table 2: Comparison of fusion accuracy and average computational burden for 200 steps.

| Fusion method | | RMSE ($/10^{-2}$ m) | Computational burden (ms) |
|---|---|---|---|
| Distributed fusion | SCC | 5.73 | 31.2 |
| | FCI | 2.56 | 49.8 |
| | IEAF | 2.48 | 52.9 |
| Centralized fusion | | 2.32 | / |

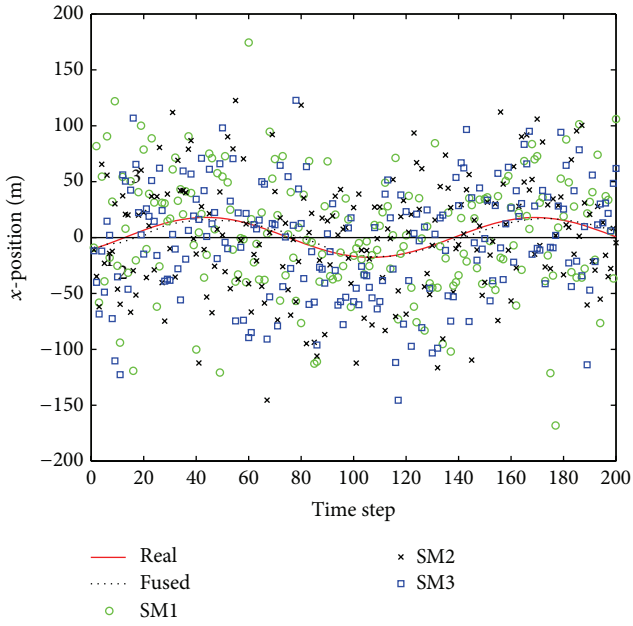



Figure 4: The real track, each measurement, and the fused estimate along $x$-axis. The measurements of the 3 sensors are denoted by SM1, SM2, and SM3, respectively, while the fused position estimate by the black dotted line, which tracks the real position on $x$-axis very closely (cf. the red solid line).

can be expected since there is no extra computation except $Y$ in Theorem 16.

## 6. Conclusion and Future Work

The problem of estimation fusion in a distributed WSN has been addressed under the conditions of correlation between process noise and sensor noises, outlier-corrupted sensor noises, and unknown cross-correlation among local estimates. To attack the correlation and outliers, a novel robust Kalman filtering (co$R^2$KF) scheme with weighted matrices of innovation has been introduced as local estimator. It has been shown that the proposed co$R^2$KF takes both conventional Kalman filter and robust Kalman filter as a special case. Then, an improved version of our previously proposed IEAF has been used in the fusion center to handle the unknown cross-correlation of local estimates. To demonstrate the robustness of proposed co$R^2$KF and the effectiveness of IEAF strategy, a simulation example of tracking a target moving on noisy circular trajectories has been included. Future work will be focused on posterior Cramer-Rao lower bounds (pCRLBs) analysis [29] and comparison with other outlier processing method (e.g., sensor validation technique) in the framework of estimation fusion [6].

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The work was supported by the National Natural Science Foundation of China (Under Grant nos. 61104210, 61100140, 61070180, and 61372049), Project of Science & Technology Department of Hunan (Under Grant no. 2012TP4032-6), and the Construct Program of the Key Discipline in Hunan Province.

## References

[1] Y. Wang and X. R. Li, "Distributed estimation fusion under unknown cross-correlation: an analytic center approach," in *Proceedings of the 13th International Conference on Information Fusion (FUSION '10)*, pp. 1–8, Edinburgh, UK, July 2010.

[2] D. Smith and S. Singh, "Approaches to multisensor data fusion in target tracking: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 12, pp. 1696–1710, 2006.

[3] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, 1981.

[4] D. P. Atherton, J. A. Bather, and A. J. Briggs, "Data fusion for several Kalman filters tracking a single target," *IEE Proceedings: Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 372–376, 2005.

[5] Y. Zhou, J. Li, and D. Wang, "Quantized measurement fusion in wireless sensor networks with correlated sensor noises," in *Proceedings of the 7th IEEE International Conference on Control and Automation (ICCA '09)*, pp. 1868–1873, Christchurch, New Zealand, December 2009.

[6] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: a review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[7] S.-L. Sun and Z.-L. Deng, "Multi-sensor optimal information fusion Kalman filter," *Automatica*, vol. 40, no. 6, pp. 1017–1023, 2004.

[8] S. Mori, K. C. Chang, and C. Y. Chong, "Comparison of track fusion rules and track association metrics," in *Proceedings of*

*the 15th International Conference on Information Fusion (FUSION '12)*, pp. 1996–2003, Singapore, July 2012.

[9] P. Maybeck, *Stochastic Models, Estimation and Control*, Academic Press, London, UK, 1979.

[10] J. K. Uhlmann, S. J. Julier, and H. F. Durrant-Whyte, "A culminating advance in the theory and practice of data fusion, filtering and decentralized estimation," Tech. Rep., Covariance Intersection Working Group, 1997.

[11] O. E. Drummond, "A hybrid sensor fusion algorithm architecture and tracklets," in *Signal and Data processing of Small Targets*, vol. 3163 of *Proceedings of SPIE*, pp. 512–524, San Diego, Calif, USA, July 1997.

[12] S. J. Julier and J. K. Uhlmann, "A non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the American Control Conference*, vol. 4, pp. 2369–2373, Albuquerque, NM, USA, June 1997.

[13] C.-Y. Chong, S. Mori, W. H. Barker, and K.-C. Chang, "Architectures and algorithms for track association and fusion," *IEEE Aerospace and Electronic Systems Magazine*, vol. 15, no. 1, pp. 5–13, 2000.

[14] C. Y. Chong and S. Mori, "Convex combination and covariance intersection algorithms in distributed fusion," in *Proceedings of the 4th International Conference on Information Fusion (FUSION '01)*, vol. 1, pp. 2–11, Montreal, Canada, 2001.

[15] Y. Zhou and J. Li, "Data fusion of unknown correlations using internal ellipsoidal approximation," in *Proceedings of the IFAC World Congress*, pp. 2856–2860, Seoul, Republic of Korea, July 2008.

[16] P. J. Huber, *Robust Statistics*, John Wiley & Sons, New York, NY, USA, 1981.

[17] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 1–20, 2013.

[18] Y. Zou, S. C. Chan, and T. S. Ng, "Robust M-estimate adaptive filtering," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 148, no. 4, pp. 289–294, 2001.

[19] Z. Djurovic and B. Kovacevic, "Robust estimation with unknown noise statistics," *IEEE Transactions on Automatic Control*, vol. 44, no. 6, pp. 1292–1296, 1999.

[20] M. McDonald and B. Bhashyam, "Outlier suppression in adaptive filtering through de-empahsis weighting," *IET Radar, Sonar & Navigation*, vol. 1, no. 1, pp. 38–49, 2007.

[21] J. Mattingely and S. Boyd, "Real-time convex optimization in signal processing," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 50–61, 2010.

[22] R. Kumar, "A new algorithm for outlier rejection in particle filters," in *Proceedings of the 13th International Conference on Information Fusion (FUSION '10)*, pp. 1–7, Edinburgh, UK, July 2010.

[23] X. Wang and H. V. Poor, "Robust multiuser detection in non-gaussian channels," *IEEE Transactions on Signal Processing*, vol. 47, no. 2, pp. 289–305, 1999.

[24] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Prentice-Hall, Boston, Mass, USA, 1979.

[25] A. Y. Vazhentsev, "On internal ellipsoidal approximations for problems of control synthesis with bounded coordinates," *International Journal of Computer and System Sciences*, vol. 39, no. 3, pp. 399–406, 2000.

[26] A. B. Kurzhanski and I. Vályi, "Ellipsoidal techniques for dynamic systems: the problem of control synthesis," *Dynamics and Control*, vol. 1, no. 4, pp. 357–378, 1991.

[27] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*, Wiley-Interscience, New York, NY, USA, 1986.

[28] W. Niehsen, "Information fusion based on fast covariance intersection filtering," in *Proceedings of the 5th International Conference on Information Fusion (FUSION '02)*, vol. 2, pp. 901–904, Annapolis, Md, USA, July 2002.

[29] Y. Zhou, J. Li, and D. Wang, "Posterior Cramér-Rao lower bounds for target tracking in sensor networks with quantized range-only measurements," *IEEE Signal Processing Letters*, vol. 17, no. 2, pp. 157–160, 2010.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 489563, 8 pages
http://dx.doi.org/10.1155/2014/489563

*Research Article*

# The Hybrid Taguchi-Genetic Algorithm for Mobile Location

**Chien-Sheng Chen,[1] Jium-Ming Lin,[2] Chin-Tan Lee,[3] and Chyuan-Der Lu[4]**

[1] *Department of Information Management, Tainan University of Technology, Tainan 71002, Taiwan*
[2] *Department of Communication Engineering, Chung Hua University, Hsinchu 30012, Taiwan*
[3] *Department of Electronic Engineering, National Quemoy University, Kinmen 89250, Taiwan*
[4] *Department of Finance, Tainan University of Technology, Tainan 71002, Taiwan*

Correspondence should be addressed to Chien-Sheng Chen; t00243@mail.tut.edu.tw

Received 29 June 2013; Accepted 25 December 2013; Published 24 March 2014

Academic Editor: Chang Wu Yu



To estimate the mobile location is an important topic in wireless communication. It is well known that non-line-of-sight (NLOS) problem is the most pivotal part that causes the estimated error. When we transmit the signal from mobile station (MS) to base stations (BSs), the direct path between MS and BS is sealed off by some obstacles, and the signal measurements will measure the error due to the signal reflection or diffraction. The hybrid Taguchi-genetic algorithm (HTGA) combines the Taguchi method with the genetic algorithm (GA). In this paper, we bring up a novel HTGA algorithm that utilizes time of arrival (TOA) measurements from three BSs to locate MS. The proposed algorithm utilizes the intersections of three TOA circles based on HTGA to estimate the MS location. Finally, we compare HTGA with GA and find that the Taguchi algorithm can enhance genetic algorithm. We also can find that the average convergence of generation number will not be affected no matter which propagation models we use. Obviously HTGA is more robust, statistically sound, and quickly convergent than the other algorithms. The simulation results show that the HTGA can converge more quickly than GA and furthermore the HTGA can enhance the accuracy of the mobile location.

## 1. Introduction

The mobile positioning is an important research topic in wireless communication. In recent years, it has gained considerable attention. Several researchers are concentrated on how to achieve higher accuracy in positioning. There are several wireless location schemes such as signal strength (SS), angle of arrival (AOA), time of arrival (TOA), and time difference of arrival (TDOA) techniques. The location of a mobile station (MS) is measured by the TOA between three or more base stations (BSs) with velocity of light [1, 2] in TOA method. The direction of the arrival signal [1] is detected by an antenna array and a directive antenna in AOA scheme where the MS is constrained along a line. The location of an MS is measured by the difference of arrival time between three or more BSs [3] in TDOA method. In particular, time-based and angle-based categories have their own advantages and disadvantages. The angle-based schemes have to know the minimum of two BSs to determine the MS location and the BSs do not require synchronization. On the contrary, the time-based schemes require at least three BSs and need synchronization. However, the time-based schemes usually provide much better positioning accuracy than angle-based schemes. Nowadays, There are a lot of applications of wireless location services, such as the intelligent transportation system (ITS) [4] and the emergency 911 (E-911). The public safety officer can see the caller's phone number and accurate location by E-911, and ITS combines a variety of positioning technologies to enhance the safety and efficiency of the transportation systems.

Non-line-of-sight (NLOS) is an important issue in MS location estimation problem. The line-of-sight (LOS) propagation is usually unavailable, especially in urban or suburban areas. NLOS error is always positive and large that arises when a distance is estimated from a calculation [5]. Some methods for mitigating NLOS error have been proposed in the literature [6–11]. In [2], the geometrical location algorithm was proposed to reduce the NLOS error. In [12, 13], range-scaling algorithm (RSA) was proposed. To adjust TOA

by scaling NLOS-corrupted TOA measurements with the utilization of the factors that are estimated from a constrained nonlinear optimization process is the way to mitigate the NLOS. Reference [14] proposed hybrid TDOA and AOA techniques for MS location estimation in wideband code-division multiple access (CDMA) cellular systems.

One of the important issues that affects the implementation of location scheme in wireless communication systems is *hearability*. The definition of *hearability* is the ability to receive sufficient signals from the number of neighboring BSs simultaneously as soon as the signal power level is defined. In rural areas, only 35% of signal strength indication received by three BSs is stronger than −100 dB; in contrast, it is about 84% in urban areas [15]. Usually, larger geographic area is covered by a BS in rural areas; an MS receives signals from neighboring BSs more difficultly, because each BS usually covers a large area and the *hearability* of an MS is very low for neighboring BSs.

Genetic algorithm (GA) was proposed by Holland [16], and it has utilized optimum approaches in various fields recently. GA has become one of the most important evolutionary computation methods. GA is an optimum approach method inspired by the ability of organisms to evolve. The major sources of such variation are genetic recombination and mutation. GA has been utilized in many applications in a variety of fields such as control engineering, aeronautics, image processing, and structure analysis. The authors of this paper proposed a novel positioning algorithm based on GA to locate MS when three BSs are available for location purpose [17].

The hybrid Taguchi-genetic algorithm (HTGA) was proposed by Jyh-Horng Chou. It combines both the Taguchi method [18–20] and the GA [21]. The Taguchi method replaced the crossover operation in the HTGA. The Taguchi method combined the systematic reasoning ability with crossover operations to select better genes. It used the crossover operations to generate the representative chromosomes and it will be the new potential offspring. Hence, the Taguchi algorithm can enhance genetic algorithms. In this way, the HTGA can quickly converge and be more robust and statistically sound [22–24]. HTGA has smaller standard deviations of function values than the orthogonal genetic algorithm with quantization (OGA/Q) proposed by [25]. It particularly proposes the following four enhancements in the HTGA with continuous variables for global optimization. First, a real coding technique utilizes continuous variables to solve optimization problems. Second, the crossover operators utilize an arithmetical operator derived from convex set theory to integrate the one-cut-point crossover. Third, the two-level orthogonal array, two tools of the Taguchi method, and signal-to-noise ratio (SNR) are applied in this study. The value of the object function decreases gradually during the iterations. Generally speaking, the value will converge when the solution does not change after specific number of generation. Fourth, the mutation operator is also derived from convex set theory. The proposed algorithm utilizes the intersections of three TOA circles, based on HTGA, to estimate the MS location in NLOS environments. Simulation results will show that we proposed a better method of location estimation compared with the Taylor series algorithm (TSA), linear lines of position algorithm (LLOP), and RSA. Moreover, HTGA is more robust and can converge more quickly than GA.

The remainder of this paper is organized as follows. In Section 2, we describe some related localization methods. Section 3 briefly describes how HTGA works. In Section 4, we propose the algorithm based on HTGA to estimate MS location. Section 5 compares the performance of the proposed algorithm with the other methods through simulation results. Finally, Section 6 draws some conclusions.

## 2. Related Localization Methods

*2.1. The Taylor Series Algorithm (TSA).* We take the constraint on *hearability* into account, so the number of BSs is limited to three in this paper for estimating the location. The coordinates for BS1, BS2, and BS3 are given by $(X_1, Y_1) = (0, 0)$, $(X_2, Y_2) = (X_2, 0)$, and $(X_3, Y_3)$, respectively. The distances between BS and the MS can be expressed as

$$r_i = \sqrt{(x - X_i)^2 + (y - Y_i)^2}, \tag{1}$$

where $(x, y)$ and $(X_i, Y_i)$ are the location of the MS and $i$th BS, respectively. Assume $(x_v, y_v)$ is the initial estimated position; then $x = x_v + \delta_x$, $y = y_v + \delta_y$. Using the Taylor series expansion by linearizing the TOA equations and keeping up with the second-order terms, the equation can be expressed as

$$A\delta \cong z, \tag{2}$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}, \quad \delta = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}, \quad z = \begin{bmatrix} r_1 - r_{v1} \\ r_2 - r_{v2} \\ r_3 - r_{v3} \end{bmatrix},$$
$$a_{i1} = \left.\frac{\partial r_i}{\partial x}\right|_{x_v, y_v}, \quad a_{i2} = \left.\frac{\partial r_i}{\partial y}\right|_{x_v, y_v}, \tag{3}$$
$$r_{vi} = \sqrt{(x_v - X_i)^2 + (y_v - Y_i)^2}, \quad i = 1, 2, 3.$$

The least-squares (LS) solution to the estimation problem can be expressed as

$$\delta = (A^T A)^{-1} A^T z. \tag{4}$$

An initial guess of the MS location starts the recursive process and then TSA repeats the iteration after computations. Because the initial guess of the MS location is not accurate enough, the iterative process is not always convergent [26, 27].

*2.2. Linear Lines of Position Algorithm (LLOP).* The algorithm uses the linear equation derived from the original nonlinear range equations. The linear LOP equation passes through the intersections of the two circles for TOA measurements instead of circular LOP. The linear equations can be obtained by squaring and subtracting the distances between BS and the MS. The MS location can be expressed as [2]

$$G\phi = h, \tag{5}$$

where $\phi = \begin{bmatrix} x \\ y \end{bmatrix}$ denotes the MS location, $G = \begin{bmatrix} X_2 & 0 \\ X_3 & Y_3 \end{bmatrix}$, and $h = (1/2)\begin{bmatrix} r_1^2 - r_2^2 + X_2^2 \\ r_1^2 - r_3^2 + X_3^2 + Y_3^2 \end{bmatrix}$.

The LS solution to (5) is also given by

$$\phi = (G^T G)^{-1} G^T h. \tag{6}$$

*2.3. Range-Scaling Algorithm (RSA).* Reference [12] utilizes TOA measurements from three BSs to estimate the MS location. The constrained optimization algorithm can find the normalized scale factors to adjust the measured distance error caused by NLOS. Some constraints on the normalized scale factors are based on the geometry of the cell layout of three BSs and the ranges of three BSs depicted in circles. The algorithm utilized the proposed object function written as the vector form to compute normalized scale factors and then mitigate the NLOS error with the utilization of the normalized scale factors.

## 3. The Hybrid Taguchi-Genetic Algorithm (HTGA)

HTGA combines the GA and the Taguchi method [22, 23]. In the HTGA, the Taguchi method replaces the step of crossover operation in GA. In GA, the crossover operation has two different ways, for example, single-point crossover and double-point crossover. Two ways can be chosen to find the crossover point randomly. In the Taguchi method, we can use the orthogonal array to calculate the SNR. And we can obtain the optimal chrome. The part of estimation method proposed by GA can be found in [17]. The following steps describe the HTGA approach, and Figure 1 shows the signal flow diagram.

*Step 1.* Input parameter setting includes population size = 50, crossover rate = 0.7, mutation rate = 0.02, reproduction rate = 0.28, and generation number = 30. Output parameter setting includes the optimal chromosome and fitness value.

*Step 2* (initialization). Execute the algorithm to generate an initial population. Calculate the fitness values of the population.

*Step 3* (crossover operation). We utilize the Taguchi method to do the crossover operation. The probability of crossover is determined by crossover rate.

*Step 4.* Select a two-level orthogonal array. The orthogonal array $L_{64}$ ($2^{63}$) is used in the proposed method.

*Step 5.* Choose two chromosomes randomly at a time to execute matrix experiments.

*Step 6.* Calculate the fitness value and SNRs in the experiment.

*Step 7.* Calculate the effects of the different factors in the experiment.

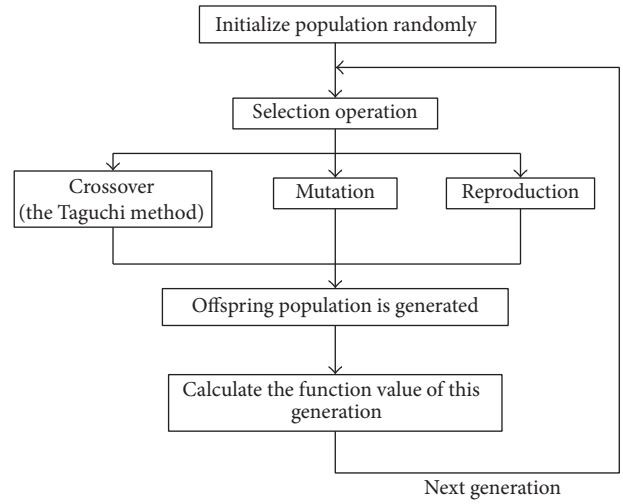



Figure 1: The signal flow chart of the hybrid Taguchi-genetic method.

*Step 8.* Based on Step 7, generate an optimal chromosome.

*Step 9.* Selection operation uses the roulette wheel approach.

*Step 10* (mutation operation). The probability of mutation is determined by mutation rate.

*Step 11.* Generate offspring population.

*Step 12.* If the condition is satisfied, then go to Step 13; if not, go back to Step 2.

*Step 13.* Generate the population via HTGA.

## 4. Proposed Location Algorithm Based on HTGA

According to a point of view of geometric approach, each BS measured distance forms a circle, and the center of the circle is BS. Multiple TOA measurements can estimate the MS position by the intersection of the circles. As shown in Figure 2, each of the following three equations indicates a circle for TOA:

$$\begin{aligned} &\text{Circle 1:} \quad x^2 + y^2 = r_1^2 \\ &\text{Circle 2:} \quad (x - X_2)^2 + y^2 = r_2^2 \\ &\text{Circle 3:} \quad (x - X_3)^2 + (y - Y_3)^2 = r_3^2. \end{aligned} \tag{7}$$

The three circles intersect at a point if there is no measured error or NLOS, and the point is the MS location. However, the location accuracy will degrade seriously because the NLOS error is very common in our life. Furthermore, the NLOS error appears as a positive bias at all times and is therefore greater than the true values, and consequently the circles will overlap with each other finally forming a region. The true MS location should be inside the overlap region of the three

circles as shown in Figure 2, that is, the area surrounded by $U$, $V$, and $W$. The intersections that are within this are defined as the feasible intersection. The feasible intersections have to satisfy the following three equations simultaneously:

$$\begin{aligned} x^2 + y^2 &\le r_1^2, \\ (x - X_2)^2 + y^2 &\le r_2^2, \\ (x - X_3)^2 + (y - Y_3)^2 &\le r_3^2. \end{aligned} \tag{8}$$

The possible MS location has to satisfy the above three equations, so we can estimate the location using the feasible intersections of the three circles, $U$, $V$, and $W$. By solving the circle equations (7), we can obtain the feasible intersections of three circles. The coordinates of $U$, $V$, and $W$ are represented as $(Ux, Uy)$, $(Vx, Vy)$, and $(Wx, Wy)$. Reference [12] proposed a nonlinear object function which can be seen as a cost function. It is the sum of the square of the distance from MS location to the intersection of the three circles. The object function of HTGA is

$$\begin{aligned} f(x, y) = {} & (x - U_x)^2 + (y - U_y)^2 + (x - V_x)^2 \\ & + (y - V_y)^2 + (x - W_x)^2 + (y - W_y)^2. \end{aligned} \tag{9}$$

Another problem is to avoid the condition of the NLOS error being too large; that is, one circle is fully covered by another circle. If $r_i > L_{ij} + r_j$, we adjust the measured TOA to $r_i = L_{ij} + r_j$ $(i, j = 1, 2, 3;\ i \neq j)$. There is at least one intersection for any two circles to ensure the algorithm is applied. We apply HTGA to obtain the approximation of the MS location. We should calculate the ranges of variables first in order to do the encoding. The ranges of variables $x$ and $y$ are the maximum and minimum among the three points $U$, $V$, and $W$:

$$\begin{aligned} x_{\min} &= \min\{Ux, Vx, Wx\}; \qquad x_{\max} = \max\{Ux, Vx, Wx\} \\ y_{\min} &= \min\{Uy, Vy, Wy\}; \qquad y_{\max} = \max\{Uy, Vy, Wy\}. \end{aligned} \tag{10}$$

We restrict chromosome in the overlap region of the three circles except for the upper bound and lower bound of the variables; that is to say, we only compute chromosomes which satisfy the three inequalities (8). This method can reduce the probability of bad convergence and computation complexity. The value of the object function decreases gradually during the iterations. Generally speaking, the value will converse when the solution does not change after specific number of generation.

## 5. Simulation Results

Computer simulations are conducted to illustrate the performance of the proposed positioning schemes. In the simulations, the coordinates of BSs are as $BS_1$: (0, 0), $BS_2$: (1732 m, 0), and $BS_3$: (866 m, 1500 m). Five thousand independent trials are performed for each simulation (and the region is formed by the points $BS_1$, $BS_2$, and $BS_3$ with sides $I$, $J$, and $K$ being of uniform distribution). And the MS location is chosen randomly within the region as shown in Figure 3. Adjusting the parameter of GA affects just a little the result because of the good convergence. We set the parameter as the following analysis: number of generation = 30, crossover rate = 0.7, and mutate rate = 0.002. We take the NLOS effects into account in the simulation. Two propagation models are adopted, namely, circular disk of scatterers model (CDSM) [25] and uniformly distributed noise model [12], respectively.

CDSM is the first NLOS propagation model which assumes that there are scatterers spreading around the MS when the signal travels between MS and BSs [25]. The signal

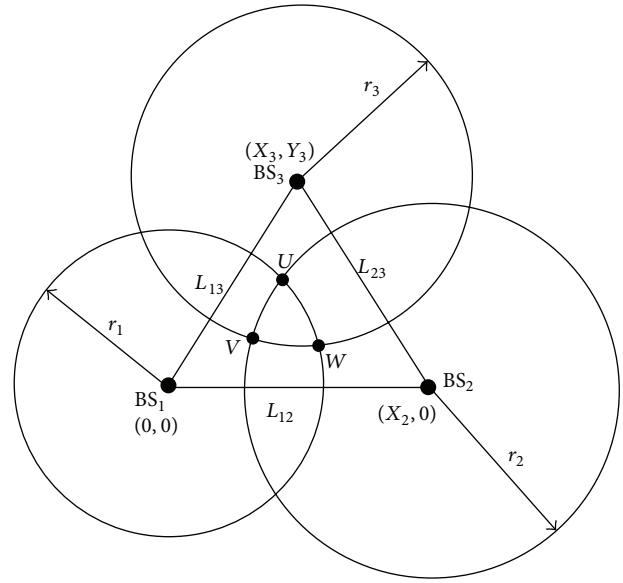



Figure 2: Geometric layout of the three circles of TOA method.

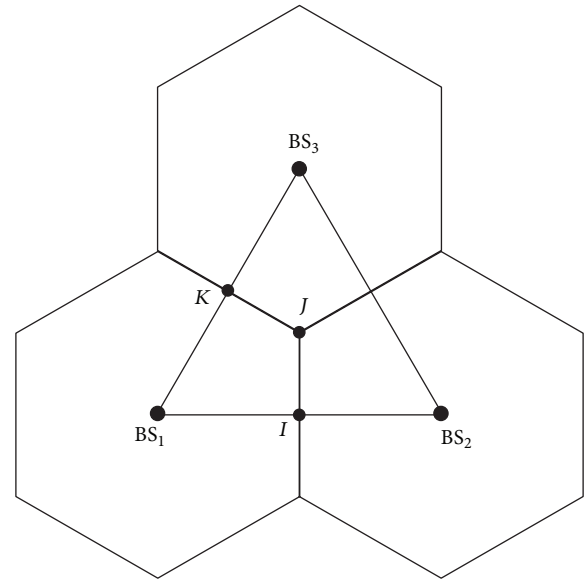



Figure 3: Cell layout showing the relationship between the true ranges and inter-BS distances.

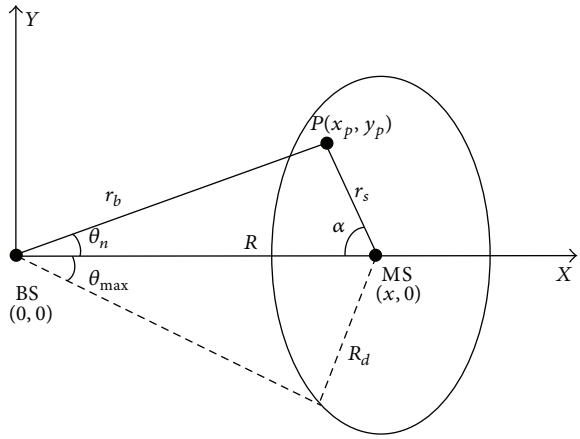



Figure 4: Geometry of circular disk of scatterers model (CDSM).

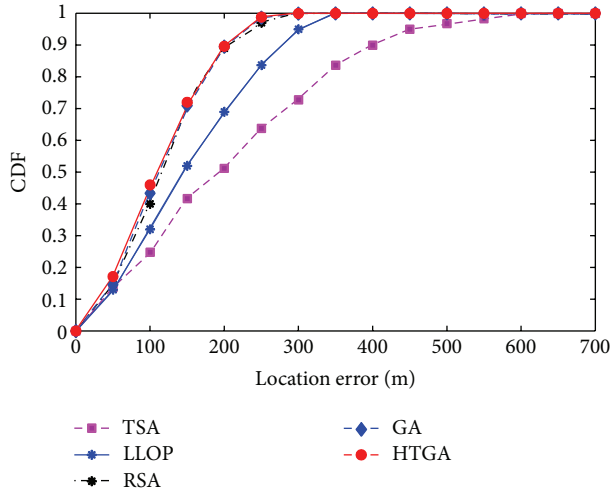



Figure 5: The CDF of the location error for various methods.

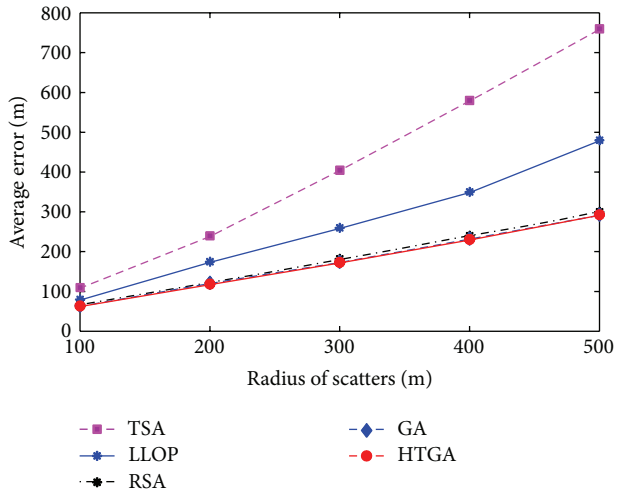



Figure 6: Average location error versus the radius of scatters.

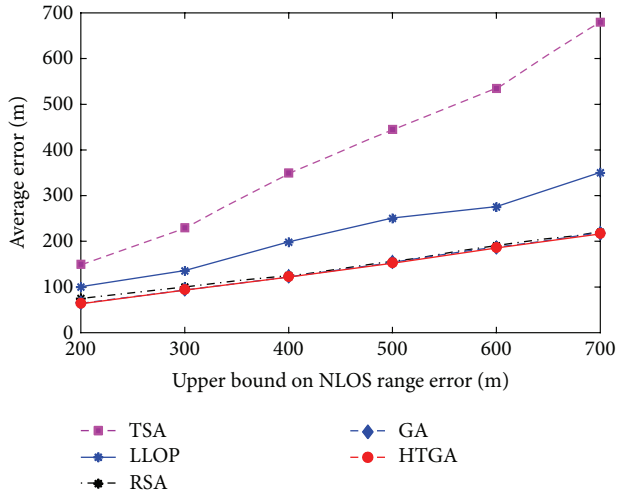


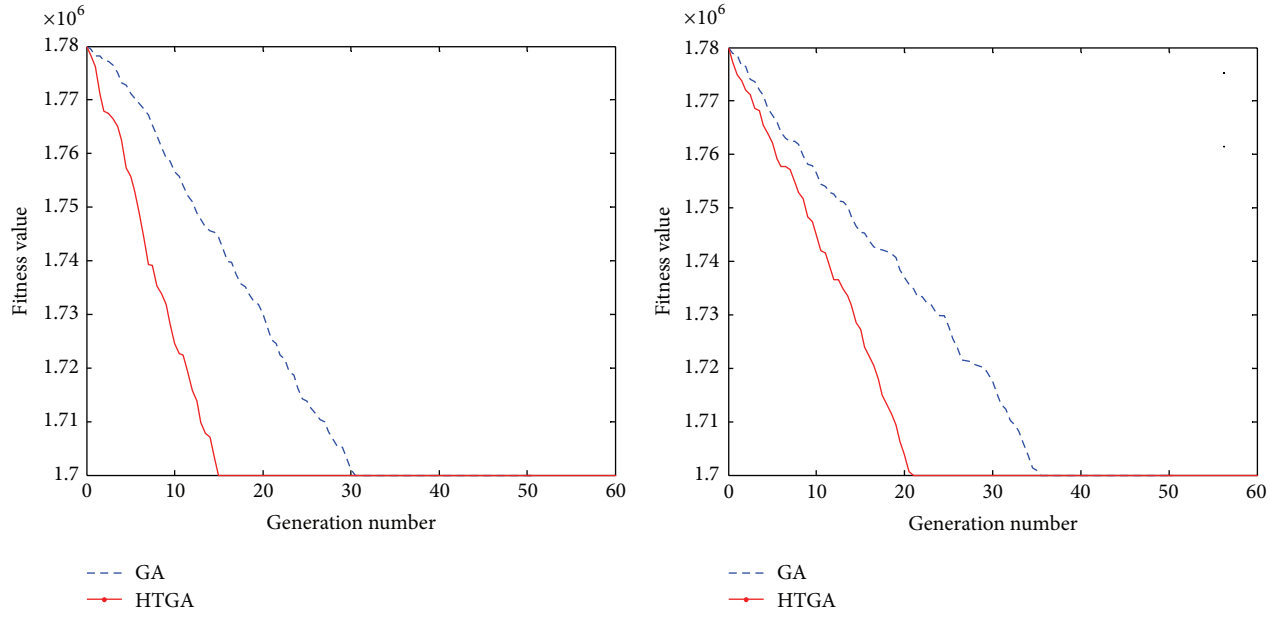

Figure 7: Average location error versus the upper bound of NLOS errors.

goes through a single reflection at the scatterers. The sum of the distances between the BS and the scatterer and between the MS and the scatterer is the measured ranges. Figure 4 shows the geometry of CDSM, where $p$ is the scatterer point. The measured ranges are the sum of the distances between the BS and the scatterer ($r_b$) and between the MS and the scatterer ($r_s$). The measured AOA at the BSs is the angle between the BSs and the scatterer where the signal is reflected. If the scatterer point is on the circle, the AOA measurement will be the maximum. Thus, if the scatterer radius is larger, the estimation error will be larger. The probability of lower range error is larger than the higher one.

Figure 5 shows the cumulative distribution functions (CDFs) of the location error for different algorithms using the CDSM. The radius of the scatterers is considered to be 200 m. Note that the proposed method is always better than TSA and LLOP for the error model considered. The positioning precision of the proposed HTGA is slightly better than that of RSA and GA.

MS location accuracy is measured in terms of root-mean-square (RMS) error between the actual MS location and the desired MS location. The radius effect of the CDSM on the average location error compared with other existing methods is as shown in Figure 6. It can also be observed that the sensitivity of the proposed methods with respect to the NLOS effect is much less than that for TSA and LLOP. Obviously the average location error of the proposed HTGA is slightly less than RSA and GA. The simulations result shows that the proposed HTGA can yield the MS location more accurately than the other algorithms.

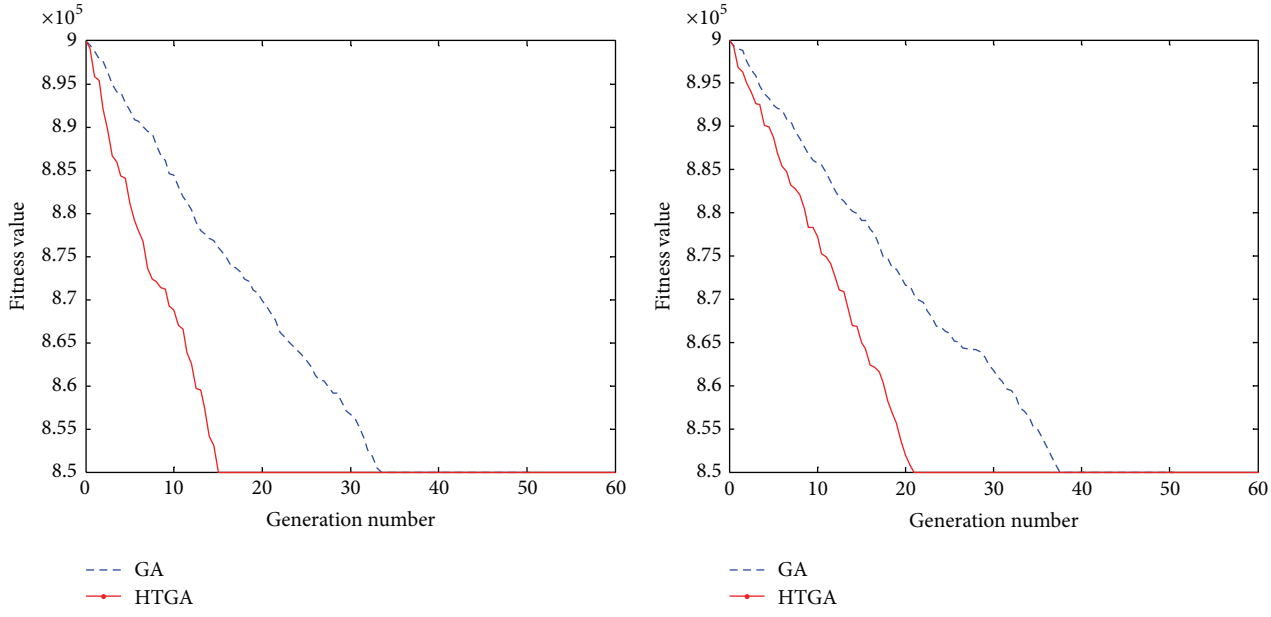
The uniformly distributed noise model [12] is applied for the second NLOS propagation model. The TOA measurement error is assumed to be uniformly distributed over $(0, U_i)$. $U_i$ is the upper bound of the measurement error for $i = 1, 2, 3$. For example, if we set the upper bound of error as 200 m, the TOA measurement error will be uniformly distributed over (0, 200 m). The effect of various methods used with upper bound of NLOS error on the average location




(a) Radius of scatters = 200 m

(b) Radius of scatters = 500 m

Figure 8: The convergence speed of GA and HTGA in each generation by using CDSM.




(a) Upper bound = 200 m

(b) Upper bound = 700 m

Figure 9: Some of the convergence results of GA and HTGA in each generation for uniformly distributed noise model.

error is shown in Figure 7, in which we estimate the average location error of the mobile station in terms of RMS error. When the upper bound of NLOS error increases, the RMS error increases. Simulation results show that the proposed HTGA has better location estimation than TSA, LLOP, RSA, and GA.

The relationships between the fitness value and the number of generation for HTGA and GA are, respectively, shown in Figures 8 and 9. HTGA outperforms traditional GA in obtaining the optimal solutions because of its fast convergence ability. When we take the location estimation of MS at each generation, the generation is considered to be convergent if the difference of the estimated MS and best performance is less than 0.01.

Figures 10 and 11, respectively, show the performance of the average generation numbers for convergence between HTGA and GA, in which the CDSM model and uniformly distributed noise model are applied. We can obtain the

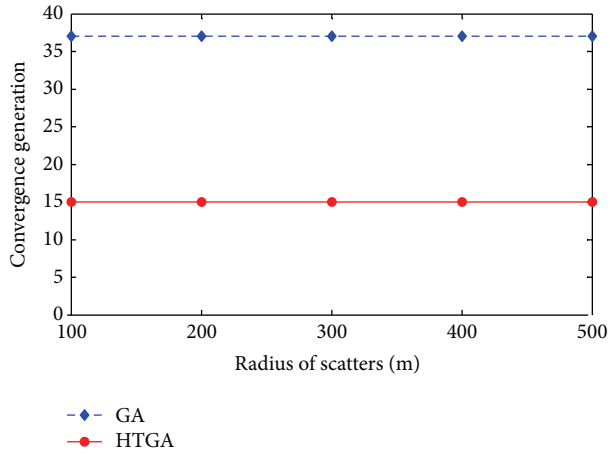



Figure 10: Convergence generation number versus the radius of scatters.

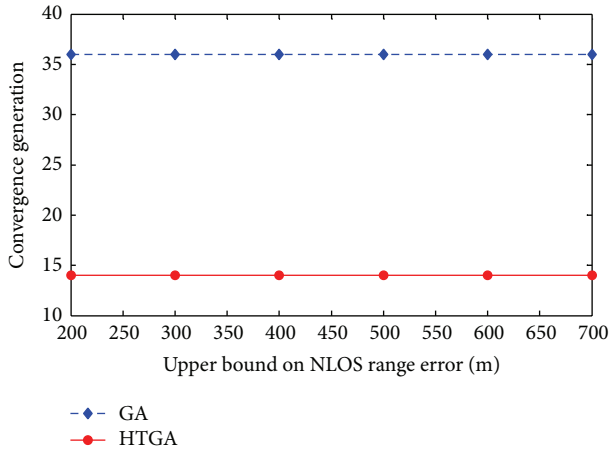



Figure 11: Convergence generation number versus the upper bound of NLOS errors.

average generation number after running five thousand times. Note that the average generation number of HTGA and GA is, respectively, 14 and 26 when we choose the maximal generation number as 30. We also can find that the average generation number is independent of the propagation models.

Simulation results show that, no matter which NLOS propagation model is considered, the proposed HTGA can always give better location estimation. Note that both HTGA and GA approach can provide more accurate MS location estimations. Although the performances of HTGA and GA are very close, HTGA still converges faster than GA. Therefore, there are many discussions about this phenomenon [22, 23]. The simulation results show that the proposed HTGA can reduce the number of iterations and decrease the calculation complexity. In mobile communication device, the computational complexity is an important topic. The proposed HTGA method can solve this problem. Thus the mobile device can decrease the computational complexity and power resources.

## 6. Conclusion

In this paper, we propose the schemes based on HTGA to estimate MS location from three BSs. In order to eliminate NLOS errors and without any *a priori* information about the NLOS error, the proposed methods utilize all the feasible intersections which are generated by three TOA circles to estimate the MS location. Obviously HTGA is not only robust but also quicker than GA. We also can found that the average generation number for convergence is not dependent on the propagation models. Simulation results show that the location accuracy of the proposed methods is much better comparing with the standard TSA, LLOP, RSA, and GA. On the other hand, reducing the signal-processing time of the mobile device can increase not only the processing capabilities available for other purposes but also the saving of the power of battery.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

[1] S. Al-Jazzar, J. J. Caffery, and H.-R. You, "A scattering model based approach to NLOS mitigation in TOA location systems," in *Proceedings of the IEEE Vehicular Technology Conference*, vol. 2, pp. 861–865, 2002.

[2] J. J. Caffery, "A new approach to the geometry of TOA location," in *Proceedings of the 52nd Vehicular Technology Conference (IEEE VTS Fall VTC '00)*, pp. 1943–1949, 2000.

[3] Y. Jeong, H. You, W. C. Lee, D. Hong, D. H. Youn, and C. Lee, "Wireless position location system using forward pilot signal," in *Proceedings of the IEEE 51st Vehicular Technology Conference Proceedings (VTC '00)*, pp. 1354–1357, 2000.

[4] J. J. Caffery and G. Stuber, "Overview of radiolocation in CDMA cellular systems," *IEEE Communications Magazine*, vol. 36, no. 4, pp. 38–45, 1998.

[5] P. C. Chen, "A non-line-of-sight error mitigation algorithm in location estimation," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '99)*, vol. 1, pp. 316–320, 1999.

[6] I. Guvenc and C.-C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.

[7] S. Al-Jazzar, J. J. Caffery, and H.-R. You, "Scattering-model-based methods for TOA location in NLOS environments," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 583–593, 2007.

[8] S. Al-Jazzar, M. Ghogho, and D. McLernon, "A joint TOA/AOA constrained minimization method for locating wireless devices in non-line-of-sight environment," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 468–472, 2009.

[9] Y. Xie, Y. Wang, P. Zhu, and X. You, "Grid-search-based hybrid TOA/AOA location techniques for NLOS environments," *IEEE Communications Letters*, vol. 13, no. 4, pp. 254–256, 2009.

[10] H. Miao, K. Yu, and M. J. Juntti, "Positioning for NLOS propagation: algorithm derivations and Cramer-Rao bounds," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 5, pp. 2568–2580, 2007.

[11] C. K. Seow and S. Y. Tan, "Non-line-of-sight localization in multipath environments," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 647–660, 2008.

[12] S. Venkatraman, J. 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.

[13] S. Venkatraman, J. J. Caffery, and H.-R. You, "Location using LOS range estimation in NLOS environments," in *Proceedings of the Vehicular Technology Conference*, vol. 2, pp. 856–860, 2002.

[14] L. Cong and W. Zhuang, "Hybrid TDOA/AOA mobile user location for wideband CDMA cellular systems," *IEEE Transactions on Wireless Communications*, vol. 1, no. 3, pp. 439–447, 2002.

[15] J. H. Reed, K. J. Krizman, B. D. Woerner, and T. S. Rappaport, "An overview of the challenges and progress in meeting the E-911 requirement for location service," *IEEE Communications Magazine*, vol. 36, no. 4, pp. 30–37, 1998.

[16] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Mich, USA, 1975.

[17] C.-S. Chen, J.-M. Lin, W.-H. Liu, and C.-L. Chi, "MS location estimation with genetic algorithm," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 95, no. 1, pp. 305–312, 2012.

[18] M. S. Phadke, *Quality Engineering Using Robust Design*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1989.

[19] P. J. Ross, *Taguchi Techniques for Quality Engineering*, McGraw-Hill, New York, NY, USA, 1989.

[20] G. Taguchi, S. Chowdhury, and S. Taguchi, *Robust Engineering*, McGraw-Hill, New York, NY, USA, 2000.

[21] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*, Wiley, New York, NY, USA, 1997.

[22] J.-T. Tsai, T.-K. Liu, and J.-H. Chou, "Hybrid Taguchi-genetic algorithm for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 4, pp. 365–377, 2004.

[23] S. Picek, M. Golub, and D. Jakobovic, "Influence of the crossover operator in the performance of the hybrid Taguchi GA," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '12)*, pp. 1–8, 2012.

[24] J.-T. Tsai, J.-H. Chou, and T.-K. Liu, "Optimal design of digital IIR filters by using hybrid Taguchi genetic algorithm," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 3, pp. 867–879, 2006.

[25] Y.-W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 1, pp. 41–53, 2001.

[26] W. Foy, "Position-location solutions by Taylor series estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 12, no. 2, pp. 187–194, 1976.

[27] D. Torrieri, "Statistical theory of passive location systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 20, no. 2, pp. 183–197, 1984.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 716232, 9 pages
http://dx.doi.org/10.1155/2014/716232

*Research Article*

# Numeric Evaluation on the System Efficiency of the EPC Gen-2 UHF RFID Tag Collision Resolution Protocol in Error Prone Air Interface

**Xin-Qing Yan,[1] Yang Liu,[1] Bin Li,[2] and Xue-Mei Liu[1]**

[1] *School of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450011, China*
[2] *Department of Traffic and Transportation, Fujian University of Technology, Fuzhou 350108, China*

Correspondence should be addressed to Xin-Qing Yan; yanxq@ncwu.edu.cn

Received 28 June 2013; Accepted 1 February 2014; Published 9 March 2014

Academic Editor: Chang Wu Yu



Efficient resolution of the tag collisions is the key for the RFID systems to be universally adopted. To resolve the tag collision occurred in the UHF RFID system, the EPC Gen-2 protocol, proposed by the EPCglobal Inc., was adopted as an international standard and have been widely accepted in industry. However, little research works has been taken to evaluate the performance of this protocol in nonperfect and error prone air interface. In this paper, to evaluate the performance of the EPC Gen-2 protocol, a system efficiency model is proposed, and, based on the model, numeric simulations are performed to evaluate the EPC Gen-2 protocol in error prone air interface. Simulation results reveal that the system efficiency of the EPC Gen-2 protocol is seriously affected by the capture and noise effects.

## 1. Introduction

Radio frequency identification (RFID) uses radio frequency signals to exchange information between the electronic transponders (tags) and the interrogator (reader) and enables the identification of multiple tagged physical items without line-of-sight. As a bridge to connect the physical world and the cyber space, RFID technology is regarded as the main enabler of the "Internet of Things [1, 2]" and ubiquitous computing [3, 4]. In fact, a RFID system can be viewed as a special wireless sensor network, in which each tag (sensor node) can only transmit its digital identifier but no other information to the reader (sink node). The past few years have witnessed the adoption of RFID systems in a lot of systems [5].

Especially in the recent years, the ultra high frequency (UHF) RFID system, which works in the frequency range of 860–920 MHz, gains special attention due to its rapid communication speed, long tag identification range, and low-cost of the passive tags. UHF RFID system is expected to bring a revolution to the logistical and supply chain management systems [6].

Despite of the promising future, the universal adoption of the RFID system is technically affected by tag collision [7]. When multiple tags try to transmit their data simultaneously to the reader, their radio signals will interfere in the wireless communication channel and be garbled at the reader. When tag collision occurs, what the reader can get is only a collision signal but no useful information. Due to tag collision, the RFID system suffers from low tag read rate and long identification delay.

Tag collision is more serious in the UHF RFID system due to the long tag identification distance and therefore more tags will be in the interrogation zone of the reader. Not only does tag collision prolong the tag identification time but also affects the tag read rate. For example, the tag read rate in typical UHF RFID system is only about 60–70% [8].

Due to the extreme constraints on computation and communication put on them, the passive tags can only get power supply by backscattering the radio frequency signals broadcasted by the reader, they cannot detect the collision occurred in the wireless communication channel or coordinate with each other to avoid the collision. Tag collision can only be

arbitrated by the reader with some deliberately designed tag collision resolution protocols.

Proposed tag collision resolution protocols can be basically categorized as the binary splitting tree based and the frame slotted ALOHA based protocols [9, 10]. The binary splitting tree based protocols suffer from scalability and message complexity and are sensitive to the error in the wireless communication channel. So, we will not discuss these binary splitting tree based protocols further in this paper.

Due to their simplicity and robustness, the frame slotted ALOHA based protocols are widely adopted in RFID systems to resolve tag collisions. Especially, to resolve the tag collision occurred in the UHF RFID system, as a variant of the frame slotted ALOHA protocol, the EPC Class-1 Generation-2 air interface protocol (the EPC Gen-2 protocol) was proposed by the EPCglobal Inc. [11] and accepted as an international standard the ISO/IEC-18600C. Now, this protocol has been widely accepted in industry and by main RFID manufactures.

To evaluate and optimize the performance of the EPC Gen-2 protocol, a lot of researches have been taken, such as the performance analysis presented in [12–14], the empirical study presented in [15, 16], and the optimization presented in [17–20]. In a lot of research works, it is assumed that the air interface between the reader and the electronic tags is perfect, without any signal loss or other communication effects. But this assumption may not hold in real UHF RFID system. Due to long tag identification distance, the wireless signals transmitted from tags and broadcasted by reader may be absorbed, reflected, garbled, and captured in the complex deployment environment; the air interface is far from perfect, and error occurs frequently in the wireless communication channel.

Our motivations in this paper are to evaluate the performance of the EPC Gen-2 protocol in error prone air interface. The main contributions of this paper are as follows. Firstly, a model is proposed to evaluate the system efficiency of the EPC Gen-2 protocol in error prone air interface. Secondly, numeric simulations are performed to evaluate the system efficiency of the protocol in the air interface with different capture and noise effects and to reveal the influences of capture and noise effects on the system efficiency of the protocol.

The rest of this paper is organized as follows. Section 2 introduces briefly the EPC Gen-2 protocol and the $Q$-adjustment algorithm adopted in the protocol. Section 3 presents a model to evaluate the system efficiency of the EPC Gen-2 protocol in error prone air interface. Section 4 evaluates the system efficiency of the protocol with different capture and noise effects and reveals the influence of these effects on the performance of the protocol. Finally, Section 5 concludes this paper and proposes some works for future research.

## 2. The EPC Gen-2 Protocol for Tag Collision Resolution in the UHF RFID System

*2.1. The EPC Gen-2 Protocol.* In the EPC Gen-2 protocol, a tag collision resolution cycle is called a round, which consists of a series of command broadcasted by the reader and responses transmitted by the tag. At the beginning of a round, the protocol asks the reader to broadcast a *SELECT* command, and only tags which receive this command will respond in the round. Afterwards, the reader broadcasts a *Query* command to start tag identification. In fact, a tag collision resolution round in the EPC Gen-2 protocol is defined as the interval between two successive *Query* commands.

In the *Query* command, there is also an integer $Q$ (with initial value 4) broadcasted by the reader to tags in its vicinity. Upon receiving the *Query* command, every unidentified tag randomly generates an integer in the range of $[0, 2^Q - 1]$ and stores the value in its *SC* register. The tag whose *SC* equals to 0 will generate a 16-bit *SN* and transmit the *SN* to the reader immediately.

If only one tag answers back after the command, the slot is single occupied, the reader can decode the *SN* and echoes back an *ACK* command with the *SN*. Each answering tag compares the *SN* which it generated with the one it received in the *ACK* command. If these two *SN*s match, the tag transmits its digital identifier to the reader, the tag is identified, and the slot results in successfulness. In such case, the identified tag turns itself into sleep mode and ceases to responding to the following queries, until being aroused by the reader again.

If no tag answers back, the slot results in idle. If two or more tags transmit their *SN*s, the slot results in collision.

After a single reply slot, the reader broadcasts a *QueryRep* command and asks every unidentified tag to decrease its *SC* by 1. The tag whose *SC* equals to 0 will respond with the randomly generated 16-bit *SN*, as described above.

After an idle or a collision slot, the protocol may decide whether to adjust the value of $Q$ according to the $Q$-adjustment algorithm described below. If the value of $Q$ is adjusted, the protocol asks the reader to broadcast a *QueryAdjust* command with the new value of $Q$, and let every unidentified tag to regenerate its *SC*. Otherwise, the protocol asks the reader to issue a *QueryRep* command to continue asking unidentified tags to decrease their *SC*s.

We can see that the performance of the EPC Gen-2 protocol is seriously affected by the choice of $Q$ and how to adjust its value.

*2.2. The Q-Adjustment Algorithm.* As we have introduced, in the EPC Gen-2 protocol, after an idle or a collision slot, the protocol uses the $Q$-adjustment algorithm to determine whether to adjust the value of $Q$ or not.

In the $Q$-adjustment algorithm, there is a float number $Q_{fp}$, representing the float value of $Q$, with initial value $Q_{fp} = 4.0$. Besides, the algorithm also uses a float value $C$ ($0.1 < C < 0.5$) to adjust the value of $Q_{fp}$.

After an idle slot, the value of $Q_{fp}$ is subtracted with $C$. After a collision slot, the value of $Q_{fp}$ is added with $C$. The value of $Q$ is set to the integer nearest to $Q_{fp}$. If the value of $Q$ is changed, the protocol will ask the reader to issue a *QueryAdjust* command with the new value of $Q$. Otherwise, the protocol asks the reader to issue an *QueryRep* command.

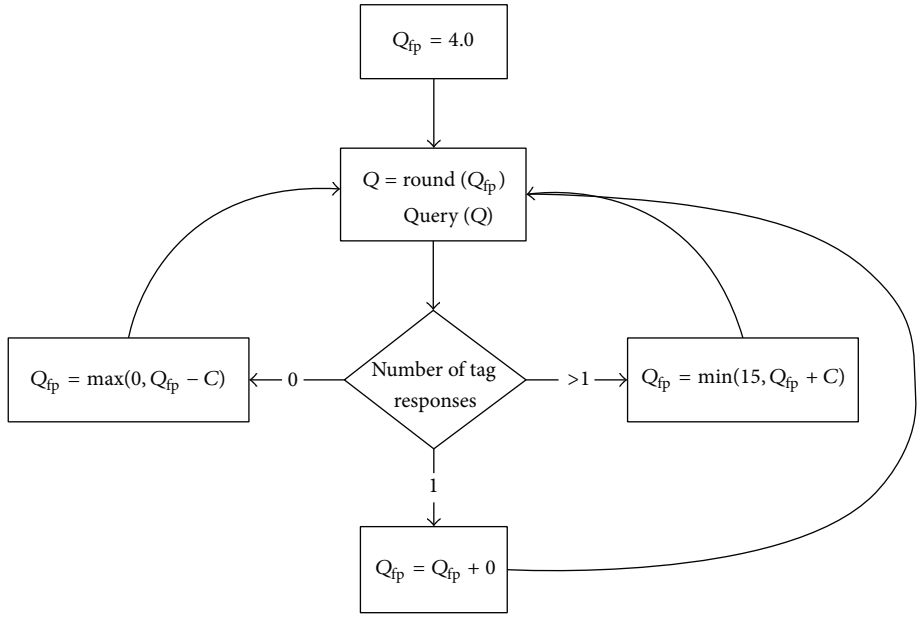



Figure 1: The $Q$-adjustment Algorithm, extracted from [11].

The $Q$-Adjustment algorithm works as depicted in Figure 1.

Although it is specified in the EPC Gen-2 protocol that the value of $C$ should be in $(0.1, 0.5)$, small values of $C$ should be used for large value of $Q$, and larger values of $C$ should be used for small value of $Q$. But no optimal choice of $C$ for different value of $Q$ is specified.

Effective $Q$-adjustment algorithm is also a key for the EPC Gen-2 protocol to resolve the collisions caused by different number of tags efficiently.

## 3. System Model

*3.1. The Imperfect Air Interface.* In the UHF RFID system, as other passive RFID systems, tag collision resolution is based on the request/response model. First, a reader sends an interrogation signal to all tags within its vicinity, and tags respond back by backscattering their signals. The back scattering signals are significantly attenuated by distance.

Due to the long identification distance and complex deployment environment of the UHF RFID system, the wireless communication channel between the reader and tags is far from error free. The backscattering signals from tags are typically very weak and can be reflected or absorbed by the environment, and frame errors may also occur due to noise or interference in the air interface.

Errors in the air interface can be classified as the the reader-tag downlink channel error or tag-reader uplink error. In the downlink, a tag may not get the command message broadcasted by the reader, so no response will be transmitted. In the uplink, tag responses, but the reader may miss the signals or misunderstand the information.

Due to the fact that the reader can adjust its signal strength to overcome the error in the downlink channel, but tags cannot, in this paper, we will consider the error occurred in the tag-reader uplink channel; the reader cannot distinguish a collision slot from a successful slot due to the frame error occurred in the uplink.

Furthermore, we can see from the EPC Gen-2 protocol that in the tag-reader uplink channel, the data transmission can be divided into two stages. In the first stage, every tag whose $SC = 0$ will only transmit its 16-bit $SN$ to the reader. If the reader can decode the $SN$, in the second stage, the reader echoes the $ACK$ command and asks the responding tag to transmit its full digital identifier.

In such case, the reader cannot continue identifying the tag even if error occurs in the second stage. Since the responding tag will turn itself into sleep mode and stop responding to the following query commands in the cycle. So, in this paper, we only consider the case that error may occur in the uplink channel when tag tries to transmit its 16-bit $SN$ to the reader.

Due to frame error occurred in the air interface, the protocol may not be able to fully distinguish a successful slot with a collision slot, it sometimes takes a successful slot as a collision slots and occasionally regards a collision slot as a successful slot. But since that there is no signal transmission in the idle slot, the protocol can always distinguish an idle slot from a responding slot.

*3.2. Evaluation Model for the System Efficiency of the EPC Gen-2 Protocol.* In the EPC Gen-2 protocol, for the identification of $n$ tags using the current value $Q$, according to the Binomial distribution, the probability that $k$ tags responding in a slot can be calculated as

$$p(Q, k, n) = \binom{n}{k} \left(\frac{1}{2^Q}\right)^k \left(1 - \frac{1}{2^Q}\right)^{n-k}, \quad (1)$$

where $\binom{n}{k} = n!/(k!(n-k)!)$.

Table 1: Suitable tag population range for different values of $Q$.

| $Q$ | min($Q$) | max($Q$) | $p_i$(min($Q$)) | $p_i$(max($Q$)) | $p_{SC}$(min($Q$)) | $p_{SC}$(max($Q$)) |
|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 0.50 | 0.25 | 0.50 | 0.75 |
| 2 | 3 | 5 | 0.42 | 0.24 | 0.58 | 0.76 |
| 3 | 6 | 11 | 0.45 | 0.23 | 0.55 | 0.77 |
| 4 | 12 | 22 | 0.46 | 0.24 | 0.54 | 0.76 |
| 5 | 23 | 45 | 0.48 | 0.24 | 0.53 | 0.76 |
| 6 | 46 | 90 | 0.49 | 0.24 | 0.52 | 0.76 |
| 7 | 91 | 181 | 0.49 | 0.24 | 0.51 | 0.76 |
| 8 | 182 | 362 | 0.49 | 0.24 | 0.51 | 0.76 |
| 9 | 363 | 724 | 0.49 | 0.24 | 0.51 | 0.76 |
| 10 | 725 | 1448 | 0.49 | 0.24 | 0.51 | 0.76 |
| 11 | 1449 | 2896 | 0.49 | 0.24 | 0.51 | 0.76 |
| 12 | 2897 | 5792 | 0.49 | 0.24 | 0.51 | 0.76 |
| 13 | 5793 | 11585 | 0.49 | 0.24 | 0.51 | 0.76 |
| 14 | 11586 | 23170 | 0.49 | 0.24 | 0.51 | 0.76 |
| 15 | 23171 | 46340 | 0.49 | 0.24 | 0.51 | 0.76 |

Especially, the probabilities for a slot to result in *idle*, *single reply* or *two or more replies* can be calculated, respectively, as

$$\begin{aligned} p_i(Q,n) &= \left(1-\frac{1}{2^Q}\right)^n, \\ p_s(Q,n) &= \frac{n}{2^Q}\left(1-\frac{1}{2^Q}\right)^{n-1}, \\ p_c(Q,n) &= 1-p_i(Q,n)-p_s(Q,n) \\ &= 1-\left(1-\frac{1}{2^Q}\right)^n-\frac{n}{2^Q}\left(1-\frac{1}{2^Q}\right)^{n-1}. \end{aligned} \tag{2}$$

The probability for a slot to result in nonidle, $p_{SC}(Q,n)$, is $p_{SC}(Q,n) = 1 - p_i(Q,n)$. We can see that for a fixed $Q$, as tag population $n$ increases, $p_i(Q,n)$ keeps decreasing, while $p_{SC}(Q,n)$ keeps increasing.

For a given tag population $n$, it has been reported that in the perfect air interface, when $Q = |\log_2^n|$, the probability for a slot to result in single reply (successful), $p_s(Q,n)$, is maximized. On the other hand, in the perfect air interface, for a given value of $Q$, the suitable tag population range $[\min(Q), \max(Q)]$ can be calculated. When the tag population is in the suitable range, the minimum and maximum probabilities for a slot to result in idle and nonidle (successful or collision) can also be calculated, as shown in Table 1.

As presented in [21], the system efficiency (SE) of the protocol is defined as the ratio between the number of tags identified in a round (also the number of successful slot), $n$, and the total number of data slots, $S$, needed in the round, SE = $n/S$. In the EPC Gen-2 protocol, since that the protocol can abandon a frame and starts a new one according to the $Q$-adjustment algorithm, the number of data slots in a frame is not fixed. But the total number of data slots needed by the EPC Gen-2 protocol in a round can be calculated as

$$S = \sum S_k, \tag{3}$$

where $S_k$ specifies the number of data slots in the $k$th frame.

For the $S_k$ data slot in the $k$th frame with $Q$ value $Q_k$, according to (4), the mathematical expectations for the number of idle, successful, and collision slots, $N_i(S_k, Q_k)$, $N_s(S_k, Q_k)$, and $N_c(S_k, Q_k)$ are

$$\begin{aligned} N_i(S_k,Q_k) &= S_k p_i(Q_k,n), \\ N_s(S_k,Q_k) &= S_k p_s(Q_k,n), \\ N_c(S_k,Q_k) &= S_k p_c(Q_k,n). \end{aligned} \tag{4}$$

Suppose that in the error prone air interface, with probability $p$, the protocol regards a successful slot as a collision slot, and, with probability $q$, the protocol regards a collision slot as a successful slot; the system efficiency of the EPC Gen-2 protocol in the error prone air interface, SE, is calculated as

$$\mathrm{SE} = \frac{\sum\left((1-p)N_s(S_k,Q_k) + qN_c(S_k,Q_k)\right)}{\sum S_k}. \tag{5}$$

For the perfect air interface, we have $p = 0$ and $q = 0$, and SE is calculated as

$$\mathrm{SE} = \frac{\sum N_s(S_k,Q_k)}{\sum S_k} = \frac{\sum S_k p_s(Q_k,n)}{\sum S_k}. \tag{6}$$

## 4. Performance Evaluation

Since that for different value of $p$ and $q$, it is difficult to find a closure solution for (5) to evaluate the system efficiency of the protocol, so in this paper, numeric simulations are performed to evaluate the performance of the EPC Gen-2 protocol using the $Q$-adjustment algorithms. In the simulations, the tags to be identified by the reader in a round are divided into 2 groups. In the first group, the tag population varies from 1 to 300 with increment 1, and, in the second group, the tag population varies from 300 to 3000 with increment 100. The protocol is required to resolve all tag collisions and identify all tags in a round.

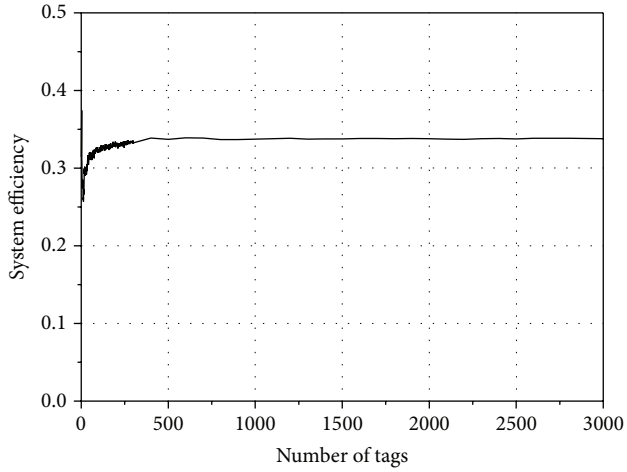



Figure 2: The system efficiency of the EPC Gen-2 protocol in perfect air interface.

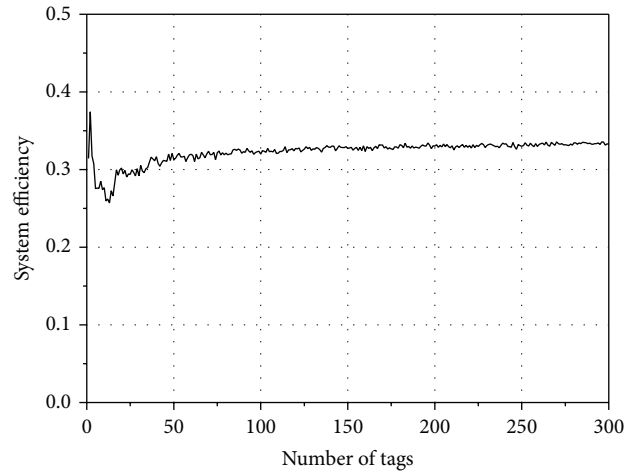



Figure 3: The system efficiency of the EPC Gen-2 protocol in perfect air interface when tag population is less than 300.

In order to gain a fair result, the simulations are performed 100 times. The system efficiencies of the EPC Gen-2 protocol gained in each time of the simulation are recorded and are averaged at last for comparison.

In the evaluation, according to (5), we will examine the system efficiency of the EPC Gen-2 protocol with the $Q$-adjustment algorithms in the following four kinds of air interface.

(i) The perfect air interface, in which no error occurs in signal transmission, and $p = q = 0$.

(ii) The air interface with capture effect, where a tag may be identified even if collision occurs in a slot; $p = 0$ and $q$ varies in $[0, 1)$.

(iii) The noise interface, where a successful slot may be regarded as a collision slot; $q = 0$ and $p$ varies in $[0, 1)$.

(iv) The general error prone air interface, where both $p$ and $q$ vary in $[0, 1)$.

In the numeric simulations, as stated in the EPC Gen-2 protocol and the $Q$-adjustment algorithm, the initial values of $Q$ and $Q_{fp}$ are set to 4 and 4.0, respectively. Since that the value of $C$ in the original $Q$-adjustment algorithm is only specified in the range of $(0.1, 0.5)$, in this paper, a middle value of $C$ is chosen, and we set $C = 0.3$.

*4.1. The Perfect Air Interface.* In the perfect air interface, there is no signal transmission error. If in a data slot, only one tag responds; the tag is identified by the reader. If two or more tags respond, collision occurs in the slot, and the reader can only detect a collision signal but no useful information.

In such case, the system efficiency of the EPC Gen-2 protocol can be calculated as

$$\mathrm{SE} = \frac{\sum N_s\left(S_k, Q_k\right)}{\sum S_k}. \tag{7}$$

The system efficiency of the EPC Gen-2 protocol using different $Q$-adjustment algorithms is shown in Figure 2. Especially, when the tag population is less than 300, the system efficiency of the protocol is shown in Figure 3.

Figures 2 and 3 reveal that for the identification of only a few tags, the EPC Gen-2 protocol performs unstably, as its system efficiency varies a little drastically. As the tag population increases, it performs stably with system efficiency varying around 0.33, which means that, in the perfect air interface, about three slots are needed to identify a tag.

*4.2. The Air Interface with Capture Effect.* In some radio frequency communication channel, a tag can be captured by the reader even if collision occurs in the data slot, and this is called the capture effect.

Capture effect is caused by the fact that responding tags may be scattered in different distances from the reader; the signals from a tag may overwhelm that from other responding tags that can be captured by the reader. Capture effect leads to the fact that there is a probability that a tag can be identified by the reader even if collision occurs in the air interface.

In the air interface with capture effect, the system efficiency of the EPC Gen-2 protocol can be calculated as

$$\mathrm{SE} = \frac{\sum \left(N_s\left(S_k, Q_k\right) + qN_c\left(S_k, Q_k\right)\right)}{\sum S_k}, \tag{8}$$

where $0 \le q < 1$ specifies the probability that one tag is identified in a collision slot due to the capture effect.

When the tag population is less than 300, the system efficiencies of the EPC Gen-2 protocol with different capture effects, where $q$ varies from 0 to 90%, are shown in Figure 4. When the tag population is more than 300 and less than 3000, the system efficiencies of the protocol are shown in Figure 5.

Figures 4 and 5 indicate that capture effect can improve the system efficiency of the EPC Gen-2 protocol effectively. Although occasionally, the system efficiency of the protocol in the air interface with small value of capture effect may exceed that with larger value of capture effect, but, in general, as the capture effect $q$ increases, the system efficiency of the protocol also increases.

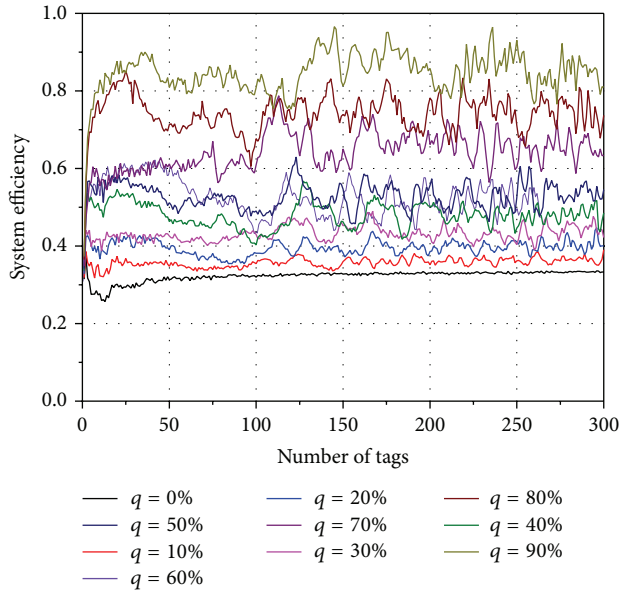



Figure 4: The system efficiency of the EPC Gen-2 protocol with different capture effects when tag population is less than 300.

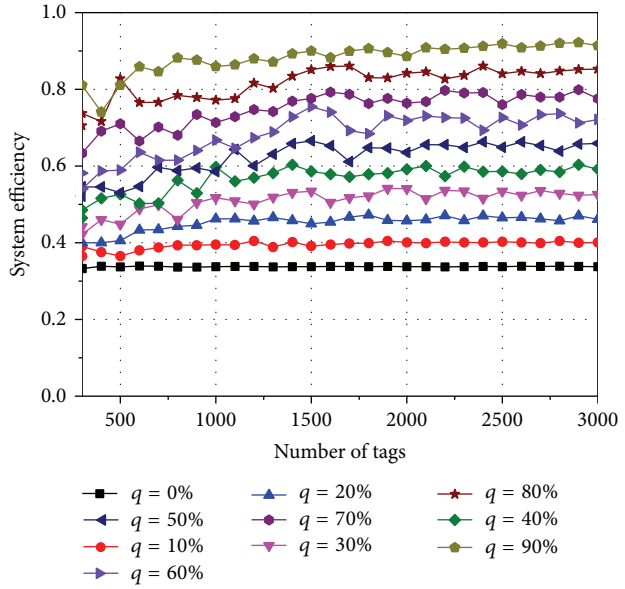



Figure 5: The system efficiency of the EPC Gen-2 protocol with different capture effects when tag population is more than 300 and less than 3000.

We can also see that when capture effect $q$ increases 10%, the system efficiency of the EPC Gen-2 protocol increases about 5%. For example, for the identification of 100 and 1000 tags, the system efficiencies of the protocol with different capture effect $q$ are depicted in Figure 6.

From Figure 6, it can be observed that for the air interface with capture effect, as more tags are within the vicinity of the reader, the system efficiency of the EPC Gen-2 protocol will also increase.

For the air interface with capture effect $q = 1$, which means that in tag responding slot, one tag can always be identified; we can always set the value of $Q$ adopted in the EPC Gen-2 protocol to 0. In such case, the system efficiency of the protocol will be 1.

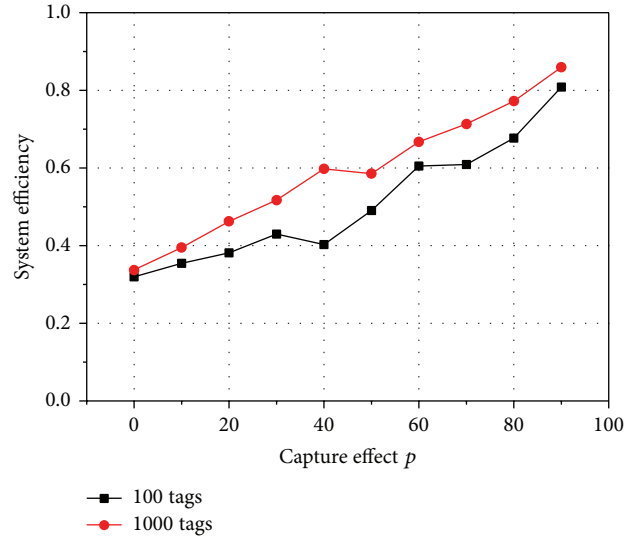



Figure 6: The system efficiency of the EPC Gen-2 protocol with different capture effects for the identification of 100 and 1000 tags.

*4.3. The Noise Air Interface.* In the noise air interface, a data slot where there is only one tag responding may be regarded as a collision slot, since that the waveforms from the tag may be affected by the noisy signals from the environment and the reader cannot decode the signal correctly.

For the noisy air interface, the system efficiency of the protocol can be calculated as

$$\mathrm{SE} = \frac{\sum \left( (1-p)\, N_s \left( S_k, Q_k \right) \right)}{\sum S_k}, \tag{9}$$

where $p > 0$ specifies the probability that the reader views a single reply slot as a collision slot.

In the simulations, set the value of $p$ in the range of $[0.0, 0.8]$. The system efficiencies of the EPC Gen-2 protocol with noise effect in the identification of 1 to 300 and 300 to 3000 tags are shown in Figures 7 and 8.

Figures 7 and 8 indicate that, in the noise air interface, when there are only a few tags, the EPC Gen-2 protocol performs much drastically and unstably. But when there are a lot of tags, the protocol start to perform stably. As we can observe, as the noise effect increases, the system efficiency of the protocol decreases.

For the identification of 100 and 1000 tags, the effect of the noise on the system efficiency of the protocol is shown in Figure 9.

We can see that for a large number of tags, the system efficiency of the protocol degrades linearly as the value of noise effect increases.

*4.4. The Effect of Capture and Noise in the Air Interface.* In this subsection, we want to examine the effect of capture

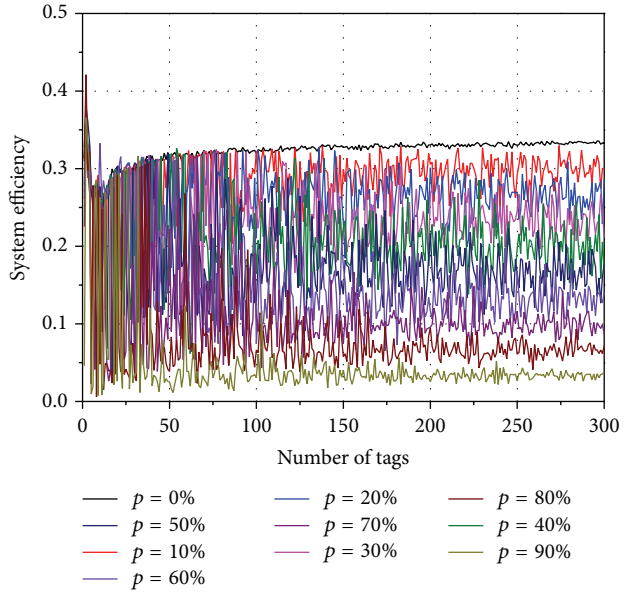



Figure 7: The system efficiency of the EPC Gen-2 protocol in air interface with different noise effects when tag population is less than 300.

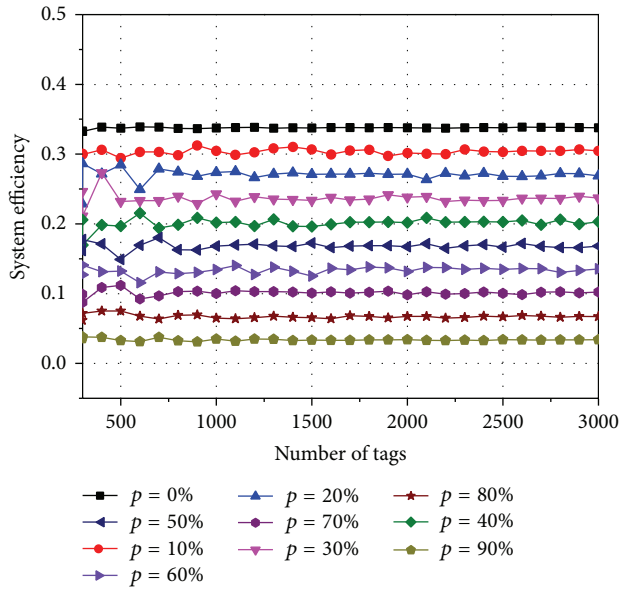



Figure 8: The system efficiency of the EPC Gen-2 protocol in air interface with different noise effects when tag population is more than 300 and less than 3000.

and noise that occurred in the air interface on the system efficiency of the EPC Gen-2 protocol. That is, to determine which effect influence more significantly on the performance of the protocol. So, in this subsection, we set $p$, $q$ in $[0, 0.8]$, $p = q$ and abandon the extreme cases.

The system efficiencies of the EPC Gen-2 protocol in such cases for the identification of 1 to 300 and 300 to 3000 tags are shown in Figures 10 and 11, respectively.

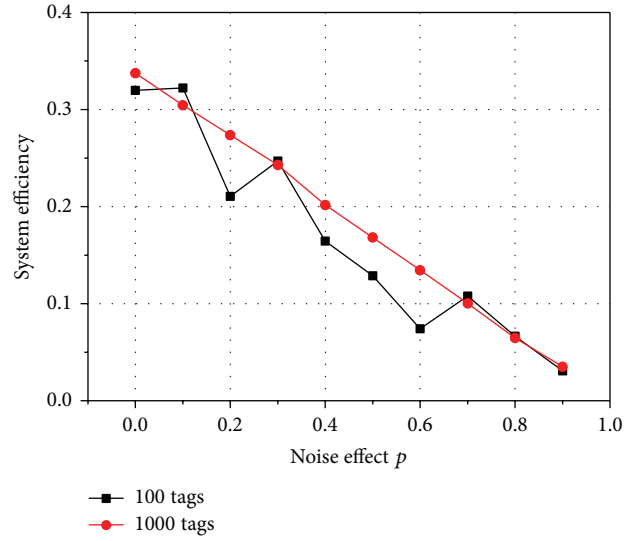



Figure 9: The system efficiency of the EPC Gen-2 protocol with different noise effects for the identification of 100 and 1000 tags.

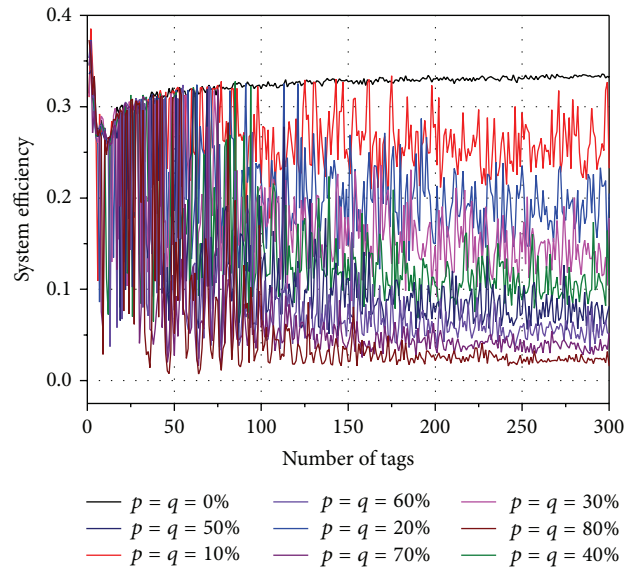



Figure 10: The system efficiency of the EPC Gen-2 protocol in air interface with equal capture and noise effects when tag population is less than 300.

Especially, for the identification of 100 and 1000 tags using the EPC Gen-2 protocol with same noise and capture effects, the system efficiencies of the protocol are shown in Figure 12.

Figure 10 indicates that the noise in the air interface leads to the unstable system efficiency of the protocol when there are only a few tags. But as tag population increases, the protocol starts to perform stably.

Figures 10, 11, and 12 also reveal that the noise plays a more significant role on the performance of the EPC Gen-2 protocol than the capture effect; it may overwhelm the capture effect as its value increases.

*4.5. Discussion.* From the above subsections, we can conclude that the performance of the EPC Gen-2 protocol is

seriously affected by the capture and noise effects. Although the capture effect can improve the system efficiency of the EPC Gen-2 protocol, but the noise effect can degrade the performance of the protocol significantly.

Besides, we can also conclude that in the air interface with serious noise effect, the performance of the protocol is unacceptable, no matter capture effect exists or not. The performance deterioration may be caused by the following two reasons.

Firstly, due to noise, the reader cannot distinguish the single reply slot with the collision slot and cannot read the responding tag. So, the responding tag will select another data slot in the next frame to respond and may cause more tag collisions.

Secondly, the $Q$-adjustment algorithm may also be a reason. Since the protocol may mistakenly take the successful slot as a collision slot, the value of $Q_{fp}$ will increase by $C$. In such case, with more probability, the protocol will ask the reader to issue the *QueryAdjust* command to update the value of $Q$ to $Q + 1$, and more data slots will be consumed in the following frame.

These suggest that for real UHF RFID system deployed in a complex environment, further researches should be taken to optimize the EPC Gen-2 protocol, for example, using biased $Q$-adjustment algorithm to update the value of $Q_{fp}$ with a pair of distinct values, $C_{\text{idle}}$ and $C_{\text{resp}}$, when an idle or a responding slot is encountered, such as the work presented in [22].

## 5. Conclusion and Future Researches

UHF RFID system plays an important role in the upcoming "Internet of Things," but tag collision prevents the universal adoption of the system technically. Although the EPC Gen-2 protocol has been widely accepted to resolve the tag collision occurred in the UHF RFID system, but its performance is seldom evaluated in error prone air interface.

In this paper, a model is established to evaluate the system efficiency of the EPC Gen-2 protocol in error prone air interface, and numeric simulations are performed to evaluate the system efficiency of the protocol in air interface with different capture and noise effects. It is revealed that the noise in the air interface can deteriorate the performance of the protocol significantly.

A lot of research is needed to be taken in the future, for example, to examine the performance of the protocol in real error prone air interface, to optimize the performance of the protocol in air interface with noise and capture effects, and so forth.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The research of this paper is sponsored jointly by the Program for Science & Technology Innovation Talents in Universities of Henan Province under Grant no. 2011HASTIT020, the Program for Innovative Research Team in Science and Technology in Universities of Henan Province under Grant no. 13IRTSTHN023, and the National Major Science and Technology Project under Grant no. 2014ZX03005001. The authors appreciate the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

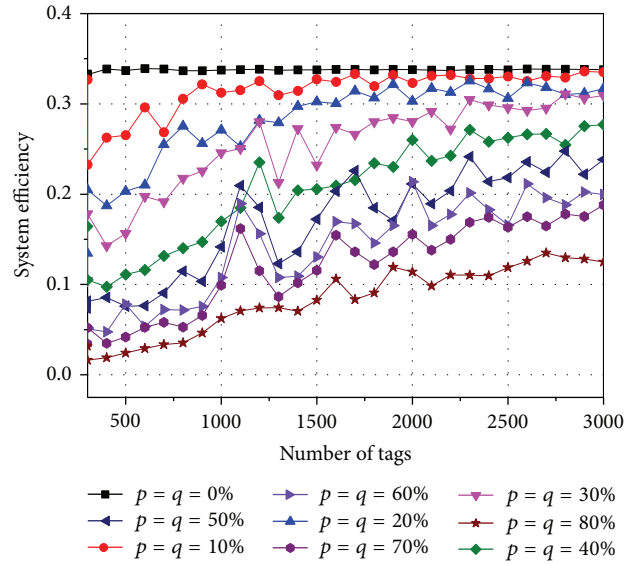



Figure 11: The system efficiency of the EPC Gen-2 protocol in air interface with equal capture and noise effects when tag population is more than 300 and less than 3000.

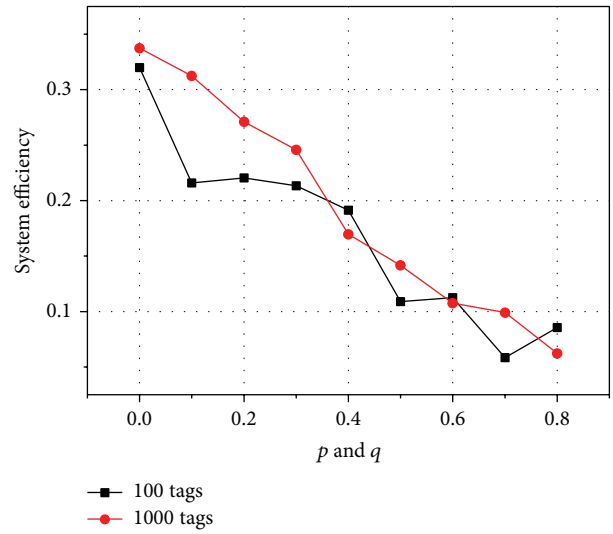



Figure 12: The system efficiency of the EPC Gen-2 protocol with same noise and capture effects for the identification of 100 and 1000 tags.

## References

[1] D. Engels, J. Foley, J. Waldrop, S. Sarma, and D. Brock, "The networked physical world: an automated identification architecture," in *Proceeding of the 2nd IEEE Workshop on Internet Applications*, pp. 76–77, 2001.

[2] E. Welbourne, L. Battle, G. Cole et al., "Building the internet of things using RFID: the RFID ecosystem experience," *IEEE Internet Computing*, vol. 13, no. 3, pp. 48–55, 2009.

[3] V. Stanford, "Pervasive computing goes the last hundred feet with RFID systems," *IEEE Pervasive Computing*, vol. 2, no. 2, pp. 9–14, 2003.

[4] G. Roussos and V. Kostakos, "rfid in pervasive computing: state-of-the-art and outlook," *Pervasive and Mobile Computing*, vol. 5, no. 1, pp. 110–131, 2009.

[5] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, John Wiley & Sons, New York, NY, USA, 3rd edition, 2010.

[6] E. Ilie-Zudor, Z. Kemény, F. van Blommestein, L. Monostori, and A. van der Meulen, "A survey of applications and requirements of unique identification systems and RFID techniques," *Computers in Industry*, vol. 62, no. 3, pp. 227–252, 2011.

[7] D. K. Klair, K.-W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 3, pp. 400–421, 2010.

[8] S. R. Jeffery, M. J. Franklin, and M. Garofalakis, "An adaptive RFID middleware for supporting metaphysical data independence," *VLDB Journal*, vol. 17, no. 2, pp. 265–289, 2008.

[9] T. La Porta, G. Maselli, and C. Petrioli, "Anticollision protocols for single-reader rfid systems: temporal analysis and optimization," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 267–279, 2010.

[10] L. Zhu and T.-S. P. Yum, "A critical survey and analysis of RFID anti-collision mechanisms," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 214–221, 2011.

[11] EPCglobal, "Epc radio-frequency identitiy protocols class-1 generation-2 uhf rfid protocol for communication at 860 mhz-960 mhz," version 1. 2. 0. 2008.

[12] C. Wang, M. Daneshmand, K. Sohraby, and B. Li, "Performance analysis of RFID Generation-2 protocol," *IEEE Transactions on Wireless Communications*, vol. 8, no. 5, pp. 2592–2601, 2009.

[13] C. Wang, B. Li, M. Daneshmand, K. Sohraby, and R. Jana, "On object identification reliability using RFID," *Mobile Networks and Applications*, vol. 16, no. 1, pp. 71–80, 2011.

[14] Y. Maguire and R. Pappu, "An optimal Q-algorithm for the ISO 18000-6C RFID protocol," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 16–28, 2009.

[15] S. R. Aroor and D. D. Deavours, "Evaluation of the state of passive uhf rfid: an experimental approach," *IEEE Systems Journal*, vol. 1, no. 2, pp. 168–176, 2007.

[16] M. Buettner and D. Wetherall, "An empirical study of UHF RFID performance," in *Proceedings of the 14th Annual International Conference on Mobile Computing and Networking*, pp. 223–234, September 2008.

[17] L. Zhu and T.-S. P. Yum, "The optimal reading strategy for EPC Gen-2 RFID anti-collision systems," *IEEE Transactions on Communications*, vol. 58, no. 9, pp. 2725–2733, 2010.

[18] L. Zhu and T.-S. P. Yum, "Optimal framed aloha based anti-collision algorithms for RFID systems," *IEEE Transactions on Communications*, vol. 58, no. 12, pp. 3583–3592, 2010.

[19] X.-Q. Yan, J. Bai, Y. Xu, and B. Li, "An optimized schema to improve the time efficiency of the EPG Gen-2 protocol," in *Proceedings of the International Conference on and 4th International Conference on Cyber, Physical and Social Computing, Internet of Things (iThings/CPSCom)*, pp. 123–126, October 2011.

[20] X. Fan, I. Song, K. Chang et al., "Gen2-based tag anti-collision algorithms using Chebyshev's inequality and adjustable frame size," *ETRI Journal*, vol. 30, no. 5, pp. 653–662, 2008.

[21] W. Su, N. Alchazidis, and T. T. Ha, "Multiple RFID tags access algorithm," *IEEE Transactions on Mobile Computing*, vol. 9, no. 2, pp. 174–187, 2010.

[22] D. Lee, K. Kim, and W. Lee, "Q+-algorithm: an enhanced rfidtag collision arbitration algorithm," in *Proceedings of the 4th International Conference of Ubiquitous Intelligence and Computing*, pp. 23–32, Springer, 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 134787, 15 pages
http://dx.doi.org/10.1155/2014/134787

*Research Article*

# Adaptive Duty-Cycling to Enhance Topology Control Schemes in Wireless Sensor Networks

**Myungsu Cha,[1] Mihui Kim,[2] Dongsoo S. Kim,[3] and Hyunseung Choo[4]**

[1] *Mobile Communications Business of Samsung Electronics, 416 Maetan-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do 442-742, Republic of Korea*
[2] *Department of Computer & Web Information Engineering, Computer System Institute, Hankyong National University, 327 Jungangro, Anseong-si, Gyeonggi-do 456-749, Republic of Korea*
[3] *Department of Electrical and Computer Engineering, Indiana University-Purdue University Indianapolis, 723 West Michigan Street, SL160, Indianapolis, IN 46202, USA*
[4] *College of Information and Communication Engineering, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 440-746, Republic of Korea*

Correspondence should be addressed to Hyunseung Choo; choo@ece.skku.ac.kr

Received 5 July 2013; Accepted 9 December 2013; Published 12 February 2014

Academic Editor: Chang Wu Yu

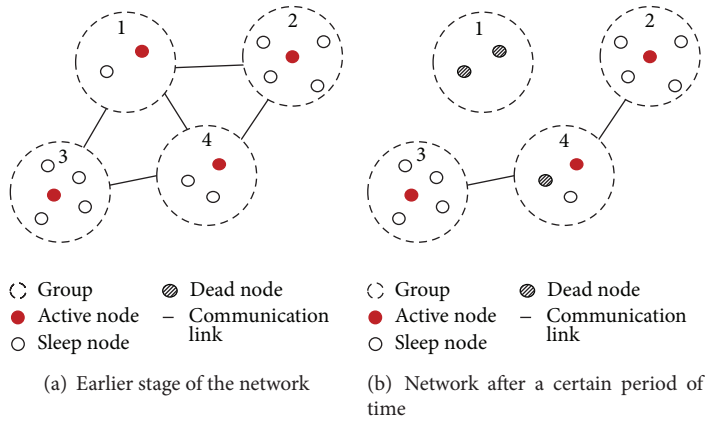


To prolong the network lifetime, various scheduling approaches that schedule wireless devices of nodes to switch between active and sleep states have been studied. Topology control schemes are one of the scheduling approaches that can extend the network lifetime and reduce the additional communication delays at the same time. However, they do not guarantee that all nodes have the same lifetime. They reduce the network coverage and prevent seamless communications. This paper proposes an adaptive duty-cycling (ADC) scheme to solve the unbalanced energy consumption generated from the conventional topology control schemes. Our scheme can be applied as a subprocess of them and enable well-balanced energy consumption among all nodes by applying a different duty-cycle to each group based on group size. Therefore, ADC scheme reduces the coverage reduction and maintains the communication delay as a constant throughout the network lifetime. Simulation results show that our scheme extends the network lifetime by at least 25%. This paper also proposes $t$-ADC scheme. It can be more effectively applied to various environments by adjusting the duty-cycle determined by the ADC scheme which is based on the network traffic amount. We show that $t$-ADC scheme prolongs the lifetime up to 17% compared to ADC scheme in a low traffic network.

## 1. Introduction

The wireless sensor networks are composed of a large number of low powered battery nodes needed to operate in an unattended status for a long duration. It is important to conserve the energy in activities of the nodes in order to keep the nodes alive and to make them run for a long period with limited energy capacity [1]. Reducing the power consumption in communication is the most effective way to extend the lifetime of nodes, as wireless communication uses the majority of the energy among the operations of the nodes [2]. Two strategies are generally used to reduce the energy consumption in wireless communication. One is to adjust the enough transmission power to reach the receiver node. The other one is to periodically schedule the nodes to switch from active mode to sleep mode to save energy during idle listening time.

Several approaches have been proposed to prolong the network lifetime by minimizing the transmission power of the nodes [3–5]. The major energy consumption of wireless sensor networks, however, is caused by the idle listening state but not by packet reception and transmission in a dense network or under light traffic [6] and reducing idle listening time is the most effective way to extend the lifetime. In sleep/wakeup protocols [7, 8], nodes follow a periodic cycle of sleep/active mode without considering the connectivity of the network. This approach can conserve energy by reducing the idle listening time. However, it can cause an additional




(a) Earlier stage of the network

(b) Network after a certain period of time

Figure 1: Problem of topology control schemes.

data transmission delay in WSNs, as the intermediate nodes have to wait for the nodes at the next hop to wake up for receiving the data. In some WSNs applications requiring real-time communication, the transmission delay is one of the most important criteria to evaluate the success of the network. Several topology control approaches [9–12] have been proposed for the reduction of the energy consumption without incurring the critical data transmission delay in a dense sensor network.

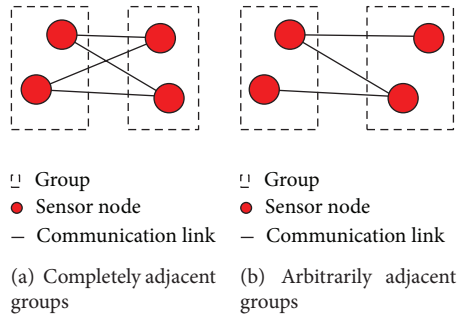
The basic idea of topology control schemes is to divide the nodes into groups, and regardless of the selecting of a node from a group, all the active nodes can form a connected backbone network. Then, only the nodes that have the highest energy are selected in each group to take charge of an active radio mode. The other nodes can conserve their energy by putting their wireless devices into sleep status. These approaches can also reduce further transmission delay, as the network has been connected. However, they cannot guarantee well-balanced energy consumption among all the nodes because they cannot guarantee that each group consists of the same number of nodes. For example, there is a network as shown in Figure 1(a). The network consists of several groups created by topology control scheme. Nodes in the different groups use their energy unevenly at each other because each group consists of a different number of nodes. As a result, nodes in group 1 run out of energy faster compared to the nodes in the other groups. The region of group 1 will be a hole and then the network will be disconnected as shown in Figure 1(b). It must be ensured that all nodes have an equal lifetime, as they are equally vital to maintain the networks [13].

We propose an adaptive duty-cycling (ADC) scheme to enable well-balanced energy consumption among all the nodes, while minimizing any extra overhead in topology control schemes. The ADC scheme can be applied as a subprocess of the conventional topology control schemes. Regardless of the number of nodes present in a group, the traditional method of topological control scheme operates such that one node is in active radio mode from each group. That is, each group has a 100% duty-cycle in which one node is selected from each group to operate in an active radio mode at all times. But we will consider the number of nodes in each group. The group that has a large number of nodes will have a high duty-cycle, whereas the group that has a small number of nodes will have a low duty-cycle. In this paper, we also propose $t$-ADC scheme. The $t$-ADC scheme can be more effectively applied to various environments by adjusting the duty-cycle, which is determined by ADC scheme, based on the network traffic amount. This approach achieves that ADC scheme can balance the energy consumption between all nodes. To evaluate our scheme, we make in-depth simulations and show that ADC scheme extends the network lifetime by at least 25% compared to the conventional ones. Also, ADC scheme keeps the transmission delay as a constant during the network lifetime. It ensures reliable transmission delay. Moreover, ADC scheme uses distributed information and is executed only once at the earlier stage of the network. It has a low overhead and high scalability. At low traffic network, $t$-ADC scheme which is an extension scheme of ADC scheme extends the network lifetime by about 17% compared to ADC scheme.

The rest of the paper is organized as follows. We show the preliminary study about ADC scheme in Section 2. In Section 3, we describe our proposed ADC algorithm. We show the simulation results in Section 4. The last Section concludes the paper.

## 2. Preliminaries

*2.1. Assumptions and Definitions.* This paper has the following assumptions. Each node consists of the communication module and sensing module. The energy consumption during sensing is negligible compared to wireless communication [14, 15]. In the wireless communication, the main energy consumption is used for idle listening, instead of packet transmission and reception. The duty cycling is an important mechanism for the reduction of energy consumption in sensor networks. The duty cycling technique saves energy by switching the wireless communication interface of each node between awake and sleeping, while the nodes always keep the sensing devices in an active status. If some data is sensed by the nodes whose wireless interface is in the sleep status,




(a) Completely adjacent groups (b) Arbitrarily adjacent groups

FIGURE 2: The kind of neighborhood groups.

then these nodes can turn on their wireless components temporarily for the transmission of data and can return to sleep status after the completion of the data transmission [12].

We use the following terms. The size of the group is the number of nodes in a group. The maximum group size is the size of the group with the largest number of nodes in the network. In a network, every pair of adjacent groups is either a completely or arbitrarily adjacent group. As shown in Figure 2(a), two completely adjacent groups ensure the connection between these two groups whichever wireless interfaces of any one node from each group operate in active mode. As shown in Figure 2(b), two arbitrarily adjacent groups cannot guarantee the connection. It depends on the active nodes that are selected from each group.

*2.2. Related Work.* Without causing a serious data transmission delay in dense sensor networks, topology control schemes are to reduce the energy consumption. Topology control schemes use redundant nodes [14]. Their purpose is to keep the connected backbone network by putting the minimum number of nodes in the active mode. In these schemes, the network lifetime can be prolonged because sleep nodes conserve their energy. Any extra communication data transmission delay can be reduced as these schemes guarantee the connectivity of the entire network. These topology control schemes are generally classified as location driven protocol and connection driven protocol [14]. Typical examples of location driven and connection driven protocols are the geographical adaptive fidelity (GAF) [9, 10] and connectivity based partitioning approach (CPA) [11, 12] schemes, respectively.

As shown in Figure 3, GAF [9, 10] divides the sensing area, where the nodes are distributed into multiple equal-size squared cells and each is with a side length of $R/\sqrt{5}$, where $R$ is a radio transmission radius. A group is organized by the nodes in the same cell. After the grouping process, each group will select one node as the active status. This guarantees that any two nodes in the neighborhood cells are connected together, as their distance is within $R$. The entire network connectivity can be guaranteed by activating only one node from each group. GAF can ensure that all nodes in the same group have a similar lifetime by alternatively activating a node from the group. However, this scheme is not suitable for the real environment since it uses an ideal

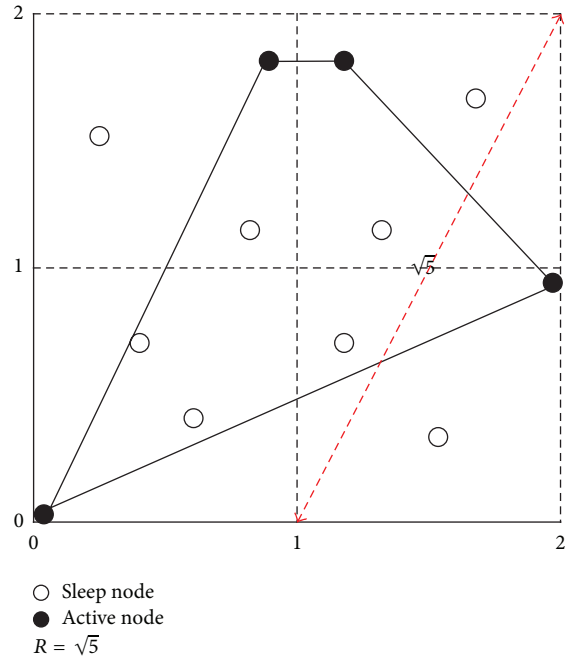



FIGURE 3: Geographical adaptive fidelity (GAF).

radio propagation model. Moreover, each group has four completely adjacent groups regardless of the traffic amount in the network, because the GAF divides the sensing area based on square grid structure.

CPA has overcome these limitations. CPA divides the nodes based on their connectivity, instead of their physical position. As shown in Figure 4(a), each node forms a group at the beginning of the grouping process. Then, the pairs of two completely adjacent groups merge into one group (Figure 4(b)). This process operates continually until the preset number of completely adjacent groups cannot be guaranteed [11, 12]. Similarly in GAF, the connectivity of the entire network can be guaranteed by activating only one node from each group. By periodically changing the active nodes, CPA also guarantees that all the nodes in the same group have a similar lifetime. This implies that all nodes in the network are equally important. However, this scheme has a problem

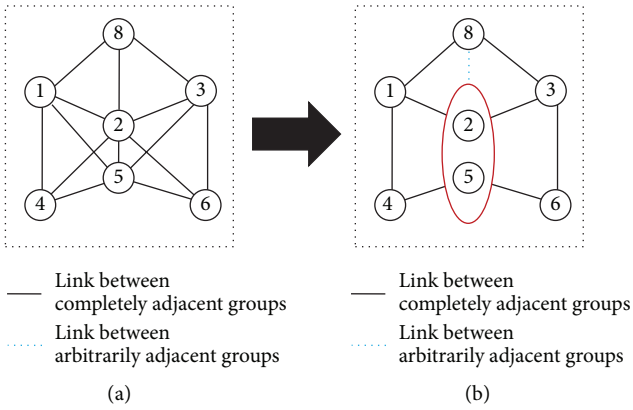



FIGURE 4: Connectivity based partitioning approach (CPA). (a) Two groups 2 and 5 that are to be merged. (b) Two groups have been merged.

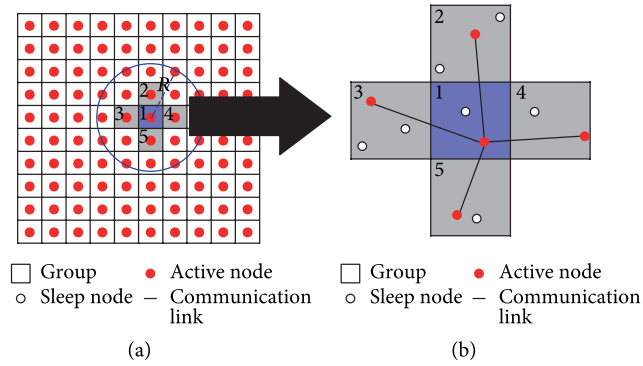



FIGURE 5: Active sensor nodes in GAF. (a) Any one node operates in the active mode from each group. (b) Completely adjacent groups for group 1.

that nodes in groups with fewer nodes consume their energy relatively fast.

## 3. Proposed Scheme

We propose the ADC scheme (see Algorithm 1) which can be applied to the conventional topology control schemes in order to guarantee that all nodes in the network have a similar lifetime. Both GAF and CPA approaches only ensure that all nodes in same group have similar lifetime. However, no one can guarantee the same number of nodes in every group. This causes an unbalance in the energy consumption between different groups. Nodes in small size groups run out of energy faster compared to the nodes in large size groups. Although nodes in large size groups may have sufficient energy remaining, the network cannot guarantee seamless data transmission if the surrounding nodes run out of energy. Thus, the network lifetime in these conventional schemes reduces. Moreover, these schemes create completely adjacent groups to guarantee the connectivity of the entire network, whichever any one node operates in active mode from each group. In the process of creating these completely adjacent groups, a greater number of arbitrarily adjacent groups are formed rather than the completely adjacent groups, as shown in Figure 5. If only one node operates in the active mode from each group, then it is likely that the node will communicate with the nodes in arbitrarily adjacent groups and as well as in the completely adjacent groups. We need a way to enable all the nodes in the network to have a similar lifetime by using the communicable nodes.

We propose an adaptive duty-cycling (ADC) approach. ADC can be applied as a subprocess of the conventional topology control and ensure that all the nodes in the network have a similar lifetime. The basic idea of the proposed scheme is to group the nodes by using the topology control and then apply adaptive duty-cycle, depending on the group size. By this, the proposed scheme ensures that all the nodes in the network have a similar lifetime. For instance, GAF and CPA schemes always require any one node to operate in the active mode from each group. In ADC, if the maximum group size is 10, then the group formed by 10 nodes always operates any one node in an active mode during 10 unit times. Conversely, the group formed by 2 nodes operates any one node in the active mode for only 2 unit times out of 10 unit times. It means a 20% duty-cycle; all nodes in the group remain in sleep mode for the remaining 8 unit times. Our proposed scheme can guarantee balanced energy consumption among all nodes in the network using the adaptive duty-cycle based on group sizes. If ADC scheme is applied to the conventional schemes, such as GAF and CPA, then the connectivity of the network will decrease at the earlier stage of the network. However, the connectivity of the network deteriorates rapidly over time in the conventional schemes, because nodes in small size groups rapidly run out of energy. If ADC scheme is applied to the conventional schemes, then the initial connectivity of the network can be maintained during the lifetime. It can guarantee seamless communication.

*3.1. ADC Approach.* ADC scheme consists of two stages to guarantee that all the nodes in the network have a similar lifetime, as shown in Figure 6. One is to estimate the maximum group size based on the distributed information and the other one is designated to determine the duty-cycle for each group. Each group determines its own duty-cycle by comparing their group size with the maximum group size estimated in the first stage. With these two stages, ADC scheme assures balanced energy consumption among all nodes in the network.

At the first stage, the size of the group is broadcasted within two-hop distance by the head node of the group [16]. After receiving the group size information of other groups, each head node estimates the average group size ($\mu$) in the network in a dispersive manner. Then, the standard deviation ($\sigma$) can be calculated by using

$$\sigma = \sqrt{\sum_{k=1}^{n} \frac{(x_k - \mu)^2}{(n-1)}}, \quad (1)$$

where $x_k$ is the size of the groups and $n$ is the number of information for the group sizes received from the other groups within two-hop distance. If $n$ is greater than 30, the

```
(1) for all hi ∈ H do    /*hi: header nodes of each group,
                           H: set of header nodes */
(2)     broadcast own group size within 2 hop
(3) end for
(4) for all hi ∈ H do
(5)     calculate average group size (μ) using information received from the other headers
(6)     calculate standard deviation (σ) using information received from the other headers
(7)     estimate maximum group size using μ and σ
(8)     determine own duty cycle by comparing own group size with the maximum group size
(9) end for
```

Algorithm 1: Adaptive duty-cycling algorithm.

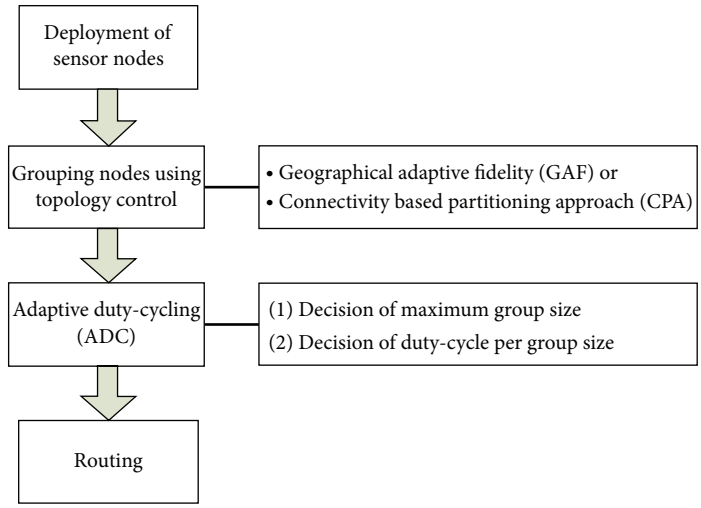



Figure 6: Overview of the proposed scheme.

group sizes known to the head node will follow the normal distribution, in accordance with the central limit theorem. According to the normal distribution, 99% of all the data are located within $\mu \pm 2.58\sigma$. We can estimate the maximum group size ($M_{\text{size}}$), after calculating the standard deviation on the group sizes as follows:

$$M_{\text{size}} = \mu + 2.58\sigma. \tag{2}$$

In the second stage, each group determines its duty-cycle based on the maximum group size estimated in the first stage. Here, the duty-cycle for each group is calculated based on the assumption that the largest group operates with 100% duty-cycle by using (3) to enable all the nodes to have similar activation times as follows:

$$S_{\text{Duty-Cycle}} = \frac{100 \cdot S_{\text{size}}}{M_{\text{size}}}, \tag{3}$$

where $S_{\text{Duty-Cycle}}$ is the duty-cycle, $S_{\text{size}}$ is the group size, and $M_{\text{size}}$ is the maximum group size. By using (3), each group operates with a different duty-cycle based on the proportion of its group size to the maximum group size.

In some prior topology control approaches, such as GAF and CPA, each group has to turn on at least one node all the time. As no one can guarantee the same number of nodes in every group, the nodes in small size groups exhaust their energy faster compared to the nodes in larger groups. As a result, nodes in the network will have a different lifetime. By applying the adaptive duty-cycle depending on the group size, the proposed scheme ensures that all the nodes in the network have a similar lifetime. On the other hand, our proposed scheme maintains similar communication paths throughout the network lifetime. Our approach can be modularly applied to the conventional topology control schemes and is conducted only once. Therefore, it has the advantage of low overhead.

*3.2. Application Method and Example.* Our proposed scheme can be applied as an additional subprocess of the conventional topology control. In order to demonstrate how the approach is applied to conventional schemes, we group the nodes by using the conventional topology control approaches, GAF and CPA. 500 nodes, each with a transmission radius of $\sqrt{5}$, are deployed in the network with an area of $10 \times 10$. Table 1 shows the result of grouping the nodes based on GAF and CPA [12].

The mindeg value represents the number of completely adjacent groups in Table 1. As shown in Table 1, the number

Table 1: Partitions of GAF and CPA.

| Partition approach | Number of groups | Average group size | Standard deviation on group size |
|---|---|---|---|
| CPA (mindeg = 2) | 71 | 7 | 0.92 |
| CPA (mindeg = 3) | 84 | 6 | 0.77 |
| CPA (mindeg = 4) | 91 | 5.5 | 0.79 |
| GAF | 100 | 5 | 0.73 |
| CPA (mindeg = 5) | 106 | 4.7 | 0.67 |
| CPA (mindeg = 6) | 116 | 4.3 | 0.63 |

Table 2: Analysis on distribution of group size.

| Distribution of group size | Group size | | | | |
|---|---|---|---|---|---|
| | 3 | 4 | 5 | 6 | 7 |
| Number of groups | 3 | 14 | 66 | 14 | 3 |

of groups, average group size, and the standard deviation on group sizes depend on the mindeg value. But neither GAF nor CPA can ensure that the groups have the same size. In order to demonstrate the differences between the group sizes in the conventional approaches, we can estimate the results shown in Table 2 by analyzing the GAF data from Table 1.

After grouping the nodes by using GAF or CPA, next step is to decide on the maximum group size in order to apply the proposed approach. In order to calculate the maximum group size, the head node of each group broadcasts its size within two-hop distance. Then, each head node dispersively calculates the average group size and the standard deviation on the group sizes based on the information received from other head nodes. After calculating the average group size and the standard deviation on the group sizes, each head node estimates the maximum group size by substituting these values in (2). If every head node is aware of the size information on at least 30 groups, the values estimated using the central limit theorem can be said to be reliable. In order to demonstrate this, this paper uses the data in Table 1 [12]. According to Table 1, all the nodes are divided into 100 groups. In other words, 100 head nodes exist on a network with the size of $10 \times 10$, and if we use (4), then we can identify the number of other head nodes located within the transmission range of any one head node. Consider

$$\text{The size of field} : n = \pi \cdot R^2 : X, \tag{4}$$

$$X = \frac{n \cdot \pi \cdot R^2}{\text{The size of field}}. \tag{5}$$

Here, the size of the field is $10 \times 10$. The number of head nodes, $n$, is 100. The radio transmission radius of each node, $R$, is $\sqrt{5}$, and $X$ is the number of head nodes located within one-hop distance. If we substitute the above figures in (5), then, we will see that there are approximately 15 other head nodes within one-hop distance of each header node. Hence, if each head node broadcasts its size within two-hop distance, then each head node will obtain the information on at least 30 groups.

In real life, radio transmission radius of nodes is not in the form of unit disks but is rather distorted [17, 18]. Hence, the proposed approach should be applied to not only ideal but also real-life radio propagation models. As GAF is not valid in real-life radio propagation model, the proposed approach is applied to CPA to show that our scheme can be applied to real-life radio propagation model. In [12], the CPA creates more groups to work well in the real-life radio propagation model. Hence, as the radio propagation model in the real life tends to have small radio transmission radius compared to ideal ones, more groups will be created. As a result, even though the proposed approach is applied in real-life radio propagation model, the number of head nodes within one-hop distance is similar to ideal radio propagation model.

According to the maximum group size estimated by using the information received in a dispersive manner, the duty-cycle for each group is calculated by using (3). This is based on the assumption that the largest group operates with 100% duty-cycle. We use the values in Table 2 in order to demonstrate that each group has a different duty-cycle depending on its size: as maximum group size is 7, a group formed of 3 nodes will operate with a 43% duty-cycle. On the other hand, a group whose size is 7 will operate with a 100% duty-cycle as it has an equal size to the largest group. In other words, this group always has the active state node until all nodes in the group run out of energy. On the other hand, a group that contains 3 nodes will allow only one node in the active state for the 43% of the network lifetime, and all the other nodes will remain in the sleep mode for the rest of the lifetime, to reduce energy consumption. As a result, the proposed approach can ensure that all the nodes in the network have a similar lifetime.

Even though all the nodes are guaranteed an equal lifetime, the proposed approach will not be applicable if the data cannot be transmitted smoothly. The proposed approach must guarantee smooth data transmission. GAF provides 4 completely adjacent groups, and CPA ensures as many completely adjacent groups as the user needs. In order to guarantee the number of completely adjacent groups, more

number of arbitrarily adjacent groups is created. If one node of each group operates in an active state, it is likely to communicate to the nodes that are in its arbitrarily adjacent groups as well as those in the completely adjacent groups. As shown in Figure 5, we can demonstrate this, by using Table 1 and (5). If one node of each group operates in an active state in GAF scheme, there will be about 15 other active nodes within one-hop distances. If we apply the ADC approach on Table 2, then the entire network has a duty-cycle of approximately 70%. If we apply this duty-cycle to Table 1 and (5), then it can be concluded that there are about 10 other active nodes within one-hop distances of each active node. Although routing paths reduce when the proposed scheme is applied to the conventional topology control approaches, the proposed approach ensures smooth and stable communication.

*3.3. Analysis for Necessity of Proposed Scheme.* All the nodes in the WSNs are equally important to minimize the coverage reduction. The proposed scheme which ensures a similar lifetime of all nodes is essential in terms of coverage. This section explains the validity and necessity of the proposed scheme in terms of data transmission by comparing the wireless sensor networks to real life.

In the sensor networks, the basic role of nodes is to sense the information and transmit it via either one hop or multihops towards the destination. These sensor networks can be compared to one of the large transport company. The information sensed by each node is a load that must be delivered to a destination by using a vehicle (wireless communication). The roads that are used to deliver the load by vehicles are the nodes and the lifetime of nodes can be compared to the lifetime of a road. Only if the state of roads (nodes) is in the active mode, then vehicles move. However, a major energy of nodes (roads) is usually used by listening among the communication processes, instead of packet reception and transmission [6]. Reducing the time spent in listening to each road is the most effective way to extend the lifetime of the roads.

As shown Figure 7, the number of active roads in ADC is less than the number of active roads in GAF. It is the same reason stated for Sections 3.1 and 3.2. In order to deliver loads towards a destination, the vehicle on any road will move to other active roads that are located closer to the destination compared to its current location and are located within a distance $R$ from its current location. Here, at the earlier stage of the network, GAF scheme delivers load to the destination faster than ADC scheme because GAF scheme compared to the ADC scheme has more active roads that can be used by a vehicle. However, the lifetime of roads in groups with fewer roads ends at relatively faster rates in GAF scheme. The number of available roads reduces fast over time compared to the earlier stage of the network. After a certain period, the topology of GAF and ADC schemes is changed as shown in Figure 8. Here, the proposed scheme delivers load to the destination faster than GAF scheme because the proposed scheme has more active roads than GAF scheme.

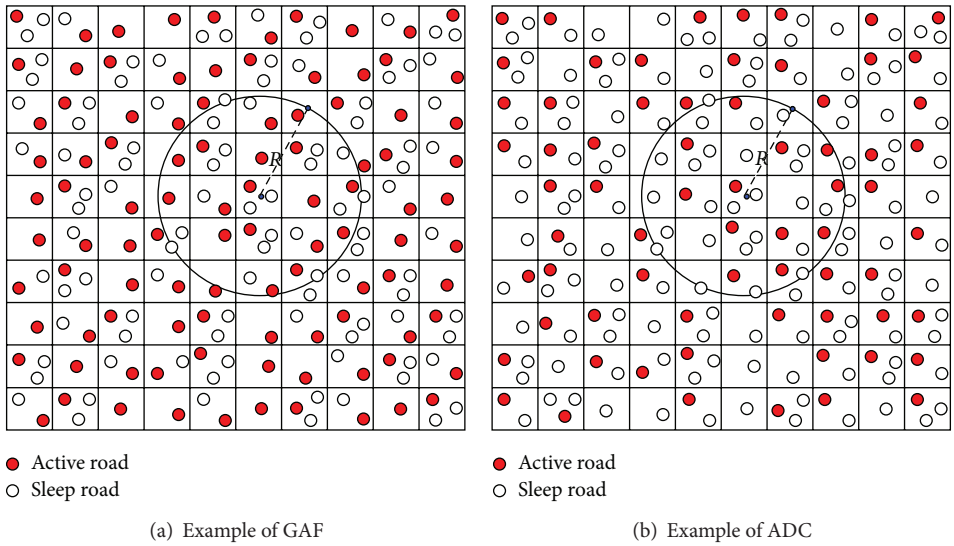
In other words, the number of available roads reduces fast over time in the GAF scheme, whereas ADC scheme always maintains similar number of available roads. It can be explained by using analogy as follows. In GAF scheme, if the transport company works for 15 days, it will use 10 roads from day 1 to day 5, 7~5 roads from day 6 to day 10, and 2~0 roads from day 11 to day 15 to deliver the load. As a result, although the transport company delivers the load rapidly at the earlier days, delivery speed of the load will decrease as days go by. On the other hand with GAF, in the proposed scheme, the transport company can guarantee a constant delivery speed on every day, as the company always uses 7 roads to deliver the load during the 15 days. Therefore, proposed scheme is more efficient compared to the previous scheme in terms of reliable data transmission. This explanation can be applied equally even if GAF scheme is changed by using the CPA scheme.

*3.4. $t$-ADC Scheme.* In the previous ADC scheme, we have only focused on guaranteeing whether all nodes have the same lifetime or not. As a result, it can achieve the similar lifetime for all nodes in the network and reliable data transmission. However, ADC scheme does not consider the various network environments (e.g., traffic amount) where the nodes are deployed. In other words, in ADC scheme, each group decides its duty-cycle depending on just the group size without the consideration of the network environments. Although, by using this way we can guarantee similar lifetime for all nodes in the network; it lacks adaptability in accordance with the various network environments. If ADC scheme can be changed depending on network environment, where nodes are deployed, it will be a more efficient scheme. For example, if each group reduces its duty-cycle in low traffic network, network lifetime will be significantly extended, whereas additional end-to-end communication delay is not big.

In this subsection, we propose the $t$-ADC scheme which is based on ADC scheme and it can adjust flexibly the duty-cycle of each group based on network environment factor (i.e., traffic amount) where a wireless sensor network is applied. In $t$-ADC scheme, each group decides its duty-cycle by using

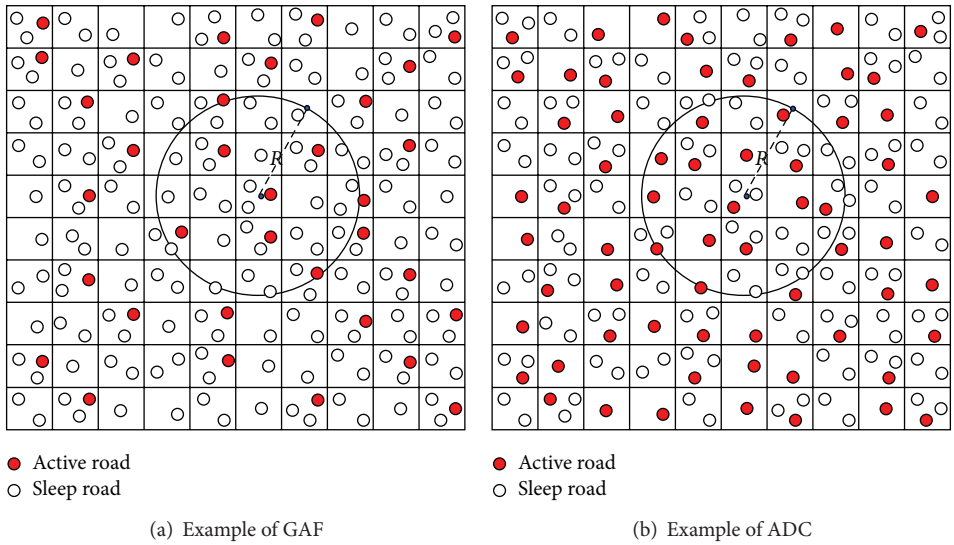
$$S_{\text{Duty-Cycle}} = t\frac{100 \cdot S_{\text{size}}}{M_{\text{size}}}, \tag{6}$$

where $S_{\text{Duty-Cycle}}$ is the duty-cycle, $S_{\text{size}}$ is the group size, and $M_{\text{size}}$ is the maximum group size. It is similar to ADC scheme. The traffic constant ($t$) is determined by the administrator and it is based on the traffic conditions of the application when a sensor network is deployed. Here, by multiplying the traffic constant ($t$), $t$-ADC schemes can be more flexibly applied to various environments. The maximum value of the traffic constant is 1 and the lower the traffic amount of the network is, the lower the value of traffic constant is. In order to show the effectiveness of $t$-ADC scheme, this paper simulates the ADC and $t$-ADC schemes while changing the range of the traffic constant value from 0.8 to 1 at environment of [12]. The result of simulation shows that $t$-ADC scheme extends the network lifetime by about 17% compared to ADC




(a) Example of GAF (b) Example of ADC

Figure 7: Active roads in GAF and ADC schemes at the earlier stage of the network.



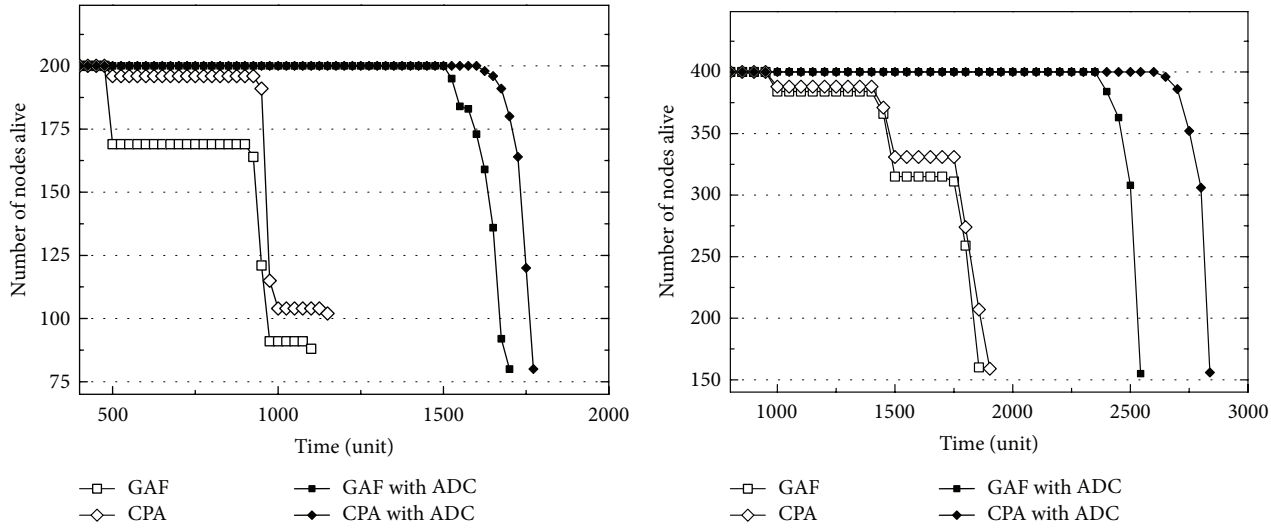

(a) Example of GAF (b) Example of ADC

Figure 8: Active roads in GAF and ADC schemes, after a certain period.

scheme without the incurring critical data transmission delay regardless of network environments.

## 4. Performance Evaluation

*4.1. Simulation Environment.* We implement the simulation by using Java to analyze the performance of the proposed scheme. We use two ways to assure the reliability of the simulation. First, we conduct the simulation in accordance with the same environment in [12], and we obtain the same trends from the graphs that represent the network lifetime. Second, all the experiments are repeated 1,000 times. As shown in Table 3, default simulation environments are as follows. We uniformly deploy 500 nodes, each with a transmission radius of $\sqrt{5}$, in the network with an area of $10 \times 10$. The ratios of the amount of energy used in the transmitting, receiving, and listening status are $1.7 : 1.2 : 1$, respectively [15]. The initial energy of each node is set to 500. So, each node will remain alive during 500 unit times in the listening mode. The simulations are conducted based on the assumption that the energy consumption is uniform over all the nodes, regardless of the location of the sink node by using the mobility-assisted approaches in order to evaluate the algorithm more accurately [19, 20]. Twenty pairs of source and sink nodes are randomly selected in each time slice [12]. Load balanced short path routing [21] is used as the routing protocol, and the network lifetime is defined as the time a certain ratio of the nodes runs out of energy [22, 23]. As shown in Table 3, some simulation environments are changed in each subsection. In this case, we state the new simulation parameters at the beginning of each subsection.




(a) The number of nodes is 200

(b) The number of nodes is 400

Figure 9: Comparison of network lifetime.

*4.2. Network Lifetime.* The number of dead nodes that run out of energy over time, when 200 and 400 nodes are deployed in the network, is illustrated by using Figures 9(a) and 9(b), respectively. The horizontal axis represents the unit time (i.e., the time elapsed from the beginning of the simulation) and the vertical axis shows the number of living nodes. As shown in Figure 9(a), the first dead node appears around 500 unit times in the GAF and CPA schemes, and the number of living nodes decreases in the pattern of steps over the next 700 unit times. On the other hand, when the proposed scheme is applied to GAF and CPA, the first dead node appears at about 1,500 unit times, and the number of the living nodes decreases in a steep curved pattern. Due to the different size of a group in the network, nodes consume very different amounts of energy in conventional approaches; some nodes run out of energy very quickly. Thus, the number of living nodes decreases in a pattern of steps over a long time. In contrast, the proposed scheme applies the adaptive duty-cycle depending on the group sizes and this enables all the nodes to have a similar lifetime. Therefore, the number of living nodes sharply decreases over a short time. Moreover, when the proposed approach is adapted, some groups convert all their nodes to sleep mode. Thus they can conserve the energy and can cause an extension of the overall network lifetime for conventional approaches.

Figure 10 shows the number of dead nodes over time, after 500 nodes are deployed in the network. The experiments are implemented under different number of completely adjacent groups (=mindeg). In GAF and CPA, the first dead node occurs around 1,000 unit times, and the number of living nodes decreases following a step pattern over the next 1,500 unit times. Moreover, by applying the proposed scheme to the GAF and CPA, the first time when a dead node appears is about 3,000 unit times. During the next 200 unit times, the number of living nodes decreases quickly which is similar to a steep curved pattern. By applying the proposed approach we can balance the lifetime of nodes and increase the network lifetime. Besides, regardless of the node density, we can ensure an even lifetime for nodes in the network by applying the proposed approach to GAF and CPA. We can also minimize the coverage reduction of the network that results due to the energy depletion of nodes. The experiment results show that the network lifetime is improved by up to 25% compared to the conventional approach.

Figure 11 shows the times when certain ratios of the nodes in a network have consumed all their energy in

Table 3: Summary about simulation method.

| | |
|---|---|
| Compared schemes | GAF, CPA, GAF with ADC, and CPA with ADC |
| Performance metrics | Lifetime<br>Transmission delay<br>Balanced energy consumption |
| Parameters | |
| Default | Number of nodes: 500<br>Network size: 10 × 10<br>Mindeg in case of CPA: 4<br>Traffic: 20 pairs of sink and source nodes (each time slice) |
| Section 4.2 | Number of nodes: 200, 400, and 500<br>Mindeg in case of CPA: 3, 4, and 5 |
| Section 4.5 | Number of nodes: 125, 500, 1125, 2000, and 3125<br>Network size: 5 × 5, 10 × 10, 15 × 15, 20 × 20, and 25 × 25 |
| Section 4.6 | Traffic: 5 pairs of sink and source nodes (each time slice) |
| Routing | Load balanced short path routing |
| Other environments | Same as environment in CPA |

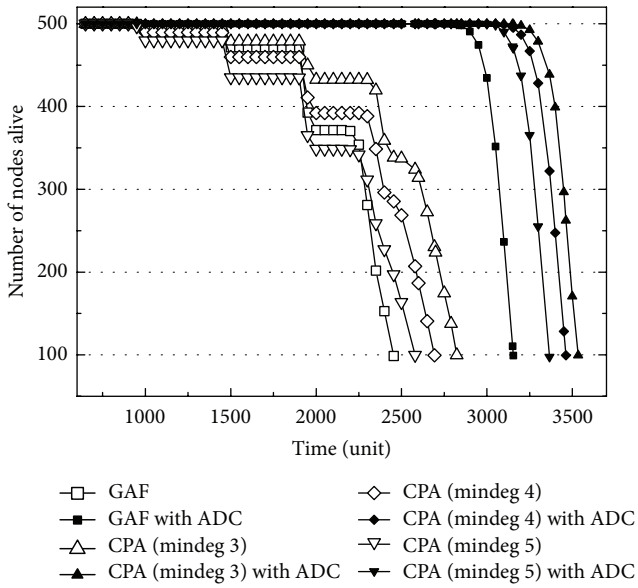



Figure 10: Comparison of network lifetime under different mindeg (the number of nodes is 500).

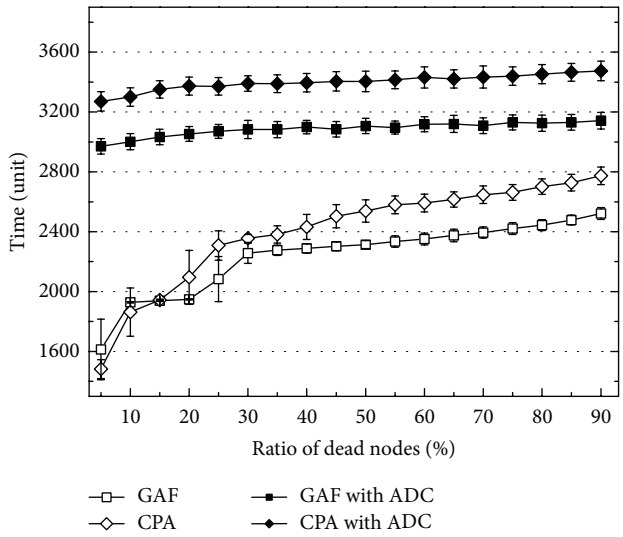



Figure 11: Comparison of time for certain ratios of dead nodes (the number of nodes is 500).

the conventional approaches and applying the proposed scheme to these. The horizontal axis represents the percentage of the dead nodes in the network. The vertical axis represents the unit time, which is the time elapsed from the beginning of the simulation. When GAF and CPA are applied, 25 nodes (5%) consume all their energy at 1,600 and 1,500 unit times, respectively, after the simulation has begun. 450 nodes (90%) run out of their energy at 2,300 and 2,700 unit times, respectively. It can be deduced that it takes about 1,000 unit times until the next 85% of the entire nodes consume all their energy after the first 5% do. On the contrary, if the proposed approach is applied in addition to GAF and CPA, 25 nodes (5%) of the entire nodes consume all their energy at 3,000 and

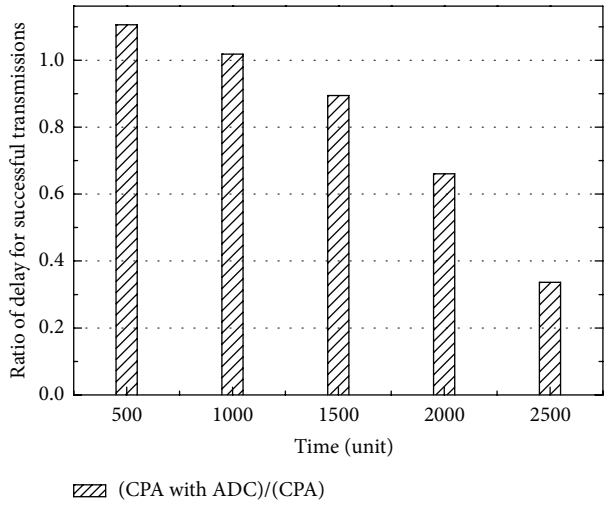



Figure 12: Ratio of transmission delay.

3,250 unit times, respectively, after the start of the simulation. 450 nodes (90%) consume all their energy at 3,150 and 3,400 unit times. In other words, the differences in the lifetime of nodes in a network are as much as 1,000 unit times when the conventional approaches are used, whereas the difference in the lifetime between the nodes is significantly reduced to only 150 unit times when the proposed approach is applied. This is due to the proposed approach which uses adaptive duty-cycle. By using such an approach, the ADC ensures even lifetime for all the nodes and it can reduce the coverage reduction of a network that results from the dead nodes.

*4.3. Transmission Delay.* The transmission delay ratios for successful data of CPA and CPA with ADC are shown in Figure 12. The transmission delay for successful data is the time taken for successfully transmitting the data from the source node to the sink node. This delay ratio (vertical axis of Figure 12) can be calculated by using

$$\text{Ratio of delay} = \frac{\text{Delay for successful transmissions}_{\text{CPA with ADC}}}{\text{Delay for successful transmissions}_{\text{CPA}}}. \quad (7)$$

If ADC is applied to CPA, the value of the delay ratio is greater than 1 at the earlier stage of the network (500 unit times). This result implies that ADC yields longer data transmission delay. It is because some groups in the network may convert all their nodes to sleep mode and the corresponding communication paths for those groups cannot be used. On the other hand, conventional approaches can use all the communication paths by sequentially activating a node for each group. However, the value of the delay ratio gets smaller over time and it becomes less than 1 after 1,000 unit times, and the value of the delay ratio decreases over time. This result implies that the ADC has shorter data transmission delay compared to CPA, and the difference in their data transmission delay time increases over time. These results

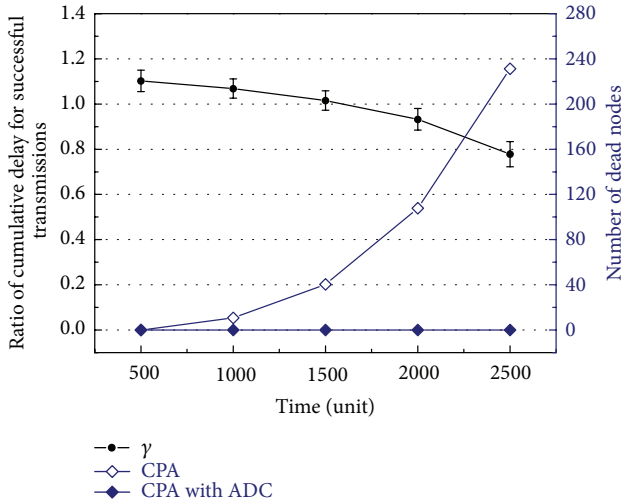



Figure 13: Ratio of cumulative transmission delay and the number of dead nodes ($\gamma$ is the ratio of delay).

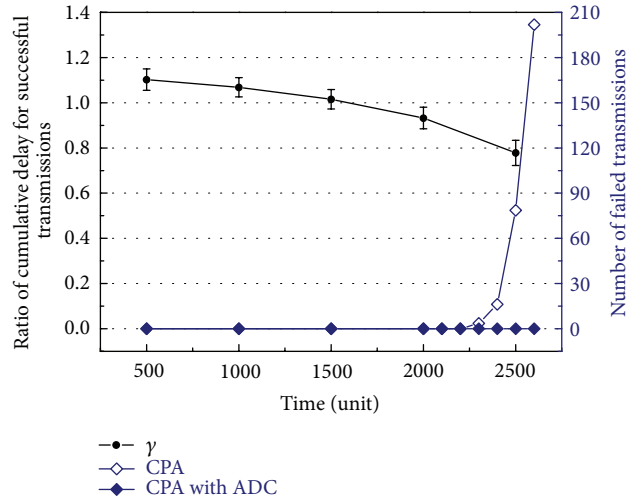



Figure 14: Ratio of cumulative transmission delay and the number of failed transmissions ($\gamma$ is the ratio of delay).

can be explained as follows. In the conventional schemes, the energy consumption of the nodes is uneven. Due to the uneven energy consumption of nodes, some nodes may deplete its energy much more quickly compared to others. In contrast, by applying the proposed scheme to conventional schemes, similar communication paths can be maintained throughout the network lifetime as the proposed scheme guarantees a similar lifetime of nodes. As a result, our scheme can guarantee consistent and reliable data transmission.

Figure 13 shows the relationship between the ratio of the cumulative data transmission delay for successful data and the number of dead nodes, under CPA and CPA with ADC. The left vertical axis is the ratio between CPA and CPA with ADC for the cumulative time taken for the successful transmissions to the sink node. The right vertical axis represents the number of dead nodes. The number of dead nodes increases over time in CPA, while dead nodes do not appear until 2,500 unit times when the proposed scheme is applied. The data transmission delay of CPA with ADC is larger compared to that of CPA at the earlier stage of the network. The reason is similar to that mentioned for Figure 12. The ratio of the cumulative data transmission delay for CPA and CPA with ADC is less than 1 at a later stage than that of the noncumulative delay illustrated in Figure 12. This is due to the cumulative data transmission that accrues time for the data transmission delay of the previous successful dates. However, there is virtually no difference between the data transmission delays for the two approaches when the standard deviation for them is taken into account. Moreover, we can observe that after a certain period of time, the transmission delay is shorter under CPA with ADC. As shown in Figure 13, it can be explained as follows. In CPA, the energy consumption of the nodes in the network is uneven and this causes some nodes to exhaust much more quickly compared to others. On the other hand, all the nodes in the network remain alive in CPA with ADC.

Figure 14 shows the relationship between the ratio of cumulative communication delay and the number of failed transmissions, under CPA and CPA with ADC, respectively. The left vertical axis represents the ratio between CPA and CPA with ADC for the cumulative time taken of successful communications to the sink nodes. The right vertical axis shows the number of failed transmissions. In order to compare the network connectivity of CPA and CPA with ADC, we assume that there is no collision occurring in the MAC (Media Access Control) layer such as [24]. At the earlier stage of the network, no failure appears in CPA and in CPA with ADC. However, after 2,300 unit times, the number of failed transmissions increases rapidly under the CPA approach. This can be explained from the following. If a conventional approach is applied, the energy consumption of the nodes is uneven and it causes some nodes to exhaust much more quickly compared to others. On the other hand, our proposed scheme ensures that all the nodes have similar lifetime. Our scheme maintains good network coverage and it also guarantees reliable communication by maintaining an even level of data transmission delay.

*4.4. Balanced Energy Consumption of Nodes.* Through the entropy method, Section 4.4 shows that the proposed approach enables nodes in a network to have similar lifetime and to consume energy at even rates. "Entropy" is a criterion of randomness, and it is known that all substances on earth follow entropy. In other words, they tend to disperse evenly rather than gathering in one space. The entropy value for the given data set $S$ is calculated by using (8) [25] as follows:

$$\text{Entropy}(S) = -\sum_{i=1}^{k} p_i \cdot \ln(p_i), \tag{8}$$

$$p_i = \frac{\text{freq}(C_i, S)}{|S|}. \tag{9}$$

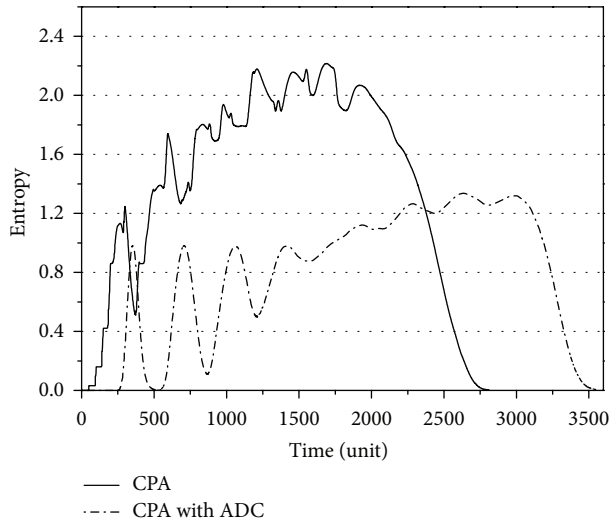



Figure 15: Entropy of CPA and CPA with ADC.

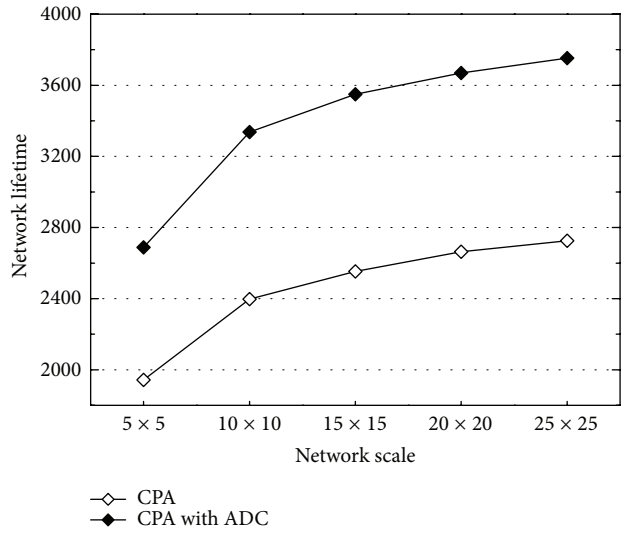



Figure 16: Network lifetime under different network scales (node density = 5).

In (8) and (9), $S$ is the set of given data and $C$ is the set of classes from $C_1$ to $C_k$. The class implies a set of data that meets a specific character. freq $(C_i, S)$ is the number of data that belong to $C_i$ in the data set $S$, and $|S|$ is the number of data in $S$. As the data given in (8) evenly disperses to many classes, the entropy values will also increase.

Figure 15 shows the entropy under CPA and CPA with ADC, respectively, when 500 nodes are deployed in a network. The initial energy of each node is set to 500 and this implies that the node will remain alive for 500 unit times in the listening state. The vertical axis represents the entropy, which is the energy distribution of the nodes and it can be calculated by using (8) and (9). In these equations, class $C_i$ is divided into different energy levels, each with a range of 50 (i.e., 0~49, 50~99, etc.). Therefore, a total of 10 classes are formed, from $C_1$ to $C_{10}$. $S$ represents the energy of each

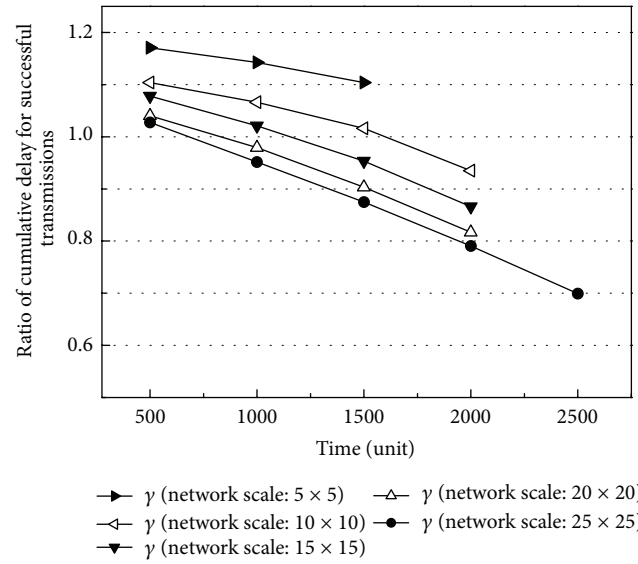



Figure 17: Ratio of cumulative transmission delay under different network scales ($\gamma$ is ratio of the delay; node density = 5).

node, and $|S|$ is the total number of nodes in the network which is 500. Figure 15 shows that the entropy value can be as high as 2.2 when CPA is applied. Meanwhile the value of entropy is lower than 1.3 when CPA with ADC is applied. As higher entropy values imply uneven energy distribution among entire nodes in a network, our proposed scheme is more effective in ensuring that all nodes use a similar amount of energy during their lifetime.

*4.5. Network Lifetime and Transmission Delay under Different Network Scales.* Figure 16 shows the network lifetime when we fix the node density to five nodes per square unit and run CPA and CPA with ADC on networks of different scales. The horizontal axis represents the network scale and the vertical axis represents the network lifetime. Here, network time is defined as the time when the first transmission fail occurs. As shown in Figure 16, when the network scale is 5 × 5, the lifetime of CPA and CPA with ADC is about 1,900 and 2,600 unit times, respectively. The proposed scheme prolongs the network lifetime by about 37% compared to the conventional one in 5 × 5 network size. Figure 16 also shows that, when the network size is large, the network lifetime of both CPA and CPA with ADC is longer because if the network size increases in same traffic amount condition, ratio of nodes, which are directly concerned with traffic, reduces. Here, an important aspect to be noted is that the proposed scheme always extends the network lifetime by about 38% compared to the conventional one regardless of the network size.

Figure 17 shows the ratio of the cumulative data transmission delay for successful data of CPA and CPA with ADC. Simulation parameters of Figure 17 are equal to Figure 16. The vertical axis represents the ratio between CPA and CPA with ADC for the cumulative time taken for successful transmissions to the sink node. This ratio value is greater than 1 at 500 unit times, the earlier stage of the network regardless of the network size. It implies that the data transmission delay

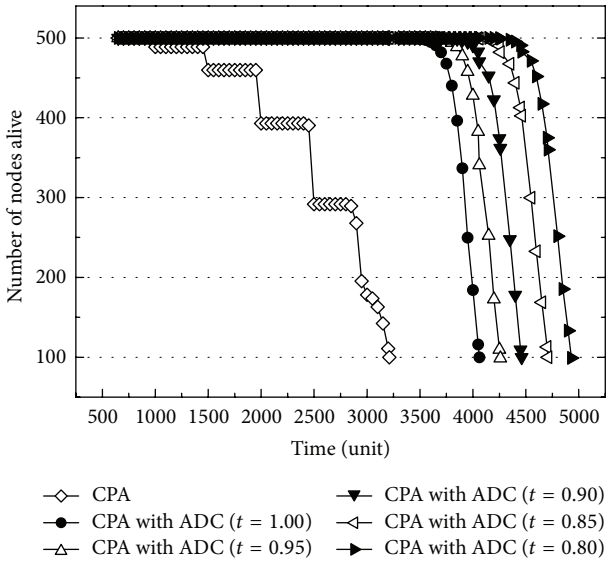



Figure 18: Network lifetime under different $t$ ($t$ is traffic constant in (6)).

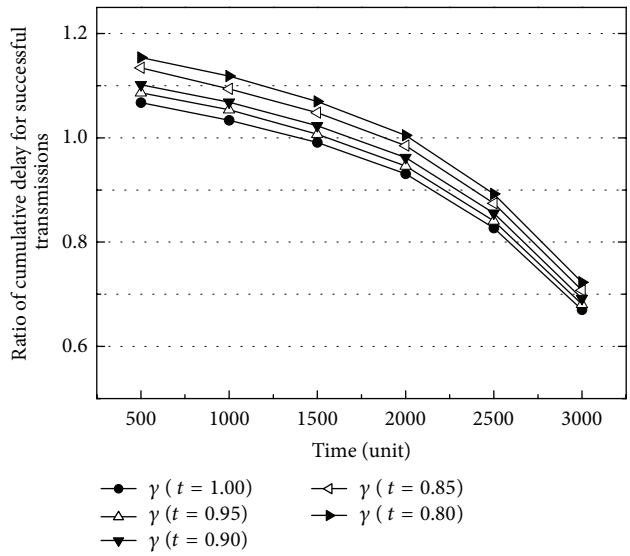



Figure 19: Ratio of cumulative transmission delay under different $t$ ($\gamma$ is the ratio of delay; $t$ is the traffic constant in (6)).

is longer when the ADC is applied to CPA, and the reason is similar to that described in Section 4.3. The ratio value of a network with $25 \times 25$ scale is the lowest at 500 unit times. The reason is that when the network size increases in the same traffic amount the effect of the traffic amount will decrease. Here, an important aspect to be noted is that the ratio value decreases over time regardless of the network scale. It implies that ADC causes shorter data transmission delay compared to CPA. We have already explained the reason for this in Section 4.3. Here, the data transmission delay of CPA and CPA with ADC cannot be compared in a network with a network scale $5 \times 5$, after 2,000 unit times, because network lifetime is over before 2,000 unit times. For a similar reason, we cannot compare the data transmission delay of CPA and CPA with ADC in a network with a network scale from $5 \times 5$ to $20 \times 20$, after 3,000 unit times.

*4.6. $t$-ADC Scheme.* This subsection evaluates the $t$-ADC scheme which we discuss in Section 3.4. Here, to assume low traffic network, five pairs of source and sink nodes are randomly selected in each time slice. In Figure 18, when the traffic constant is 1, ADC scheme is the same as $t$-ADC scheme. When the traffic constant is 0.9, duty-cycle of each group in $t$-ADC scheme is 90% of that of ADC scheme.

Figure 18 shows the network lifetime of ADC and $t$-ADC scheme when the traffic constant is changed from 0.8 to 1. The horizontal axis represents the unit time (i.e., the time elapsed from the beginning of simulation) and the vertical axis shows the number of living nodes. Similarly in Section 4.2, first dead node occurs at around 1,000 unit times in CPA scheme, and the number of living nodes decreases in the pattern of steps over time. On the other hand, when $t$-ADC scheme is applied to CPA, similar lifetime is guaranteed for all the nodes. Figure 18 also shows that, as the traffic constant is low, network lifetime is extended and this is because the smaller the traffic constant is, the bigger the sleeping ratio will be. When the traffic constant is 0.8, the network lifetime is extended by about 17% compared to the traffic constant which is 1.

Figure 19 shows the ratio of the cumulative delay for successful transmissions of CPA and CPA with $t$-ADC. Simulation parameters of Figure 19 are equal to Figure 18. The vertical axis represents the ratio between CPA and CPA with $t$-ADC for the cumulative time that is taken for the successful transmissions to the sink node. Similarly, in Section 4.3, the ratio value is greater than 1 at 500 unit times, the earlier stage of the network regardless of traffic constant. It implies that the data transmission delay is longer when proposed scheme is applied to CPA. Moreover, the smaller the traffic constant is, the larger the ratio value is because, as traffic constant is small, the number of active nodes decreases. However, an important aspect to be noted is that the ratio value decreases over time regardless of the traffic constant, and the proposed scheme eventually guarantees faster data transmission delay compared to CPA.

In this subsection, we compare the lifetime and data transmission delay of CPA and CPA with $t$-ADC while the traffic constant is varied. Here, we show that, when the traffic constant is 0.8, ratio of data transmission delay is increased by about 7% compared to traffic constant which is 1, whereas network lifetime is extended by about 17%. It is because network is a low traffic environment (i.e., in low traffic network, even if sleeping ratio of each group is enlarged, the additional data transmission delay is not longer, whereas network lifetime is considerably extended). If we adjust duty-cycle of each group based on the network traffic environment, network lifetime is efficiently extended.

## 5. Conclusions

In this paper, the ADC approach is proposed in order to balance the energy consumption of the nodes. ADC approach

can be used as a supplement scheme for other conventional protocols such as GAF and CPA. Simulation results show that, by applying the proposed scheme to these protocols, the network lifetime is improved by at least 25% and the energy consumption is guaranteed equally among the nodes. And the proposed scheme guarantees more reliable communication compared to other protocols and it has a low overhead because it is executed only once at the earlier stage of the network. Moreover, it uses distributed information so that it enhances the scalability of proposed scheme. We also propose $t$-ADC scheme which makes ADC scheme more flexible to adapt to various environments. Simulation results show that $t$-ADC scheme efficiently extends the network lifetime by adjusting the duty-cycle of each group based on the traffic environments of network where nodes are deployed.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This research was supported in part by MSIP (KEIT) and MOE (NRF), the Korean government, the IT R&D Program (10041244, SmartTV 2.0 Software Platform), and BSRP(NRF-2011-0014020)/PRCP(NRF-2010-0020210), respectively.

## References

[1] D. Ganesan, A. Cerpa, W. Ye, Y. Yu, J. Zhao, and D. Estrin, "Networking issues in wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 799–814, 2004.

[2] B. S. Krishnan, M. Ramaswamy, and N. Alamelu, "Optimising energy∗delay metric for performance enhancement of wireless sensor networks," *International Journal of Engineering Science and Technology*, vol. 2, no. 5, pp. 1289–1297, 2010.

[3] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, vol. 2, pp. 404–413, Tel Aviv, Israel, March 2000.

[4] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang, "Distributed topology control for power efficient operation in multihop wireless ad hoc networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '01)*, vol. 3, pp. 1388–1397, Anchorage, Alaska, USA, April 2001.

[5] N. Li and J. C. Hou, "FLSS: a fault-tolerant topology control algorithm for wireless networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 275–286, Philadelphia, Pa, USA, October 2004.

[6] M. Medidi and Y. Zhou, "Extending lifetime with differential duty cycles in wireless sensor networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 1033–1037, Washington, DC, USA, November 2007.

[7] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '02)*, vol. 3, pp. 1567–1576, New York, NY, USA, June 2002.

[8] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 171–180, Los Angeles, Calif, USA, November 2003.

[9] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 70–84, Rome, Italy, July 2001.

[10] Y. Xu, S. Bian, Y. Mori, J. Heidemann, and D. Estrin, "Topology control protocols to conserve energy in wireless ad hoc networks," Tech. Rep., 2003.

[11] Y. Ding, C. Wang, and L. A. Xiao, "Connectivity based partition approach for node scheduling in sensor networks," in *Proceedings of the 3rd IEEE International Conference on Distributed Computing in Sensor Systems*, Santa Fe, NM, USA, June 2007.

[12] Y. Ding, C. Wang, and L. Xiao, "An adaptive partitioning scheme for sleep scheduling and topology control in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 9, pp. 1352–1365, 2009.

[13] Y. Obashi, H. Chen, H. Mineno, and T. Mizuno, "An energy-aware routing scheme with node relay willingness in wireless sensor networks," *International Journal of Innovative Computing, Information and Control*, vol. 3, no. 3, pp. 565–574, 2007.

[14] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

[15] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 85–96, Rome, Italy, July 2001.

[16] D. L. Trong and H. Choo, "Efficient flooding scheme based on 2-hop backward information in ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 2443–2447, Beijing, China, May 2008.

[17] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Models and solutions for radio irregularity in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 2, pp. 221–262, 2006.

[18] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services (MobiSys '04)*, pp. 125–138, Boston, Mass, USA, June 2004.

[19] J. Luo and J.-P. Hubaux, "Joint mobility and routing for lifetime elongation in wireless sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '05)*, vol. 3, pp. 1735–1746, Miami, Fla, USA, March 2005.

[20] W. Wang, V. Srinivasan, and K.-C. Chua, "Using mobile relays to prolong the lifetime of wireless sensor networks," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MobiCom '05)*, pp. 270–283, Cologne, Germany, September 2005.

[21] Z. Fan, "Prolonging lifetime via mobility and load-balanced routing inwireless sensor networks," in *Proceedings of the*

*23rd IEEE International Symposium on Parallel & Distributed Processing (IPDPS '09)*, pp. 1–6, Rome, Italy, May 2009.

[22] F. Shen, M.-T. Sun, C. Liu, and A. Salazar, "Coverage-aware sleep scheduling for cluster-based sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '09)*, pp. 1–6, Budapest, Hungary, April 2009.

[23] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *IEEE Communications Letters*, vol. 9, no. 11, pp. 976–978, 2005.

[24] Y.-J. Han, M.-W. Park, and T.-M. Chung, "SecDEACH: secure and resilient dynamic clustering protocol preserving data privacy in WSNs," in *Computational Science and Its Applications—ICCSA 2010*, vol. 6018 of *Lecture Notes in Computer Science*, pp. 142–157, Springer, Berlin, Germany, 2010.

[25] L. Feinstein, D. Schnackenberg, and D. Kindred, "Statistical approaches to DDoS attack detection and response," in *Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX '00)*, vol. 1, pp. 303–314, Washington, DC, USA, April 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 649505, 12 pages
http://dx.doi.org/10.1155/2014/649505

*Research Article*

# Web Services Integration Strategy with Minimum Hops Constraint in Wireless Sensor Networks

**Wen Ouyang and Min-Lang Chen**

*Department of Computer Science and Information Engineering, Chung Hua University, 707 Sec. 2, WuFu Road, Hsinchu 300, Taiwan*

Correspondence should be addressed to Wen Ouyang; ouyang@chu.edu.tw

Received 28 June 2013; Revised 30 October 2013; Accepted 31 October 2013; Published 6 January 2014

Academic Editor: Chang Wu Yu



Both wireless sensor networks and service-oriented architectures (SOA) are very popular distributed computing paradigm. Web service is a widely accepted implementation of SOA. However, web service is based on Internet protocols and requires adaptation to suit the restrictions posed by WSNs. The development of Internet of Things has triggered research of web services of WSNs which made the consolidation of these technologies possible. At the same time, web service integration enables the support of more functionalities since many applications rely on not just one web service, but a whole school of them. Thus, how to compose and integrate different web services efficiently to provide complicated services becomes an essential topic. This paper investigates a problem which minimizes the number of hops of web services while integrating these web services to finish a set of tasks. We call this problem Minimum Hops of Service Integration Problem. It is proved that, when there are no precedence relationships between the tasks, the decision problem is NP-complete. This implies that this web services integration problem is NP-hard. For the case when the relationships between the tasks are in linear order, a polynomial-time, optimal web service integration algorithm, using greedy strategy, is provided.

## 1. Introduction

Wireless sensor networks (WSNs) are used to monitor the concerned events through the collaboration between sensor nodes via wireless communication [1–3]. Recently WSN technologies have been recognized as one of the most important technologies that will seriously impact the world. Due to the advancement of the manufacturing technologies, especially on the communication and battery productions, tiny size of sensors can be equipped with the capabilities of sensing, wireless communication, and data processing. These types of sensors can not only sense and detect targets and changes in their environments but also process the collected data and transmit the data back to the data collection center or base station via wireless communication. The users can thus retrieve the status of the environments and develop interesting application accordingly. WSNs are to deploy massive amount of sensors in the sensing region to collect all kinds of environmental information and then pass the information via wireless network to the base station (sink) and then to the back-end user or manager via the Internet or satellites.

WSN has gained much popularity because of the advanced development of microelectromechanical systems (MEMS) and its wide range of monitoring/tracking applications. Applications of WSNs range in diversified areas [3] such as military, environment, health, home, and other business fields. For military applications, sensors can be used for people identification, battlefield monitoring, enemy detection, and so on. Our living environment can be monitored by sensors to detect events such as forest fire, flooding, pollutions, and volcano eruption. Sensors can also be applied to monitor the health conditions and behaviors of people. Intelligent home also relies on sensors to let people monitor and control the devices in our home. Other applications such as manufacturing automation requires robots equipped with sensors to work together to complete sophisticated work. Traffic control and automobile tracking are also popular applications.

Since the sensors may be deployed randomly to the sensing areas, they need to be able to self-organize using specified protocols to form a communication network so that all sensors in the same network can communicate with each other and transmit the collected data back to the base station. Due to the reason that the required amount of deployed sensors is huge, the sensors have to satisfy the requirements of low cost, low energy consumption, and easy deployment. At the same time, they are programmable, equipped with environment-sensing devices, and can be dynamically reorganized.

Several factors can influence the WSN design [1, 2]. One of them is fault tolerance which means the abilities to sustain the WSN functionalities without any interruptions no matter what type of node failure occurs. Another one is scalability which means the sensor density endurance in the WSNs. Developed schemes for WSNs should work no matter how sparse or how dense the network is. The other factors are hardware constraints, sensor network topology, deployment environment, transmission media, and power consumption. Power consumption is actually a big concern for WSNs due to the reachability of these sensors after deployment. Other factors [3] such as the application's design objectives, cost, and system constraints should also be considered.

Due to the advancement of Internet of Things, more complicated WSNs [4] integrated with heterogeneous devices and multidiscipline systems are a trend to be. These systems have to satisfy many performance requirements such as scalability and security. At the same time, software engineering concerns such as requirement evolvement due to product improvement make the adaptability an important factor to consider. There are many software architecture models for WSNs. The architecture model has to be able to satisfy the rigorous requirements for the WSNs. One popular model is to incorporate a software middle layer to the system [5]. To increase the flexibility, service-oriented middleware is proposed and many related issues are discussed [5–11].

Service-oriented computing paradigm which focuses on loose-coupling software architectures has been much investigated due to its highly modularized properties and platform independent features which enhance system flexibility and adaptability. It relies on service-oriented architecture (SOA) concepts to create large-scale software applications whereas, most of the time, a large set of interacting web service-based components are involved. Nowadays, the challenges for information technology departments have moved from internal processes management to efficiently collaborating with new ideas, functionalities, and partners so that the corporation can be competitive enough [12, 13]. The flexibility of SOA makes it right there for this purpose. However, the SOA practices face many challenges and obstacles although many enterprises, such as Intel, Microsoft [14], Oracle [15], IBM [16], and others [17–19], have endorsed and/or adopted the SOA philosophy in their products or internal business processes.

Web services are distributed web applications which identify a popular implementation of SOA with standard models and protocols. This technology involves open standards such as XML, SOAP, and HTTP. The mechanism involves service providers, service requestors, and service brokers. Although both WSN and web services are taking advantages of distributed computing paradigm, their technologies are far different from each other. Web services are based on Internet protocols. WSNs are based on MEMS with all kinds of resource constraints such as energy, computing power, and its very own set of communication protocols and architectures. The development of Internet of Things has brought these two together [20] with a lot of mending to do.

It is obvious that SOA inherits the concept of component-based software development and thus the integration and composition of SOA play an essential role in the development process. Natis [21] and McCormick [22] point out six keys to the success of SOA which include to invest integration infrastructure and to design service systematically. Another one is to anticipate obstacles which foresees the challenges in it. This reveals how challenging it is to make SOA practice a success, and it also indicates that component integration requires quite much attention and plays an essential role when applying SOA into practices. The performance of the software application heavily relies on the way the services are composed or integrated to support the corresponding applications.

Many of the current SOA developments are still focused on the static software application. However, the issues raised when software applications are composed or integrated dynamically have gained more and more popularity. Dynamic business process modeling has been viewed as a way to improve the organization competition. This can be applied to the following three dynamic software application situations.

(1) When for the first time an application is composed, it follows that maybe many web services are out there in the network, waiting to be employed. The choices made in picking up appropriate web service will affect the efficiency of the application greatly.

(2) When fault tolerance strategy is required for critical systems, a back-up plan is always needed. There are always times when a node in the network fails to perform its functions. Some web service providers may want to provide redundant node to avoid out-of-service situations. This is necessary in certain critical situations and systems depending on how critical the functionality is and how nonreplaceable the web service is. In certain situations, when the web services are replaceable, the problem of how we can make a good replacement for this web service is also a topic worth discussion.

(3) When the software allows dynamic composition, for example, dynamic business processes or some game software which may allow game players to develop their own games dynamically with a set of ready-to-use software modules, the software development process always requires software development professionals to engage in the implementation work. The users are contributors of ideas and suggestions in the normal paradigm of software development processes.

More advanced platform of collaborative development allows the users to take part in the development by themselves. The trend can be traced back to many open online services like Google Video and YouTube till Wikipedia which encourages the users to also be the authors to increase the interactions between all the users. The exact same concept can also be applied in the software development. With web services or SOA in mind, the time for this new paradigm to be realized may be near. For example, for game players, they can reuse the existing building blocks to create new games by themselves and even building new game components using existing services available. In this case, when there are many supporting web services, how to choose the existing components more wisely is essential to the performance of the resulting software.

Lin [23] proposed the idea of accountability of web services. Web services own the advantages of platform independency, component reusability, and flexibility. However, this loosely coupled structure also poses potential problem of the quality of service since there is always a concern about the quality of the whole system while openly integrating all the services from different resources and platforms via network. Thus, an in-depth mechanism is required for making sure the service quality will not suffer. The idea is to establish a service accountability supervision scheme to make sure the services, especially from external resources, can maintain performance and stability. Managing business integrity has been considered [24], by the IBM Research Global Technology Outlook, as one of the systems development trends that will significantly impact the information technology in the near future. This involves the management of policy integrity, process integrity, and core entities information integrity throughout the whole enterprise.

Many studies have been focused on service-oriented architecture [13, 25], especially on applying web services. Quite a few of them are discussing how to compose these web services or how to apply these web services. Some studies are investigating the web services' nonfunctional properties, for example, how to make web services secure, how to provide highly reliable web services, and how to evaluate web services' performance.

Following the trend of complicated application requirements and the fast advancement of Internet technologies, the issue with applying web services is not just how to find a single web service which can accomplish one single function. It is essential that we need to satisfy the need of applications with complex requirements. Thus, how to dynamically integrate different web services to satisfy the changing world and achieve better performance is an important problem to solve. Web service composition/integration has been a hot topic. However, the focus has been on the composition models, frameworks, and mechanisms [26] such as how to extend the WSDL language and how to build the QoS model.

This work investigates the web service integration issues, which consider how to choose a set of web services to satisfy the functional requirements of a pool of tasks and also reduce the web service response time. The idea is based on the concern that the communication time used when hopping between web services contributes the most of response time for the most of the time. To reduce the hop count between web services usually means that we can effectively reduce the response time and the network traffic.

We propose a novel problem of web service integration, that is, how to integrate the existing web services in order to satisfy a pool of tasks and, at the same time, to minimize the web service hop count with the preliminary work published in [27]. This problem plays important roles when the response time is an important factor in evaluating the performance of the applications. This paper considers two types of task precedence relationships. The first case concerns that there are no relationships between the tasks. The second one assumes that the tasks are required to be executed in a specified linear order. We also provide a polynomial-time, optimal greedy solution to integrate the web services in order to achieve the goal of minimizing the hop count of web services for the second case.

The rest of the paper is organized as follows. Section 2 of this paper discusses related work of WSNs, service-oriented WSNs, and web services. Section 3 describes the web service integration problem and the proposed integration system architecture. Section 4 illustrates and analyses the two special cases of the problem and we propose a polynomial-time, optimal algorithm for the second case. Detailed data structure demonstrations for an example are provided in Section 5. Section 6 concludes this work.

## 2. Related Work

WSNs are used to monitor concerned events through the collaboration between sensor nodes via wireless communication. They have been widely investigated in many issues since it was adopted. WSNs can be classified as terrestrial, underground, underwater, multimedia, and mobile WSN [3]. Terrestrial WSNs are the typical sensor networks which have been discussed widely. It is usually composed of enormous amount, may be thousands, of inexpensive wireless sensor nodes deployed in a specified area, such as battlefield. The power is limited and thus energy saving is an important issue. Besides, communication protocol is an essential part to consider including data link layer, network layer, and transport layer protocols. Another interesting part is the services which can be provided by WSNs such as localization, coverage, and security. WSNs are usually more vulnerable to attacks and faulty situations possibilities which makes fault-tolerant mechanisms essential when designing and developing WSN-related systems. These mechanisms can be classified as fault-tolerant communications and fault-tolerant data sensing. The purpose is to safeguard the true sensed data to the back end for further processing.

An event of interest can be a single target moving across the sensing area or it may be an event which covers a substantial amount of regions in the sensing area. The job of detecting the former one becomes localization/tracking

of targets and that of the latter is recognized as the event boundary detections. Mobility is an interesting topic to study in WSN systems. A target can be mobile and it usually is. Sensor nodes can be either static or mobile. Events can also be static or mobile (more likely). Mobile WSNs can take the advantages of mobile nodes to fix problems such as connectivity or coverage issues so that the network can be strongly connected or the whole sensing area can be covered to the maximum degree. Mobile anchor nodes which are aware of their own locations can help in the localization for the other normal nodes too. On the other hand, mobile WSN shows problem in routing since the existing links may be broken when the nodes are moving to different locations. Dynamic routing strategies are required to solve this type of issues. For mobile events or dynamically changing events, the detection work becomes more complicated.

The adoption of WSNs has become a complicated task, especially after the Internet of Things has been seriously studied. Much effort was spent in integrating WSN into Internet of Things in the areas such as IP-based WSNs [28], signaling [29], path and coverage problem [30], and security [31]. Due to their wide application fields, WSNs may need to face constant change of applications and rapid advancement of technologies. To comprehend more complicated situations in a flexible and scalable way, middleware for WSN becomes a solution which can be virtual machine based, database, application driven, message oriented [32], and service oriented [6, 7]. The service-oriented middleware layer approach of WSNs is investigated due to the highly abstract mechanism including interface, its modularity of the architecture which are flexible and scalable, and its standard model and protocols. This approach focuses on building WSNs as services to the applications. Delicato et al. [8] proposed a sensor network architecture based on services in which the sink is a service provider to the applications and a service requestor to the sensor nodes whereas a WSDL document is used to describe the services provided by the sink and sensor nodes. TinySOA [9] was proposed as a service-oriented architecture for WSN. Each WSN provides services which can be internal to the WSN itself or external to outside applications. The architecture components are node, gateway, registry, and server. Nodes are basically the sensor nodes with internal service-providing capability. A gateway bridges the WSN with the external application and knows the services for each node and the whole network. The whole infrastructure is stored in the registry which relies on an internal database to manage the whole system. The last component in TinySOA is the server which is a provider of web services whereas each WSN is considered as a separate web service. Another service-oriented middleware framework for programming [10], called USEME, was also provided so that applications can be easily built using the defined API. USEME provides high-level programming model and languages so that much work will be handled by the middleware such as service publication and discovery, event invocation and communication, and group management. Other service-oriented middleware research [5, 11] works in similar way. Since web service is the most common implementation of SOA, this work adopts the terminology of web service to represent SOA.

From the web service point of view, due to the network technology advancement, network bandwidth and speed have grown greatly. Web service technologies have been widely researched, applied, and integrated including how to efficiently manage the web services. Guan et al. [33] proposed a web services management structure named FASWSM trying to take care of this issue. Web services can be plugged into the application server as a web service adapter to reduce the web services management efforts.

Some researches explore the method of combining different web services to satisfy complicated function requests. Cheng et al. [34] proposed an approach to integrate existing web services and implement a web service integration tool for users who are not familiar with programming skills. Users can compose new services by combining the existing service modules. Mohanty et al. [35] used finite state machines to modularize different web services in order to integrate them to new and more complex network services.

Since web service architecture poses open structures to the outside world, security problems are introduced. Many studies investigate them. Tang et al. [36] presented a comprehensive performance evaluation to understand the security issue of web services. Sidharth and Liu [37] presented a framework for enhancing web services security to prevent DoS or DDoS attacks when designing and implementing web services. Gutiérrez et al. [38] proposed a web services security mechanism as a process that adds a set of specific stages into the traditional phases of WS-based systems development to provide more security. Others like Wu and Weaver [39] proposed, for federated trust management, a framework to facilitate security token exchange across autonomous security domains. Li et al. [40] presented a policy language for adaptive web services security framework and proposed a mixing reasoning framework based on rule and ontology.

Besides security concern, other web services' nonfunctional properties have also drawn attentions. Abramowicz et al. [41] discussed the web services' reliability from the perspectives of both providers and clients. Zo et al. [42] provided a basis to measure reliability of an application system that is integrated using web services. Chan et al. [43] surveyed and addressed applicability of replication and design diversity techniques for reliable web services and proposed a hybrid approach to improving the web services availability.

However, in some applications, the nonfunctional concerns for composing or integrating web services move away from just reliability [42]. Menascé [44] has discussed the major QoS issues of web services to be availability, security, response time, and throughput. Many other issues [26] come into the picture such as how to expand the existing web service protocols, standards, or architectures to accommodate the service composition need. Ardagna and Pernici [45] proposed a QoS modeling scheme for web service selection problem by formulating into a linear programming considering the quality matrix of execution time, availability, price, reputation, and data quality. Zeng et al. [46] also provide a quality model for web service composition with integer programming. Lopes et al. [47] present a service

selection middleware based on service quality metadata and user preference.

In our work, we find a new perspective to investigate the web service integration. In the situation of completing a *job* with many *tasks* for an application, each of those tasks can be supported by more than one web service. The issue is when the response time plays an important role; we might want to find a set of web services which can not only finish all the tasks, but also finish them as quickly as possible. This is essential when the outside application needs real-time response from the WSN to react promptly enough to protect life and property. Since web services are connected over the network, communication speed contributes most to the response time in most scenarios. Thus, the first step to minimize the response time is to minimize the web service hop count while executing the tasks. The idea is that if there are more than one consecutive task which can be executed within the same web service, the communication time between web services is saved.

## 3. Problem Formulation and an Architecture

This work concerns the problem of web service integration for WSNs; jobs (applications or services) consist of many tasks which are assigned to different web services over the network for execution. The web services can be WSN related or nonrelated. WSNs can provide both internal and external web services [9]. Since nowadays the amount of web services is increasing tremendously, it is very likely that a task can be supported by several web services. Our work deals with the situations which satisfy the following conditions.

(1) All web services are located on different WSNs or machines and require network connections to communicate with each other; otherwise, we can consider them as the same one web service with more than one functionality.

(2) The execution time for tasks is negligible compared to network communication time.

(3) Each web service is capable of relaying the task requests and associated data to the next web service if necessary. This can further speed up the application finish time since each web service can propagate its results along with other required information to the next web service, instead of passing the result immediately back to the application and letting the application handle the next task assignment. However, even in the situation that the application requires those intermediate results back right away, to minimize the web service hop count should still facilitate the shortening of response time.

The architecture proposed for this web service integration problem is shown in Figure 1. On the Internet side, there are many applications for different types of devices, web services, and directory services. On the other side are WSNs which are used to monitor and track the fields. Each WSN is associated with a gateway which communicates with the outside. The gateway also serves as an internal service directory since each node in the WSN can also be an internal service provider. Between the WSN world and Internet world is the Web Service Integration System (WSIS) which has several components: a WS Integration Agent, a WS Search Agent, and a WS Database. A service requester—maybe an application or service from a terminal device—sends a job integration request, containing associated tasks information, to the WS Integration Agent which in turn sends request to the WS Search Agent. The WS Search Agent collects information of supporting web services from WS Database and passes related information back to the WS Integration Agent which then decides the way in which the tasks are supported by the web services according to the appropriate methods. Then, the WS Integration Agent invokes those corresponding web services and collects results. Due to the dynamic web service publishing environment over the network, this service integration request can be executed periodically, or while a service requestor either is first built or encounters a task change depending on the requirement of the service requestor. The web service integration agent can apply different strategies, depending on the characteristics of the tasks precedence relationships, for web service integration. The job information can be stored in the WS Database for later use. After the first invoking, the WS Integration Agent can communicate directly with the web services and then get the result back.

The WS Search Agent works by collecting matching web services information from the WS Database or going to the network, most of the time using UDDI or other directory services, to look for more complete list of web services. The search work can be specified by the service requestor using different modes depending on performance concerns. For example, if the real-time response is a concern for the service requestor, the specified mode can be "static" and the search agent will only search the WS Database for matching web services. And if the specified mode is "dynamic," the search agent will go to the Internet jungle for more matching web services information, pass them to the WS Integration Agent, and store them in the web services database. The integration system can be deployed in a sense that there are multiple WS Search Agents and multiple WS Integration Agents for load sharing purpose to enhance the real-time performance of the system. Or, a master-slave WS Search Agent subsystem can also replace a single search agent for load balancing. The same concern applies to WS Integration Agent too. Furthermore, in case that a WSSA is idle, it can do update work as a background job to enrich the collections in WS Database.

A service requester can interact with the WS Integration Agent in several different ways. The first one is that it has fixed requirements of functionalities and it sends service requests to the agent. After that, it saves the information in its own storage and uses those pieces of information in the storage since then. It may go to WS Integration Agent again for help only when either there are modifications of the application or service or the chosen web services encounter some problems and cannot function properly. Of course, it can also send information to WS Integration Agent every time it needs services which may cause serious performance issues when

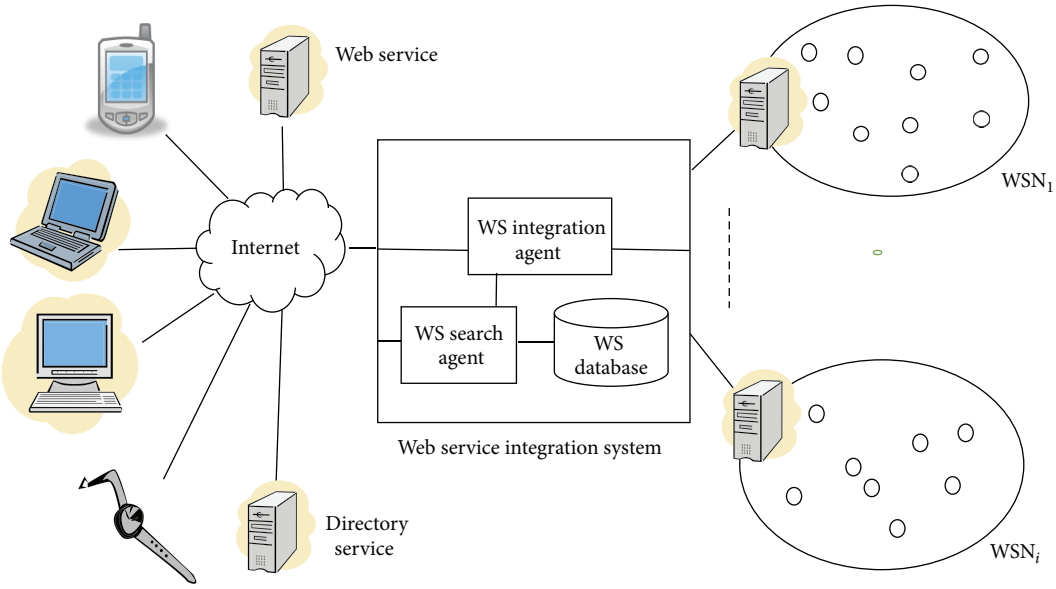



Figure 1: Web service integration system with WSNs.

real time is a concern. Another proper time for web service information update is when the service requester is idle.

While assigning tasks to those supporting web services, there may be many combinations of solutions. The *hop count* of a task assignment is defined as, by applying that specific task assignment, the number of *links* that an application uses to transport information between web services. We call this problem *Minimum Hops of Service Integration Problem*. The following example shows two different ways to integrate the services of an emergency management application in Figures 2 and 3. The application requires the tasks of (a) emergency detection service, (b) emergency unit dispatch service, and (c) notification service to conclude the job. The first case shows that the tasks are carried out by three web services while the second uses only two web services, by choosing a web service which can do both (b) and (c), leading the hop counts to be four and three, respectively. This simple example demonstrates that, by properly choosing web services, the number of web services hops, and thus the response time, can be much improved.

## 4. Two Special Cases and an Optimal Greedy Algorithm

In this section, we consider two special cases of the Minimum Hops of Service Integration Problem. The first case is proven to be NP-hard. For the second case, we proposed a polynomial time implementable, optimal algorithm to achieve the goal of minimum hop count.

The first case considered is where there exist no precedence relationships between the application tasks. We would like to find the web services assigned to tasks with minimum number of hop counts. This problem, no precedence task assignment (NPTA) problem, is equivalent to the problem of using a minimum number of web services to cover all the tasks. It can be proved to be NP-hard with the minimum set cover problem [48] reduction.

The set cover problem is considering the situation of giving an input of several sets. These sets may have some elements in common. The problem is to find the minimum number of sets so that the sets chosen contain all the elements which are in any of the sets. This problem is NP-hard [48].

**Theorem 1.** *The NPTA problem is NP-hard.*

*Proof.* We can reduce any set cover problem to one of the NPTA problems by the following steps:

(1) converting each element of the set cover problem into a task in the NPTA problem,

(2) converting each set of the set cover problem into a web service in the NPTA problem.

A solution which leads to the minimum number of web services to finish the execution of an application in NPTA problem is corresponding to the minimum number of sets to cover all the elements in the set cover problem. Thus, the NPTA problem with minimum hop count constraint is NP-hard. □

**Theorem 2.** *The web services integration problem with minimum hop count constraint is NP-hard.*

*Proof.* This is trivial from Theorem 1 since case 1 is a subset of the web service integration problem. □

The test cover problem [49] is also equivalent to the first case of task assignment problem. Many heuristic algorithms [49–54] have been proposed for the minimum set cover or test cover problem, and a recent heuristic work in test suite minimization problem using concept analysis [55] can also be applied directly to solve this problem.

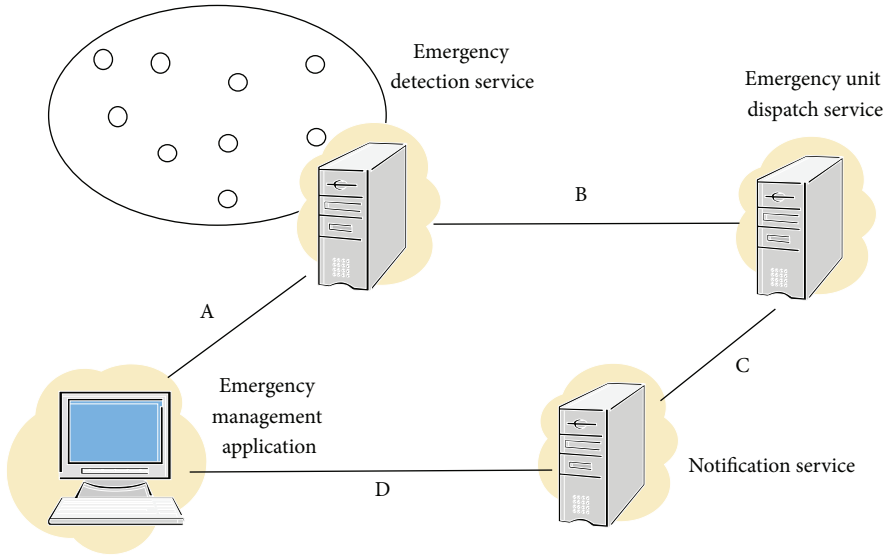



Figure 2: Emergency management example using three web services.

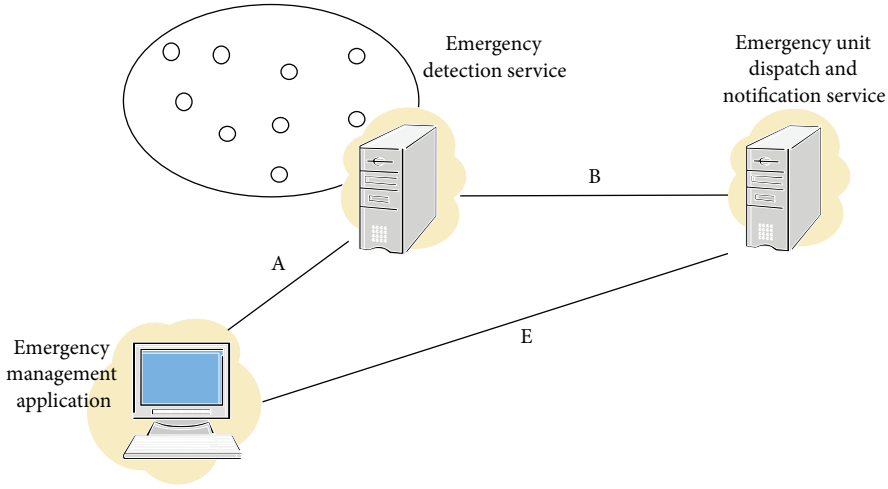



Figure 3: Emergency management example using two web services.

The second case we are focusing on solving is what can be done and how good they are when the tasks have to be finished in a specified linear order, called Linear Order Web Service Integration Problem (LOWSIP). That is, the tasks can be sorted in the order of which they need to be finished. Task $i$ has to be done before task $j$ can be proceed if and only if $i < j$. For any two consecutive tasks called task $i$ and task $i+1$, if these two tasks are placed on different web services, then the web service assigned to execute task $i$ needs to carry over information to the web service that is to execute task $i+1$ via SOAP protocol. In other words, if two consecutive tasks are assigned to the same web service, then there is no information passing between web services.

We provide an optimal algorithm, OPT, to integrate web services using greedy strategy in the sense of minimizing the hop count between web services in order to speed up the response time. The OPT algorithm is shown in the flow chart as in Figure 4 and works as follows. Recall that there are a set of linear-ordered tasks to be executed and a set of web services each of which is capable of performing certain tasks.

(1) Divide each web service $WS_i$ into the minimum number of miniweb services $ws_{i1}, ws_{i2}, \ldots, ws_{ij}$ where each miniweb service can serve a set of maximal consecutive tasks. Each miniweb service contains trio information: its original web service, the starting task ID, and the last task ID.

(2) Sort the miniweb services in the increasing order of the first tasks they serve. If two miniweb services share the same starting tasks, then remove the one with the smaller last task ID. Keep the leftover miniweb services in an ordered miniweb service (OMWS) list.

(3) Loop Steps 4 and 5 through all the tasks starting from task one. Let the current task be task $i$.

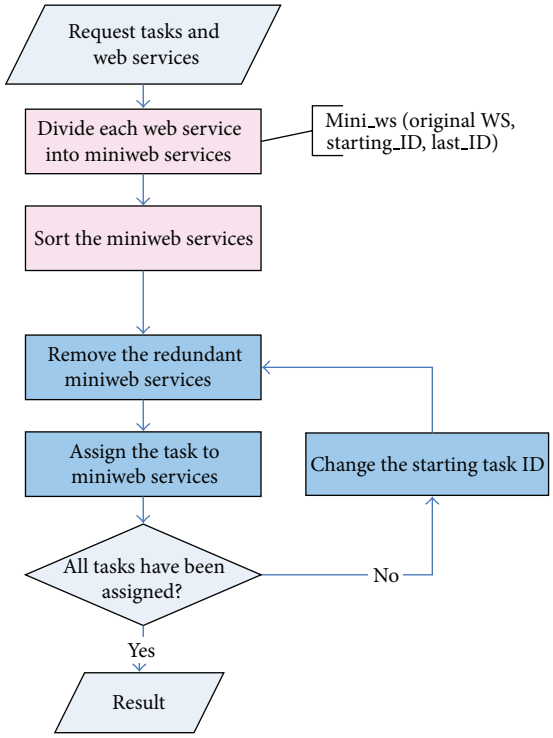



Figure 4: Flowchart for the OPT algorithm.

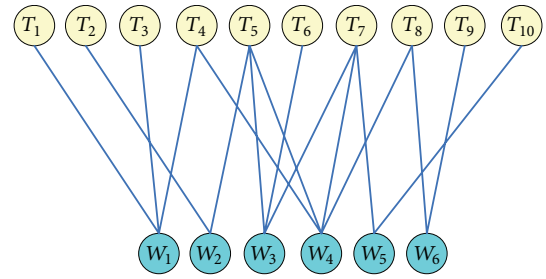



Figure 5: Tasks and web services relationship as an example.

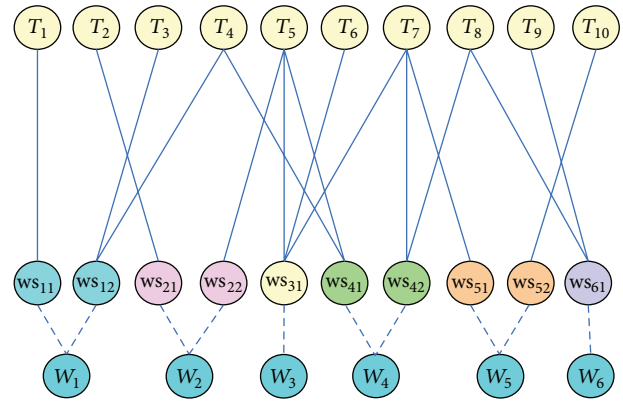



Figure 6: Task and miniweb services relationships.

(4) Task $i$ is assigned to the first miniweb service, $w$, remaining in the OMWS list.

(5) Update the OMWS list to remove the redundant miniweb services and the just assigned one. (a) For all miniweb services remaining in the list, update their start task ID to remove the tasks already covered by $w$. After this, if a miniweb service $a$ is covered by another one $b$, remove $a$ from the OMWS list. (b) Remove $w$ from the list.

(6) The true web service assigned to each task can be retrieved from the first part of the miniweb service trio information.

An example is used to demonstrate the concepts and relationships between the tasks and web services of OPT algorithm. In Figure 5, the original task and web service correlations are shown. It shows, for instance, that web service $W_1$ can serve tasks $T_1$, $T_3$, and $T_4$.

Figure 6 shows the result of the web services' breaking up into miniweb services (Step 1). For example, web service $W_1$ is breaking up into miniweb services $ws_{11}$ and $ws_{12}$, where $ws_{11}$ serves $T_1$ and $ws_{12}$ serves $T_3$ and $T_4$.

Figure 7 (Step 2) exhibits the relationships after the sorting of all miniweb services. The miniweb services are lined up in the order of $ws_{11}$, $ws_{21}$, $ws_{12}$, $ws_{41}$, $ws_{22}$, $ws_{31}$, $ws_{51}$, $ws_{42}$, $ws_{61}$, and $ws_{52}$. Miniweb service $ws_{21}$ is placed before $ws_{12}$ since it has the starting task ID, $T_2$, before that of $ws_{12}$, $T_3$.

Then, the task assignment and redundant mini task removing start. From Step 4, task $T_1$ is assigned to $ws_{11}$ which is actually $W_1$. There are no redundant miniweb services at this time (Step 5). The process continues. Task $T_2$ is in $W_2$. Task $T_3$ is in $W_1$ which makes $T_4$ in $W_1$ too. Now, $ws_{41}$ has $T_5$, $ws_{22}$ has $T_5$, and $ws_{31}$ has $T_5$, $T_6$, and $T_7$. Thus, $ws_{41}$ and $ws_{22}$ are eliminated since they are redundant. The result after looping Steps 4 and 5 is in Figure 8. The dash lines in Figure 8 are the removed link and eliminated miniweb services due to the duplication of the coverage of tasks (Step 5). The final result can also be viewed from Figure 8 which reveals the web service serving each task. Thus, to complete all tasks $T_1$ to $T_{10}$, the web service sequence used for the tasks is $WS_1$, $WS_2$, $WS_1$, $WS_1$, $WS_3$, $WS_3$, $WS_3$, $WS_6$, $WS_6$, and $WS_5$. The number of web service hops for this example is five. The result is an optimal solution which means any solution other than this one would have at least five web service hops.

**Theorem 3.** *Algorithm OPT is optimal.*

*Proof.* The algorithm OPT can be proved to be optimal. Assume that the solution generated from OPT algorithm, $S_{opt}$, is not optimal for a tasks-web services scenario. Then, there exists a real optimal solution, say $S_r$, which leads to a minimum number of web service hops while $S_{opt}$ does not. That is, this real optimal solution can divide the tasks into fewer chunks than $S_{opt}$ can, where each chunk of tasks is a set of consecutive tasks which are served by the same web service. Let is compare the first chunks in both $S_{opt}$ and $S_r$. Since OPT algorithm always picks up a web service which supports the current task and supports the most other remaining tasks, the size of the first chunk in $S_{opt}$ must be larger than or equal to the size of that in $S_r$. So, we can replace

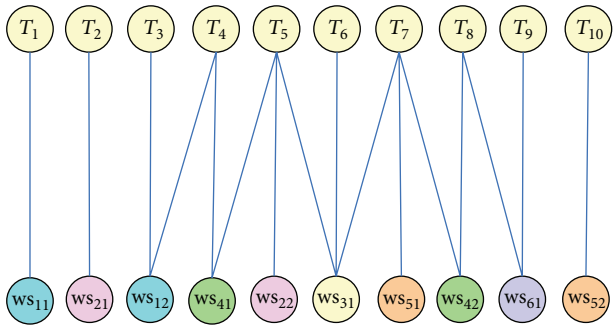



Figure 7: Result of miniweb service sorting.

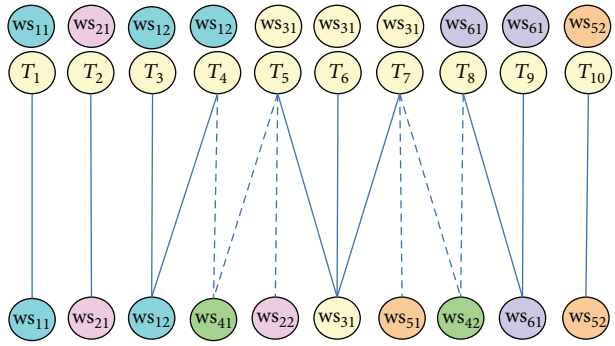



Figure 8: Task and web service assignments.

Table 1: Web services and task relationship.

| Web services supporting relationships |
|---|
| $W_1$ (0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0) |
| $W_2$ (1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0) |
| $W_3$ (0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0) |
| $W_4$ (1, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0) |
| $W_5$ (0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0) |
| $W_6$ (1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1) |
| $W_7$ (1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0) |
| $W_8$ (0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1) |
| $W_9$ (0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1) |
| $W_{10}$ (1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1) |

the first web service in $S_r$ with the first web service in $S_{opt}$ without affecting the number of web service hops for $S_r$. This process continues and we find out that $S_r$ can be completely converted into $S_{opt}$ without increasing the number of web service changes, so there is no way for $S_r$ to have a less number of web service changes than $S_{opt}$. Thus, we prove that OPT produces an optimal solution to have the least number of web service changes. This algorithm requires polynomial time to complete the task assignment work. □

## 5. Detailed Data Structure Illustration

We use another example to illustrate the details when the data structures used between web services and tasks are a Boolean matrix. In Table 1, there are 20 tasks and 10 web services, and the matrix exhibits the supporting relationships between them. Each $W_i$ represents a web service, and the Boolean values following $W_i$ are the supporting conditions between this web service and all tasks. The value 1 means that $W_i$ supports the corresponding task whereas value 0 otherwise. As in the table, web service $W_1$ supports tasks $T_2$, $T_3$, $T_6$, $T_8$, $T_9$, $T_{10}$, $T_{15}$, $T_{16}$, $T_{17}$, $T_{18}$, and $T_{19}$.

Table 2: Tasks and mini web services relationship after sorting.

| Start task | Mini web services |
|---|---|
| $T_1$ | $ws_{2.1}$ (1, 2), $ws_{4.1}$ (1, 3), $ws_{6.1}$ (1, 2), $ws_{7.1}$ (1, 1), $ws_{10.1}$ (1, 2) |
| $T_2$ | $ws_{1.1}$ (2, 3), $ws_{5.1}$ (2, 4), $ws_{9.1}$ (2, 4) |
| $T_3$ | $ws_{3.1}$ (3, 5), $ws_{8.1}$ (3, 3) |
| $T_4$ | $ws_{6.2}$ (4, 6) |
| $T_5$ | $ws_{4.2}$ (5, 7), $ws_{7.2}$ (5, 7), $ws_{8.2}$ (5, 6), $ws_{10.2}$ (5, 5) |
| $T_6$ | $ws_{1.2}$ (6, 6), $ws_{2.2}$ (6, 7), $ws_{5.2}$ (6, 6) |
| $T_7$ | $ws_{3.2}$ (7, 8) |
| $T_8$ | $ws_{1.3}$ (8, 10), $ws_{5.3}$ (8, 10), $ws_{6.3}$ (8, 13), $ws_{8.3}$ (8, 10), $ws_{10.3}$ (8, 9) |
| $T_9$ | $ws_{2.3}$ (9, 11), $ws_{7.3}$ (9, 10) |
| $T_{10}$ | $ws_{3.3}$ (10, 10), $ws_{9.2}$ (10, 14) |
| $T_{11}$ | $ws_{4.3}$ (11, 12) |
| $T_{12}$ | $ws_{3.4}$ (12, 14), $ws_{7.4}$ (12, 13) |
| $T_{13}$ | $ws_{10.4}$ (13, 15) |
| $T_{14}$ | $ws_{2.4}$ (14, 14), $ws_{8.4}$ (14, 16) |
| $T_{15}$ | $ws_{1.4}$ (15, 19), $ws_{4.4}$ (15, 16), $ws_{5.4}$ (15, 18), $ws_{7.5}$ (15, 17) |
| $T_{16}$ | |
| $T_{17}$ | $ws_{2.5}$ (17, 18), $ws_{3.5}$ (17, 19) |
| $T_{18}$ | $ws_{6.4}$ (18, 20) |
| $T_{19}$ | $ws_{4.5}$ (19, 19), $ws_{7.6}$ (19, 19), $ws_{8.5}$ (19, 20), $ws_{10.5}$ (19, 20) |
| $T_{20}$ | $ws_{9.3}$ (20, 20) |

The web services are divided up into miniweb services, each of which supports maximum number of consecutive tasks. The next step is to sort the miniweb services according to their starting task, and the result is in Table 2. After that, the miniweb service with the smaller last task ID is removed, if two miniweb services share the same starting tasks. Also, after each assignment, the OMWS list is updated.

The OPT algorithm then repeatedly selects the best miniweb services and updates the OMWS list until all tasks are satisfied. The final updated OMWS list is shown in Table 3. The web service assignment result is in Figure 5.

Starting from task $T_1$, OPT algorithm picks up the miniweb services in the OMWS list which supports $T_1$ and also supports most other tasks, that is, $ws_{4.1}$. Thus, tasks $T_1 \sim T_3$ will be assigned to $ws_{4.1}$. The next step is to update OMWS list. After that, the algorithm works by starting from task $T_4$, and the miniweb service left which supports $T_4$ and is with the most tasks, $ws_{6.2}$, to support will be picked to execute $T_4$, and so on.

The number of web service hop counts is six, as shown in Figure 9, and this is one optimal result derived by the OPT algorithm.

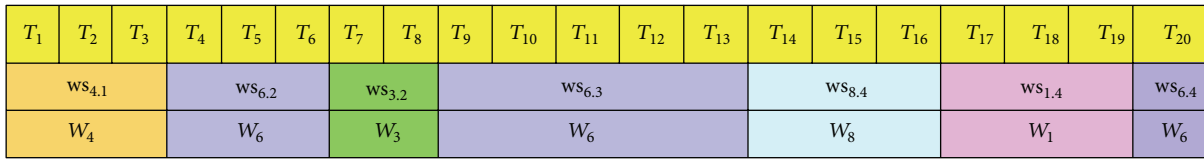



Figure 9: Tasks and their web service assignments.

Table 3: Updated miniweb services.

| Start task | Miniweb services | Start task | Mini-web services |
|---|---|---|---|
| $T_1$ | $ws_{4.1}$ (1, 3) | $T_{11}$ | $ws_{4.3}$ (11, 12) |
| $T_2$ | $ws_{5.1}$ (2, 4) | $T_{12}$ | $ws_{3.4}$ (12, 14) |
| $T_3$ | $ws_{3.1}$ (3, 5) | $T_{13}$ | $ws_{10.4}$ (13, 15) |
| $T_4$ | $ws_{6.2}$ (4, 6) | $T_{14}$ | $ws_{8.4}$ (14, 16) |
| $T_5$ | $ws_{4.2}$ (5, 7) | $T_{15}$ | $ws_{1.4}$ (15, 19) |
| $T_6$ | $ws_{2.2}$ (6, 7) | $T_{16}$ | |
| $T_7$ | $ws_{3.2}$ (7, 8) | $T_{17}$ | $ws_{3.5}$ (17, 19) |
| $T_8$ | $ws_{6.3}$ (8, 13) | $T_{18}$ | $ws_{6.4}$ (18, 20) |
| $T_9$ | $ws_{2.3}$ (9, 11) | $T_{19}$ | $ws_{8.5}$ (19, 20) |
| $T_{10}$ | $ws_{9.2}$ (10, 14) | $T_{20}$ | $ws_{9.3}$ (20, 20) |

## 6. Conclusion and Future Work

WSNs have attracted much attention and have been integrated into bigger infrastructures such as Internet of Things. Service-oriented middle layer architecture of WSN enables the adaptability and flexibility of the whole systems. In this work, we propose a problem to enhance the WSN web service integration to reduce the hop count of web service integration with the reduction of traffic and response time in mind. The problem is called Minimum Hops of Service Integration. Two special cases are investigated. The first one is for the case when there are no precedence relationships between the tasks and it is proven to be NP-complete. However, when the task precedence relationships are linearly ordered, we develop a polynomial-time greedy algorithm, OPT, to solve this problem.

In the future, we will investigate the effectiveness of different web services integration methods on other non-functional factors in order to develop a more efficient and comprehensive way of integrating web services.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This work is sponsored by NSC of Taiwan under Grant no. NSC 100-2632-E-216-001-MY3-3.

## References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[3] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[4] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of Wireless Sensor Networks towards the Internet of Things: a Survey," in *Proceedings of the 19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM '11)*, pp. 1–6, September 2011.

[5] K. K. Khedo and R. K. Subramanian, "A service-oriented component-based middleware architecture for Wireless Sensor Networks," *International Journal of Computer Science and Network Security*, vol. 9, no. 3, pp. 174–182, 2009.

[6] N. Mohamed and J. Al-Jaroodi, "A survey on service-oriented middleware for wireless sensor networks," *Service Oriented Computing and Applications*, vol. 5, no. 2, pp. 71–85, 2011.

[7] N. Mohamed and J. Al-Jaroodi, "Service-oriented middleware approaches for wireless sensor networks," in *Proceedings of the 44th Hawaii International Conference on System Sciences (HICSS-44 '10)*, pp. 1–9, January 2011.

[8] F. C. Delicato, P. F. Pires, L. Pirmez, and L. F. R. da Costa Carmo, "A flexible web service based architecture for wireless sensor networks," in *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCSW '03)*, p. 730, 2003.

[9] E. Avilés-López and J. A. García-Macías, "TinySOA: a service-oriented architecture for wireless sensor networks," *Service Oriented Computing and Applications*, vol. 3, no. 2, pp. 99–108, 2009.

[10] E. Cañete, J. Chen, M. Díaz, L. Llopis, and B. Rubio, "A service-oriented middleware for wireless sensor and actor networks," in *Proceedings of the 6th International Conference on Information Technology: New Generations (ITNG '09)*, pp. 575–580, Las Vegas, Nev, USA, April 2009.

[11] F. Golatowski, J. Blumenthal, H. Matthias, M. Haase, H. Burchardt, and D. Timmermann, "Service-oriented software architecture for sensor networks," in *Proceedings of International Workshop on Mobile Computing (IMC '03)*, pp. 93–98, June 2003.

[12] M. P. Singh and M. N. Huhns, *Service-Oriented Computing: Semantics, Processes, Agents*, John Wiley and Sons, 2005.

[13] A. Maurizio, J. Sager, P. Jones, G. Corbitt, and L. Girolami, "Service oriented architecture: challenges for business and academia," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS '08)*, pp. 315–322, January 2008.

[14] MSDN Architecture Center, "Service Oriented Architecture," http://msdn.microsoft.com/en-us/architecture/aa948857.aspx.

[15] Oracle, "Oracle Service-Oriented Architecture," http://www.oracle.com/technologies/soa/index.html.

[16] IBM, "Service Oriented Architecture-SOA," http://www-306.ibm.com/software/solutions/soa/.

[17] Sun, "Service-Oriented Architecture (SOA)," http://www.sun.com/products/soa/index.jsp.

[18] M. Nicolai Josuttis, *SOA in Practice: The Art of Distributed System Design*, O'Reilly, 2007.

[19] Thomas Erl, *SOA Principles of Service Design*, Prentice Hall, 2007.

[20] G. Moritz, F. Golatowski, C. Lerche, and D. Timmermann, "Beyond 6LoWPAN: web services in WSNs," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 1795–1805, 2013.

[21] Y. V. Natis, "Applied SOA: transforming fundamental principles into best practices," Gartner Research Note Id G00147098, 2007.

[22] J. McCormick, "6 Keys to SOA Success," Baseline, 2007, http://www.baselinemag.com/article2/0,1540,2139359,00.asp.

[23] K.-J. Lin, "The design of an accountability framework for service engineering," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS '08)*, pp. 108–116, January 2008.

[24] C.-S. Li, *Managing Business Integrity Across Policies, Processes and Information*, IBM Information On Demand, 2007.

[25] D. Woods and T. Mattern, *Enterprise SOA: Designing IT for Business Innovation*, O'Reilly, 2006.

[26] S. Dustdar and W. Schreiner, "A survey on web services composition," *International Journal of Web and Grid Services*, vol. 1, no. 1, pp. 1–30, 2005.

[27] W. Ouyang and M. -L. Chen, "An Optimal web services Integration using Greedy Strategy," in *IEEE Asia-Pacific Services Computing Conference*, pp. 568–573, 2008.

[28] S. Hong, D. Kim, M. Ha et al., "SNAIL: an IP-based wireless sensor network approach to the Internet of things," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 34–42, 2010.

[29] S. Li, L. Da Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and internet of things," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2177–2186, 2013.

[30] L. Liu, X. Zhang, and H. Ma, "Percolation theory-based exposure-path prevention for wireless sensor networks coverage in internet of things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3625–3636, 2013.

[31] F. Li and P. Xiong, "Practical secure communication for integrating wireless sensor networks into the internet of things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3677–3684, 2013.

[32] S. Hadim and N. Mohamed, "Middleware: middleware challenges and approaches for wireless sensor networks," *IEEE Distributed Systems Online*, vol. 7, no. 3, pp. 1–23, 2006.

[33] H. Guan, B. Jin, J. Wei, W. Xu, and N. Chen, "A framework for application server based web services management," in *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC '05)*, pp. 1–8, December 2005.

[34] F.-C. Cheng, T.-C. Hung, Y.-J. Chiou, and T.-C. Chang, "Design and implementation of Web service integration tool," in *IEEE International Workshop on Service-Oriented System Engineering*, pp. 91–96, October 2005.

[35] H. Mohanty, J. Mulchandani, D. Chenthati, and R. K. Shyamasundar, "Modeling Web Services with FSM Modules," in *Proceedings of the 1st Asia international Conference on Modelling & Simulation*, pp. 100–105, March 2007.

[36] K. Tang, S. Chen, D. Levy, J. Zic, and B. Yan, "A performance evaluation of web services security," in *Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC '06)*, pp. 67–74, chn, October 2006.

[37] N. Sidharth and J. Liu, "IAPF: a framework for enhancing web services security," in *Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC '07)*, pp. 23–30, July 2007.

[38] C. Gutiérrez, E. Fernández-Medina, and M. Piattini, "PWSSec: process for Web services security," in *Proceedings of the IEEE International Conference on Web Services (ICWS '06)*, pp. 213–222, September 2006.

[39] Z. Wu and A. C. Weaver, "Using web services to exchange security tokens for federated trust management," in *Proceedings of the IEEE International Conference on Web Services (ICWS '07)*, pp. 1176–1178, July 2007.

[40] J.-X. Li, B. Li, L. Li, and T.-S. Che, "A policy language for adaptive Web services security framework," in *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, pp. 261–266, July 2007.

[41] W. Abramowicz, M. Kaczmarek, and D. Zyskowski, "Duality in Web services reliability," in *International Conference on Internet and Web Applications and Services/Advanced International Conference on Telecommunications*, p. 165, February 2006.

[42] H. Zo, D. L. Nazareth, and H. K. Jain, "Measuring reliability of applications composed of web services," in *Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS '07)*, p. 278c, January 2007.

[43] P. P. W. Chan, M. R. Lyu, and M. Malek, "Reliable Web services: methodology, experiment and modeling," in *Proceedings of the IEEE International Conference on Web Services (ICWS '07)*, pp. 679–686, July 2007.

[44] D. A. Menascé, "QoS issues in web services," *IEEE Internet Computing*, vol. 6, no. 6, pp. 72–75, 2002.

[45] D. Ardagna and B. Pernici, "Adaptive service composition in flexible processes," *IEEE Transactions on Software Engineering*, vol. 33, no. 6, pp. 369–384, 2007.

[46] L. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for Web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.

[47] F. Lopes, T. Batista, E. Cavalcante et al., "Dynamic and semantic web services composition for ubiquitous computing," in *Proceedings of the 18th Brazilian symposium on Multimedia and the web*, pp. 151–160, 2012.

[48] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Co., San Francisco, Calif, USA, 1979.

[49] H. Agrawal, "Dominators. super blocks, and program coverage," in *Proceedings of the 21st Annual ACM Symposium on Principles of Programming Languages*, pp. 25–34, January 1994.

[50] H. Agrawal, "Efficient coverage testing using global dominator graphs," in *Proceedings of the ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering*, pp. 11–20, Toulouse, France, 1999, SIGSOFT Software Engineering Notes, vol. 24, no. 5, pp.11–20, September 1999.

[51] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, 1979.

[52] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2nd, edition, 2001.

[53] M. J. Harrold, R. Gupta, and M. L. Soffa, "Methodology for controlling the size of a test suite," *ACM Transactions on Software Engineering and Methodology*, vol. 2, no. 3, pp. 270–285, 1993.

[54] M. Marré and A. Bertolino, "Using spanning sets for coverage testing," *IEEE Transactions on Software Engineering*, vol. 29, no. 11, pp. 974–984, 2003.

[55] S. Tallam and N. Gupta, "A concept analysis inspired greedy algorithm for test suite minimization," in *ACM SIGPLAN-SIGSOFT Workshop on Program Analysis for Software Tools and Engineering*, pp. 35–42, September 2005, SIGSOFT Software Engineering Notes, vol. 31, no. 1, January 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 475730, 16 pages
http://dx.doi.org/10.1155/2013/475730

*Research Article*

# Generalized Predictive Control in a Wireless Networked Control System

**Min-Fan Ricky Lee, Fu-Hsin Steven Chiu, Hsuan-Chiao Huang, and Christian Ivancsits**

*Graduate Institute of Automation and Control, National Taiwan University of Science and Technology, Taipei 106, Taiwan*

Correspondence should be addressed to Min-Fan Ricky Lee; rickylee@mail.ntust.edu.tw

Received 28 June 2013; Accepted 11 September 2013

Academic Editor: Qin Xin



The NCS (networked control system) is different from the conventional control systems which is the integration of the automation and control over communication network. When an NCS operates over the communication network, one of the major challenges is the network-induced delay in data transfer among the controllers, actuators, and sensors. This delay degrades system performance and causes system unstablility. This paper proposes a GPC (generalized predictive control) with the Kalman state estimator to compensate for the network-induced delay and packet loss. The GPC is implemented in WiNCS (Wireless NCS) based on IEEE 802.11 standard. An analytical NCS model and NS2 (network simulator version 2) are developed to simulate and evaluate the performance under the effect of various delays and packet loss rates. The result shows that the proposed GPC is adaptive and robust to the uncertainties in a time-delay system. The WiNCS is evaluated with latency and throughput measurements in various environments. The experiment setup conforming to the IEEE 802.11 standard achieves an average latency of 1.3 ms and a data throughput of 3.000 kB/s up to a distance of 70 m. The results demonstrate the feasibility of real-time closed-loop control with the proposed concept.

## 1. Introduction

In recent years, there has been an increasing interest in implementing networked transmission protocols (e.g., wire/wireless local area networks) in automation and control system. Cost effectiveness and flexibility are achieved using communication protocol in the feedback control.

The NCS closes the feedback control loops through a real-time network. The control signals to the actuators and the feedback signals from sensors are in the form of information packages [1, 2]. Interconnecting the sensors, actuators, and controllers via networks can eliminate wiring, reduce installation costs, and enable remote monitoring and tuning. Additional components and modules can be added without additional circuitry to the existing layout. The controllers effectively share the data via the information technology allowing easy data fusion and integration to the controller for an intelligent decision or optimal operation in a large and complex process [3, 4]. The potential applications of NCS include industrial automation, military, hazardous environment exploration, or robots application.

Three methods on scheduling packets were proposed to improve NCS performance and stability as static scheduler, try-once-discard (TOD) scheduler with continuous priority level, and TOD with discrete priority level [5, 6].

A networked DC motor control system was proposed using controller gain adaptation to compensate the changes in QoS (quality-of-service) requirements over time-varying network [7].

Stabilization of NCS was investigated in the discrete-time domain with random delays [8]. Two Markov chains were applied to model the delay on controller-to-actuator delay and sensor-to-controller.

Model-based NCS was proposed using an explicit model of the plant to produce an estimate of the plant state during transmission delay. The stability was evaluated for the controller/actuator which was updated with sensor information at nonconstant time intervals [9].

An NCS model including network-induced delay and packet loss in transmission network was proposed. The feedback gain of a memory-less controller and the maximum allowable value of the network-induced delay were derived by solving a set of linear matrix inequalities [10]. Two predictors estimating the plant outputs in open-loop and closed-loop were proposed [11].

An error predictive model was built using a back propagation neural network to reduce the error on estimation of output. Three control methods were compared as PID, GPC, and GPC with error correction. GPC with error correction was validated to have the best performance [12].

A novel GPC strategy was proposed controlling NCS with respect to the NSC structure characteristics. The timestamp mechanism of data communication network was applied. Accurate measurement to the system output and timely modification to the predictive value were required under the random network-induced delay [13].

A client-server control architecture was implemented on the dual-axis hydraulic position system of an industrial fish-processing machine. The GPC algorithm was adopted to compensate for data-transmission delays. It incorporates a minimum-effort estimator to estimate missing or delayed sensor data and a variable-horizon adaptive GPC controller to predict the required future control efforts to drive the plant to track a desired reference trajectory [14].

Time-varying delays for the transmission of sensor and control signal over the wireless network were evaluated using a randomized multihop routing protocol. The proposed predictive control scheme with a delay estimator was based on a Kalman filter [15].

This paper presents a model of the NCS with network-induced delay and packets loss for a general SISO NCS model. The stochastic time delays reduce the system performance (e.g., stability, controllability, and observability). This paper applies GPC to predict the network-induced delay and simulate it through the wireless network environment setup by NS2 in Linux. The PiccSIM is used as the platform in the client/server architecture for the WiNCS. The MPC concept was adopted and the GPC control algorithm with the Kalman state estimator is implemented in WiNCS to reduce the effect on network-induced delay and packets loss.

The contributions in the paper are summarized as follows.

(i) The main factors affecting the performance of NCS in communication networks have been identified.

(ii) The NCS with network-induced delay and packet loss is modeled.

(iii) The GPC algorithm is implemented in WiNCS to cope with the time-varying delay issue.

(iv) The simulated platform is constructed which connects NS2 and Matlab\Simulink for implementation of GPC in WiNCS.

## 2. Method

The GPC is proposed to compensate the network-induced delay in WiNCS. The algorithm, closed-loop structure, and CARIMA model structure are developed. The GPC in state space with state estimator is derived for WiNCS simulation. The state space is adopted to reduce the algebraic complexity in the GPC control law.

*2.1. Formulation of GPC.* The SISO (single-input single-output) system is given which considers the operation around a specific set point after linearization. A predictive model known as CARIMA (controlled autoregressive integrated moving average) for GPC is

$$A\left(z^{-1}\right) y(k) = B\left(z^{-1}\right) u(k-1) + \frac{C\left(z^{-1}\right)}{\Delta} e(k), \quad (1)$$

with

$$\Delta = 1 - z^{-1}, \quad (2)$$

where $y(k)$ is output signal, $u(k)$ is input signal, $e(k)$ is zero mean white noise, and $A$, $B$, and $C$ are

$$\begin{aligned} A\left(z^{-1}\right) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{na} z^{-na} \\ B\left(z^{-1}\right) &= b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{nb} z^{-nb} \\ C\left(z^{-1}\right) &= 1 + c_1 z^{-1} + c_2 z^{-2} + \cdots + c_{nc} z^{-nc}, \end{aligned} \quad (3)$$

where $C(z^{-1})$ is selected to be 1 for the simplicity. The cost function including the influence of $u(k)$ on future system is to enhance the system robustness. GPC algorithm applies a control sequence to minimize a multistage cost function as

$$\begin{aligned} \mathbf{J} = &\sum_{j=n_1}^{n_2} \delta(j) \left[\hat{y}(k+j) - w(k+j)\right]^2 \\ &+ \sum_{j=1}^{H_c} \lambda(j) \left[\Delta u(k+j-1)\right]^2, \end{aligned} \quad (4)$$

where $\hat{y}(k+j)$ is optimum $j$-step ahead prediction of system output, $n_1$ and $n_2$ are the minimum and maximum of the prediction horizons $H_p$ and the order of $n_2$ must be larger than $B(z^{-1})$, $H_c$ is control horizon ($H_c \le H_p$), $\delta(j)$ and $\lambda(j)$ are weighting sequences, $\delta(j)$ is selected to be 1, and $\lambda(j)$ is a constant. $w(k+j)$ is the future reference trajectory as

$$w(k+j) = \alpha^j y(k+j) + \left(1 - \alpha^j\right) y_r \quad (j = 1, 2, 3 \cdots, n), \quad (5)$$

where $y(k)$ and $y_r$ are the set point and the future output of the system, respectively. $\alpha$ is a parameter between 0 and 1 that affects the response of the system (closer to 1, smoother response curve).

The optimal prediction of the output $y(k+j)$ is driven close to $w(k+j)$ to optimize the cost function. Diophantine equation for predicting the precede $j$-step output is given by

$$1 = E_j\left(z^{-1}\right) \widetilde{A}\left(z^{-1}\right) + z^{-j} F_j\left(z^{-1}\right), \quad (6)$$

with

$$\widetilde{A}\left(z^{-1}\right) = \Delta A\left(z^{-1}\right) = 1 + \widetilde{a}_1 z^{-1} + \widetilde{a}_2 z^{-2} + \cdots + \widetilde{a}_{na+1} z^{-(na+1)}, \tag{7}$$

where

$$E_j\left(z^{-1}\right) = e_{j,0} + e_{j,1} z^{-1} + e_{j,2} z^{-2} + \cdots + e_{j,j-1} z^{-(j-1)}$$
$$F_j\left(z^{-1}\right) = f_{j,0} + f_{j,1} z^{-1} + f_{j,2} z^{-2} + \cdots + f_{j,na} z^{-na}. \tag{8}$$

Equation (1) is multiplied by $\Delta E_j(z^{-1}) z^j$ to obtain the predictive equation of $j$-step after time $k$ as

$$\widetilde{A}\left(z^{-1}\right) E_j\left(z^{-1}\right) y(k+j) = E_j\left(z^{-1}\right) B\left(z^{-1}\right) \Delta u(k+j-1) + E_j\left(z^{-1}\right) e(k+j). \tag{9}$$

Equation (9) is rewritten in consideration of (6) as

$$y(k+j) = F_j\left(z^{-1}\right) y(k) + E_j\left(z^{-1}\right) B\left(z^{-1}\right) \times \Delta u(k+j-1) + E_j\left(z^{-1}\right) e(k+j). \tag{10}$$

Noise term in the future on the system is neglected in (10). Letting $G_j(z^{-1}) = E_j(z^{-1}) B(z^{-1})$, the best prediction of future output is

$$\hat{y}(k+j) = G_j\left(z^{-1}\right) \Delta u(k+j-1) + F_j\left(z^{-1}\right) y(k). \tag{11}$$

The set of the control signals $u(k), u(k+1), \ldots, u(k+N)$ is obtained to optimize (4). The values $n_1$, $n_2$, and $H_c$ are defined by $n_1 = 1$ and $n_2 = N$ and the predictive horizon $H_p = N$ and the control horizon $H_c = N$. $N$ ahead optimal prediction is considered as

$$\begin{aligned}
\hat{y}(k+1) &= G_1 \Delta u(k) + F_1\left(z^{-1}\right) y(k) \\
\hat{y}(k+2) &= G_2 \Delta u(k+1) + F_2\left(z^{-1}\right) y(k) \\
\hat{y}(k+3) &= G_3 \Delta u(k+2) + F_3\left(z^{-1}\right) y(k) \\
&\vdots \\
\hat{y}(k+N) &= G_N \Delta u(k+N-1) + F_N\left(z^{-1}\right) y(k).
\end{aligned} \tag{12}$$

The above equations are marshaled as

$$\hat{\mathbf{Y}} = \mathbf{G}\Delta\mathbf{U} + F\left(z^{-1}\right) y(k) + H\left(z^{-1}\right) \Delta u(k-1), \tag{13}$$

where

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \\ \vdots \\ \hat{y}(k+N) \end{bmatrix}; \quad \mathbf{G} = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix};$$

$$\Delta\mathbf{U} = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix}; \quad F\left(z^{-1}\right) = \begin{bmatrix} F_1\left(z^{-1}\right) \\ F_2\left(z^{-1}\right) \\ \vdots \\ F_N\left(z^{-1}\right) \end{bmatrix}$$

$$H\left(z^{-1}\right) = \begin{bmatrix} \left(G_1\left(z^{-1}\right) - g_0\right) z \\ \left(G_2\left(z^{-1}\right) - g_0 - g_1 z^{-1}\right) z^2 \\ \vdots \\ \left(G_N\left(z^{-1}\right) - g_0 - g_1 z^{-1} - \cdots g_{N-1} z^{-(N-1)}\right) z^N \end{bmatrix}. \tag{14}$$

In (13), it includes known and unknown sequences at time $k$. The known sequence which is the last two terms is grouped into $\mathbf{f}$ as

$$\hat{\mathbf{Y}} = \mathbf{G}\Delta\mathbf{U} + \mathbf{f}, \tag{15}$$

where

$$\mathbf{f} = F\left(z^{-1}\right) y(k) + H\left(z^{-1}\right) \Delta u(k-1). \tag{16}$$

Equation (4) is written in consideration (15) as

$$\mathbf{J} = (\mathbf{G}\Delta\mathbf{U} + \mathbf{f} - \mathbf{W})^T (\mathbf{G}\Delta\mathbf{U} + \mathbf{f} - \mathbf{W}) + \lambda \Delta\mathbf{U}^T \Delta\mathbf{U}, \tag{17}$$

where

$$\mathbf{W} = \begin{bmatrix} w(k+1) \\ w(k+2) \\ \vdots \\ w(k+N) \end{bmatrix}. \tag{18}$$

The minimum of $\mathbf{J}$, assuming there are no constraints on the control signal, is found by taking gradient of $\mathbf{J}$. Let $\partial\mathbf{J}/\partial\Delta\mathbf{U} = 0$ which leads to

$$\Delta\mathbf{U} = \left(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}\right)^{-1} \mathbf{G}^T (\mathbf{W} - \mathbf{f}). \tag{19}$$

In (19), the actually control signal that is sent to the system is the first element of $\Delta\mathbf{U}$ as

$$u(k) = u(k-1) + K(\mathbf{W} - \mathbf{f}), \tag{20}$$

where $K = [k_1 \; k_2 \; \cdots \; k_N]$ is the first row of the matrix $(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T$.

The optimization in GPC is different from the general optimal algorithm; the optimized target is moving by time (i.e., local optimization in every sampling time). The first element of $\Delta\mathbf{U}$ is applied and the optimal procedure is repeated at the next sampling time [16, 17].

*2.2. GPC Control Strategy.* The following equations are rewritten to illustrate the GPC control block diagram.

(a) The reference trajectory in (5) is

$$\mathbf{W} = Qy(k) + Py_r, \tag{21}$$

where $Q = [\alpha \ \alpha^2 \ \cdots \ \alpha^n]^T$ and $P = [1-\alpha \ \ 1-\alpha^2 \ \ \cdots \ \ 1-\alpha^n]^T$.

(b) The predictive model (CARIMA) in (1) is

$$\Delta A\left(z^{-1}\right) y(k) = z^{-1} B\left(z^{-1}\right) \Delta u(k) + C\left(z^{-1}\right) e(k). \tag{22}$$

From (7) and (22), the CARIMA model is driven as

$$y(k) = \frac{z^{-1}B}{A} u(k) + \frac{C}{A \cdot \Delta} e(k), \tag{23}$$

where the polynomial $C$ is selected to be 1.

(c) The predictive output in (16) is

$$\mathbf{f} = F\left(z^{-1}\right) y(k) + \overline{H}\left(z^{-1}\right) \Delta u(k), \tag{24}$$

where

$$\overline{H}\left(z^{-1}\right) = \begin{bmatrix} \left(G_1\left(z^{-1}\right) - g_0\right) \\ \left(G_2\left(z^{-1}\right) - g_0 - g_1 z^{-1}\right) z \\ \vdots \\ \left(G_N\left(z^{-1}\right) - g_0 - g_1 z^{-1} - \cdots g_{N-1} z^{-(N-1)}\right) z^{N-1} \end{bmatrix}. \tag{25}$$

(d) The control-increment vector is

$$\Delta u(k) = K(\mathbf{W} - \mathbf{f}). \tag{26}$$

Figure 1 shows that the GPC control-loop structure consists of smoothing, tuning, and prediction processes. The thick line indicates the vector signal and the thin line indicates the scalar signal. At each moment, the desired output vector $W$ is obtained after smoothing the set point $y_r$. Compared with the predictive output and desired output, the declination vector is obtained. The control signal $\Delta u$ at this moment is the product of declination vector and vector $K$. The control signal $\Delta u$ also generates the new predictive output $\mathbf{f}$ with the vector $\overline{H}$ and the system output.

*2.3. GPC in State-Space Formulation.* Consider a state-space description [18, 19] of the system plant which was given as follows. The dimension of the state vector is $n = \max(n_a + 1, n_b + 1, n_c)$ :

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + \Pi\omega(k) \\ y(k) &= Cx(k) + \nu(k), \end{aligned} \tag{27}$$

where

$$A = \begin{bmatrix} -\widetilde{a}_1 & 1 & 0 & \cdots & 0 \\ -\widetilde{a}_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\widetilde{a}_n & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -\widetilde{a}_{na+1} & 0 & 0 & \cdots & 0 \end{bmatrix}; \qquad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ b_0 \\ \vdots \\ b_{nb-1} \end{bmatrix};$$

$$\Pi = \begin{bmatrix} c_1 - \widetilde{a}_1 \\ c_2 - \widetilde{a}_2 \\ \vdots \\ c_n - \widetilde{a}_n \\ \vdots \\ c_{nc} - \widetilde{a}_{nc} \end{bmatrix}, \qquad C = [1 \ \ 0 \ \cdots \ 0 \ \cdots \ 0 \ \cdots], \tag{28}$$

where $\widetilde{a}_i$ are the coefficient of polynomial $\widetilde{A}$. The random variables $\omega$ and $\nu$ represent disturbance input and measurement (sensor) noise, and they are assumed to be white Gaussian zero mean with normal probability distributions.

The noise and disturbance in (22) are neglected; the predictive model is rewritten as

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k). \end{aligned} \tag{29}$$

From (27), the $z$-domain transfer function $R(z)$ is derived as

$$\begin{aligned} R(z) &= C(z\mathbf{I} - A)^{-1} B \\ &= \frac{C(\mathbf{I} - (A/z))^{-1} B}{z} = \frac{C}{z}\left(\mathbf{I} + \frac{A}{z} + \frac{A^2}{z^2} + \frac{A^3}{z^3} + \cdots\right) B \\ &= \frac{CB}{z} + \frac{CAB}{z^2} + \frac{CA^2B}{z^3} + \frac{CA^3B}{z^4} + \cdots. \end{aligned} \tag{30}$$

From (30), the current output is obtained as

$$\begin{aligned} y(k) = {} & CB\Delta u(k-1) + CAB\Delta u(k-2) \\ & + CA^2 B\Delta u(k-3) + \cdots. \end{aligned} \tag{31}$$

Since the predictive horizon $H_p = N$, the future output is obtained as

$$\begin{aligned} y(k+1) = {} & CB\Delta u(k) + CAB\Delta u(k-1) \\ & + CA^2 B\Delta u(k-2) + \cdots \\ y(k+2) = {} & CB\Delta u(k+1) + CAB\Delta u(k) \\ & + CA^2 B\Delta u(k-1) + \cdots \\ & \vdots \\ y(k+N) = {} & CB\Delta u(k+N-1) + CAB\Delta u(k+N-2) \\ & + CA^2 B\Delta u(k+N-3) + \cdots. \end{aligned} \tag{32}$$

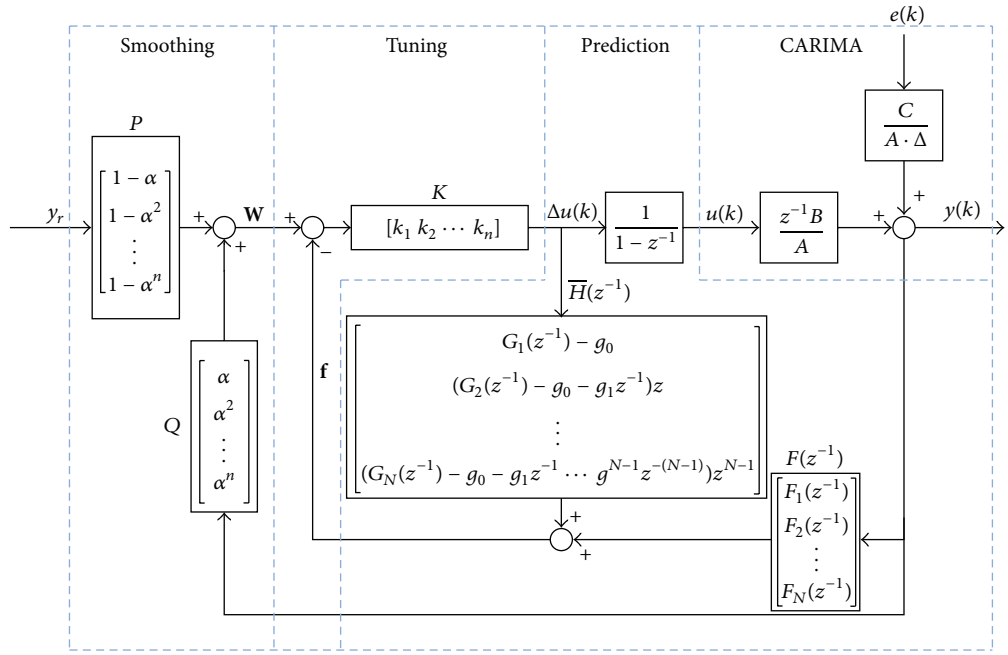



Figure 1: GPC control-loop structure.

From the previous equations, the prediction state of the system is also obtained as

$$\begin{aligned} x(k+1) &= Ax(k) + B\Delta u(k) \\ x(k+2) &= Ax(k+1) + B\Delta u(k+1) \\ &= A^2x(k) + AB\Delta u(k) + B\Delta u(k+1) \\ x(k+3) &= A^3x(k) + A^2B\Delta u(k) \\ &\quad + AB\Delta u(k+1) + B\Delta u(k+2) \\ &\vdots \\ x(k+j) &= A^jx(k) + \sum_{i=1}^{j-1} A^{j-i-1}B\Delta u(k+i). \end{aligned} \tag{33}$$

A general term of $y(k+j)$ with $(j = 1, 2, 3, \ldots, N)$ is obtained as

$$\begin{aligned} y(k+j) &= \sum_{i=1}^{\infty} CA^{i-1}B\Delta u(k+j-i) \\ &= \sum_{i=1}^{j} CA^{i-1}B\Delta u(k+j-i) \\ &\quad + \sum_{i=j+1}^{\infty} CA^{i-1}B\Delta u(k+j-i) \\ &= \sum_{i=1}^{j} CA^{i-1}B\Delta u(k+j-i) \\ &\quad + \sum_{m=0}^{\infty} CA^{m+j}B\Delta u(k-m-1) \\ &= \sum_{i=1}^{j} CA^{i-1}B\Delta u(k+j-i) \\ &\quad + \sum_{m=0}^{\infty} CA^{m+j}\left[x(k-m) - Ax(k-m-1)\right] \\ &= \sum_{i=1}^{j} CA^{i-1}B\Delta u(k+j-i) \\ &\quad + \sum_{m=0}^{\infty} CA^{j}\left[A^m x(k-m) - A^{m+1}x(k-m-1)\right] \\ &= \sum_{i=1}^{j} CA^{i-1}B\Delta u(k+j-i) + CA^jx(k). \end{aligned} \tag{34}$$

Therefore, the predictive output is denoted as

$$\hat{\mathbf{Y}} = \mathbf{G}\Delta\mathbf{U} + \mathbf{f}, \tag{35}$$

where

$$\hat{\mathbf{Y}} = \begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \\ \vdots \\ \hat{y}(k+N) \end{bmatrix}; \quad \Delta\mathbf{U} = \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix}; \quad \mathbf{f} = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} x(k) \tag{36}$$

$$\mathbf{G} = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CA^{N-N_u}B \end{bmatrix}.$$

The cost function is fundamental for the determination of control action [20] and it is rewritten as

$$\mathbf{J_{adp}} = \sum_{j=1}^{N} \left\| \left[ \hat{y}(k+j) - w(k+j) \right] Q_y \right\|^2 + \sum_{j=1}^{H_c} \left\| \left[ \Delta u(k+j-1) \right] Q_u \right\|^2, \tag{37}$$

where $Q_y$ and $Q_u$ are the penalization matrixes. The cost function in (37) is further rewritten as

$$\mathbf{J_{adp}} = \left[ \left( \hat{\mathbf{Y}} - \mathbf{w} \right)^T \quad \Delta \mathbf{U}^T \right] \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix}^T \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} \begin{bmatrix} \hat{\mathbf{Y}} - \mathbf{w} \\ \Delta \mathbf{U} \end{bmatrix}. \tag{38}$$

The cost function $\mathbf{J_{adp}}$ based on square-root minimization is separated into two square roots as

$$\mathbf{J_{adp}} = \mathbf{J_m} \cdot \mathbf{J_m}, \tag{39}$$

where

$$\mathbf{J_m} = \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} \begin{bmatrix} \hat{\mathbf{Y}} - \mathbf{w} \\ \Delta \mathbf{U} \end{bmatrix}. \tag{40}$$

After obtaining the predictive outputs $\hat{\mathbf{Y}}$, the cost function $\mathbf{J_m}$ in (40) is derived as

$$\begin{aligned} \mathbf{J_m} &= \begin{bmatrix} Q_y & 0 \\ 0 & Q_u \end{bmatrix} \begin{bmatrix} \mathbf{G}\Delta\mathbf{U} + \mathbf{f} - \mathbf{w} \\ \Delta \mathbf{U} \end{bmatrix} \\ &= \begin{bmatrix} (\mathbf{G}\Delta\mathbf{U} + \mathbf{f} - \mathbf{w}) Q_y \\ \Delta \mathbf{U} Q_u \end{bmatrix} \\ &= \begin{bmatrix} Q_y \mathbf{G} \\ Q_u \end{bmatrix} \Delta \mathbf{U} - \begin{bmatrix} Q_y (\mathbf{w}-\mathbf{f}) \\ 0 \end{bmatrix}. \end{aligned} \tag{41}$$

To minimize the cost function in (41), the solution of the algebraic equation (the control action) is derived as

$$\begin{bmatrix} Q_y \mathbf{G} \\ Q_u \end{bmatrix} \Delta \mathbf{U} - \begin{bmatrix} Q_y (\mathbf{w}-\mathbf{f}) \\ 0 \end{bmatrix} = 0. \tag{42}$$

Equation (42) is further presented as follows:

$$\overline{A} \Delta \mathbf{U} = \overline{b}, \tag{43}$$

where

$$\overline{A} = \begin{bmatrix} Q_y \mathbf{G} \\ Q_u \end{bmatrix}, \qquad \overline{b} = \begin{bmatrix} Q_y (\mathbf{w}-\mathbf{f}) \\ 0 \end{bmatrix}. \tag{44}$$

For solving (42), the QR decomposition [21] method based on the Householder algorithm [22, 23] is used to decompose matrix $\overline{A}$ as

$$\overline{A} = QR, \tag{45}$$

where $R$ is an upper triangular matrix and $Q$ is an orthogonal matrix as

$$\begin{aligned} R &= H_N H_{N-1} \cdots H_1 \overline{A}, \\ Q &= H_1 H_2 \cdots H_{N-1} \\ Q^T Q &= \left[ H_1 H_2 \cdots H_{N-1} \right]^{\mathrm{T}} \left[ H_1 H_2 \cdots H_{N-1} \right] = \mathbf{I}, \end{aligned} \tag{46}$$

where $H_i$ (i = 1, 2, ..., $N$) is a Householder matrix.

Considering (45), the solution of least squares in (43) is

$$\overline{A}^T \left( \overline{b} - \overline{A} \Delta \mathbf{U} \right) = 0 \Longrightarrow R^T Q^T \left( \overline{b} - \overline{A} \Delta \mathbf{U} \right) = 0 \tag{47}$$

as $Q^T \overline{A} = Q^T Q R = R$.

The preceding Equation (47) is rewritten as

$$\begin{aligned} & R^T Q^T \left( \overline{b} - \overline{A} \Delta \mathbf{U} \right) = 0 \\ & \Longrightarrow R^T \left( Q^T \overline{b} - Q^T \overline{A} \Delta \mathbf{U} \right) = 0 \\ & \Longrightarrow R^T \left( Q^T \overline{b} - R \Delta \mathbf{U} \right) = 0. \end{aligned} \tag{48}$$

Thus, the control signal is obtained as

$$\begin{aligned} & Q^T \overline{b} = R \Delta \mathbf{U} \Longrightarrow \Delta \mathbf{U} = R^{-1} Q^T \overline{b} \\ & \Longrightarrow \Delta \mathbf{U} = R^{-1} Q^T \begin{bmatrix} Q_y (\mathbf{w}-\mathbf{f}) \\ 0 \end{bmatrix}. \end{aligned} \tag{49}$$

Obtained vector $\Delta \mathbf{U}$ represents the control signal for the whole predictive horizon $\mathbf{N}$, and the actual control signal sent to plant is the first element in (49).

*2.4. State Estimator.* Consider the system in state space which is presented in (27) as follows:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + \Pi \omega(k) \\ y(k) &= Cx(k) + \nu(k), \end{aligned} \tag{50}$$

where $\omega$ and $\nu$ are sequences of white Gaussian noise with zero mean with known covariance as

$$E\{\omega(k)\} = E\{\nu(k)\} = 0. \tag{51}$$

The joint covariance matrix is

$$E\left\{ \begin{bmatrix} \omega(k) \\ \nu(k) \end{bmatrix} \begin{bmatrix} \omega^T(k) & \nu^T(k) \end{bmatrix} \right\} = \begin{bmatrix} Q_0 & 0 \\ 0 & R_0 \end{bmatrix}. \tag{52}$$

The initial state $x_0$ of a Gaussian random vector with mean presents $i$ as

$$E\{x_0\} = \hat{x}_0. \tag{53}$$

The covariance matrix is given by

$$E\left\{ (x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T \right\} = \Sigma_0. \tag{54}$$

The conditional probability density functions (pdf) represent the Gaussian pdf as

$$\mathcal{P}(x(k)) \sim \mathcal{N}(\hat{x}(k), P(k)), \tag{55}$$

where the state estimate $\hat{x}(k)$ and the covariance matrix $P(k)$ are presented as

$$\begin{aligned} \hat{x}(k) &= E\{x(k)\} \\ P(k) &= E\left\{(x(k) - \hat{x}(k))(x(k) - \hat{x}(k))^T\right\}. \end{aligned} \tag{56}$$

Considering (55), the filtering cycle states at the instant $k+1$ are presented as

$$\mathcal{P}(x(k+1)) \sim \mathcal{N}(\hat{x}(k+1), P(k+1)), \tag{57}$$

where

$$\begin{aligned} \hat{x}(k+1) &= E\{x(k+1)\} \\ P(k+1) &= E\left\{(x(k+1) - \hat{x}(k+1)) \times (x(k+1) - \hat{x}(k+1))^T\right\}. \end{aligned} \tag{58}$$

The Gaussian pdf is characterized by the mean and covariance matrix. Considering (27) by applying the mean value operator which is presented as

$$E\{x(k+1)\} = A \cdot E\{x(k)\} + B \cdot E\{u(k)\} + \Pi \cdot E\{\omega(k)\}. \tag{59}$$

From (55) and (57), the $\omega$ with zero mean is obtained as

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k). \tag{60}$$

The prediction error is defined as

$$\tilde{x}(k+1) = x(k+1) - \hat{x}(k+1), \tag{61}$$

which is replaced in expression of $x(k+1)$ and $\hat{x}(k+1)$. Equation (61) is rewritten as

$$\begin{aligned} \tilde{x}(k+1) &= Ax(k) + Bu(k) + \Pi\omega(k) \\ &\quad - A\hat{x}(k) - Bu(k) = A\tilde{x}(k) + \Pi\omega(k), \end{aligned} \tag{62}$$

where the filtering error is $\tilde{x}(k) = x(k) - \hat{x}(k)$.

Equation (59) is rewritten as follows:

$$E\left\{(\tilde{x}(k+1))(\tilde{x}(k+1))^T\right\} = AE\{\tilde{x}(k)\}A^T + \Pi Q_0 \Pi^T. \tag{63}$$

From (63), the notations in (56) and (58) result in

$$P(k+1) = AP(k)A^T + \Pi Q_0 \Pi^T. \tag{64}$$

The predictive estimated states of the system and the associated covariance matrix in (60) and (64) correspond to the optimal system state at the time instant $k+1$ before making observation at time instant $k$. The predicted measurement with a Gaussian pdf is given by

$$\hat{y}(k+1) = E\{y(k+1)\} = C\hat{x}(k+1). \tag{65}$$

The measurement prediction error $\tilde{y}(k+1) = y(k+1) - \hat{y}(k+1)$ is rewritten by replacing the $y$ and $\hat{y}$ as

$$\tilde{y}(k+1) = C\hat{x}(k+1) + \nu(k). \tag{66}$$

Considering (66), the covariance matrix is obtained as

$$P_{\tilde{y}}(k+1) = CP(k+1)C^T + R_0. \tag{67}$$

Multiply $x(k+1)$ on both sides of transpose in (66) as

$$\begin{aligned} x(k+1)\tilde{y}^T(k+1) &= x(k+1)C^T\hat{x}^T(k+1) \\ &\quad + x(k+1)\nu^T(k). \end{aligned} \tag{68}$$

Consider $E\{x(k+1)\tilde{y}^T(k+1)\} = P(k+1)C^T$ and evaluate the estimate state $\hat{x}$ at time instant $k+1$ as

$$\begin{aligned} \hat{x}(k+1) &= E\{x(k+1)\} \\ &\quad + E\left\{x(k+1), \tilde{y}^T(k+1)P_{\tilde{y}}^{-1}(k+1)\tilde{y}(k+1)\right\}. \end{aligned} \tag{69}$$

The optimal estimator to compute the state is based on a Kalman filter. The $j$-step ahead system output presented in (34) is

$$y(k+j) = CA^j\hat{x}(k) + \sum_{i=1}^{j} CA^{i-1}B\Delta u(k+j-i). \tag{70}$$

In (69), the estimation of the state vector $\hat{x}$ is obtained by the Kalman filter as

$$\begin{aligned} \hat{x}(k) &= A\hat{x}(k-1) + B\Delta u(k-1) + K_g(k) \\ &\quad \times \{y(k) - C[A\hat{x}(k-1) + B\Delta u(k-1)]\}, \end{aligned} \tag{71}$$

where $K_g$ is Kalman filter gain matrix represented in (72) to adapt the estimation of model states to measure the outputs from controlled system. Consider

$$K_g(k) = P(k-1)C^T\left[CP(k-1)C^T + R\right]^{-1}, \tag{72}$$

where the updated error covariance is $P(k) = [I - K_g(k)C]P(k-1)$.

Figure 2 shows the block diagram of the state estimator using Kalman filter to provide the estimate state for GPC. The Kalman filter is linear, discrete time, and finite dimension. The filter gain is independent of the system outputs. The error covariance and the filter gain are calculated before the filter is executed.

## 3. Result

*3.1. GPC Implementation in WiNCS.* Figure 3 shows the setup for the proposed GPC controller with Kalman state estimator implemented; control and sensor signals are encapsulated into packets to transmit in a wireless network environment emulated by NS2 (see the Appendix) under the WiNCS client/server architecture (IEEE 802.11b protocol) provided by PiccSIM (see the Appendix).

The simulation architecture is illustrated in Figure 4. The control system is in the server. The sensor, actuator, and plant are in the client under an emulated IEEE 802.11b wireless network.

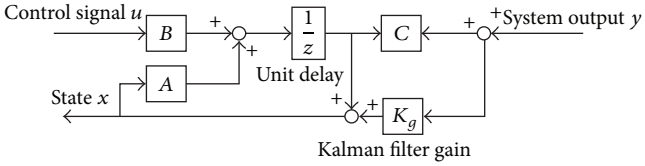



Figure 2: State estimator with Kalman filter.

*3.1.1. Controller Node.* The controller node in server module includes GPC controller, Kalman state estimator, sender, and receiver as shown in Figure 5. The square wave (period: 15 seconds, peaks: 1, and duration: from 0 to 100 seconds) is applied as the controller input.

*3.1.2. Actuator/Sensor Node.* The actuator/sensor node in the client module includes actuator, plant, sensor, sender, and receiver as shown in Figure 6. A white noise is applied as the disturbance to sensor measurement. The discrete state-space plant model is given by

$$x(k+1) = \begin{bmatrix} -0.0304 & -0.0998 \\ 0.1058 & 0.3476 \end{bmatrix} x(k) + \begin{bmatrix} 0.2116 \\ 1.3834 \end{bmatrix} u(k)$$
$$y(k) = \begin{bmatrix} 0 & 15.858 \end{bmatrix} x(k). \tag{73}$$

*3.2. Latency and Throughput on Actual Wireless Network Environment.* The performance of the networked data acquisition system for an actual small UAV (CoaX helicopter) is evaluated with the topology in Figure 7. The UAV sends images to a ground-based computer where the data is processed, and control packages are sent back to the UAV. Latency and throughput of the system are determined with the hrPing utility and the Interprocess Communication (IPC) library (see the Appendix).

Experiments were performed in two different indoor environments. The first is a room with a square ground plane and the second is a long corridor as shown in Figures 8 and 9. The line of sight connection of the transmission is uninterrupted by massive structures like concrete walls at all times.

The latency and throughput are tested for several distances between the CoaX helicopter and the stationary wireless router under two standards (IEEE 802.11 g and IEEE 802.11n). IEEE 802.11n provides longer range and higher throughput. The throughput of the wireless connection is determined for data packet sizes between 1 kB and 256 kB. The transfer rate for smaller packets is lower because the overhead of the transmission is dominant.

*3.2.1. Laboratory Environment.* The average latency of the IEEE 802.11 g connection (Figure 10) is constantly very low, and also the maximum values are stable over the whole distance range. Figure 11 shows the measurements for the connection conforming IEEE 802.11n. The mean latency is slightly higher, but the maximum latency does not significantly exceed the results of the previous measurements.

The throughput of the connection with the CoaX helicopter (IEEE 802.11 g), shown in Figure 12, is practically independent of the distance in this environment. In the measurements with the IEEE 802.11n connection (Figure 13), the throughput decreases with longer distance for bigger data packets.

*3.2.2. Corridor Environment.* The measurements are taken in the long corridor at distances from 10 to 70 meters. The latency of the connection with the CoaX helicopter (IEEE 802.11 g) is illustrated in Figure 14. The results show that the average latency is very low at around 1.2 ms, which is close to the minimum value. The worst case of the latency in the measurements is 20 ms. The measurements of a connection with the recently introduced standard IEEE 802.11n are shown in Figure 15. The average latency for distances from 30 meters and higher is low at around 1.2 ms. In close distance, however, the mean latency rises to 6.5 ms and the maximum value of 100 ms is comparatively high.

The results for the throughput of the connection with IEEE 802.11 g are depicted in Figure 16. The throughput decreases with rising distance up to 60 meters; however, the measurement for 70 meters gave a higher value. As expected, the connection with IEEE 802.11n achieved significantly higher transfer rates. The data in Figure 17 shows that the throughput decreases as the distance is increased.

*3.3. System Response with Random Delay and Packets Loss.* This experiment is to evaluate the capability of the proposed GPC with Kalman state estimator approach for compensating the random time delay in NCS. The random delay model is adopted to validate the GPC capability of compensating the delays. The system response is shown in Figure 18 with the random delay between 160 ms and 200 ms. The GPC parameters are listed in Table 1. The system response is stable but with the higher overshoot and the longer settling time. When the delay time closes to the sample time, the system response occurs with highly jitter.

The packet loss phenomenon is emulated by a switch with various packet loss rates. Figure 19 shows the system response with random delay between 120 ms and 160 ms and packets loss rate 5%. The higher packet loss rate causes the higher jitter of the system response (unstable).

*3.4. System Response with Network-Induced Delay in NS2.* The network-induced delay is generated by NS2 to evaluate if the system can follow the reference trajectory. Figure 20 shows the system response. Figures 21, 22, and 23 show the controller-to-actuator delay, sensor-to-controller delay, and sensor disturbance measurement, respectively. The simulation information is listed in Table 2. All of the packets were successfully transmitted without dropped packets. The system response successfully follows the reference trajectory as shown in Figure 20.

Figure 24 shows the system response with sample time 0.3 sec. Figures 25, 26, and 27 show the controller-to-actuator delay, sensor-to-controller delay, and sensor disturbance measurement, respectively. The simulation information is

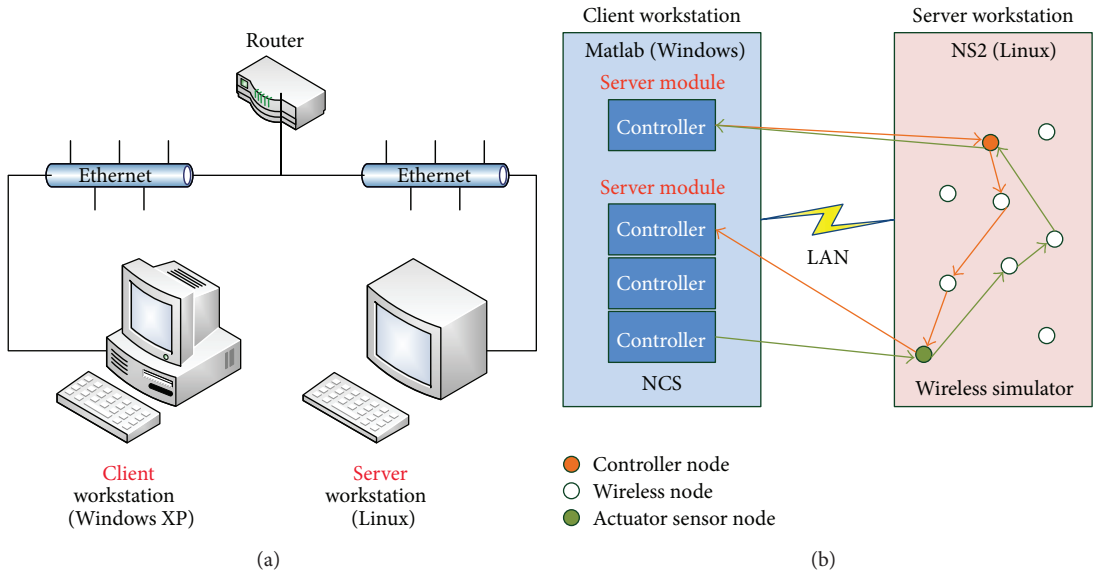



Figure 3: Experiment setup: (a) hardware and (b) software.

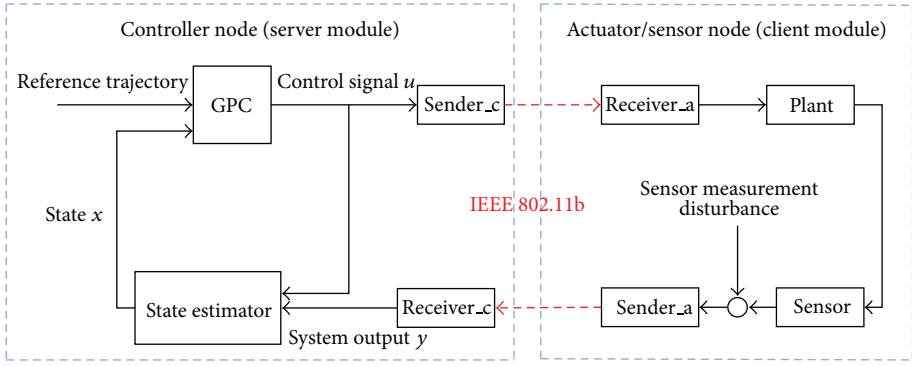



Figure 4: Structure of WiNCS.

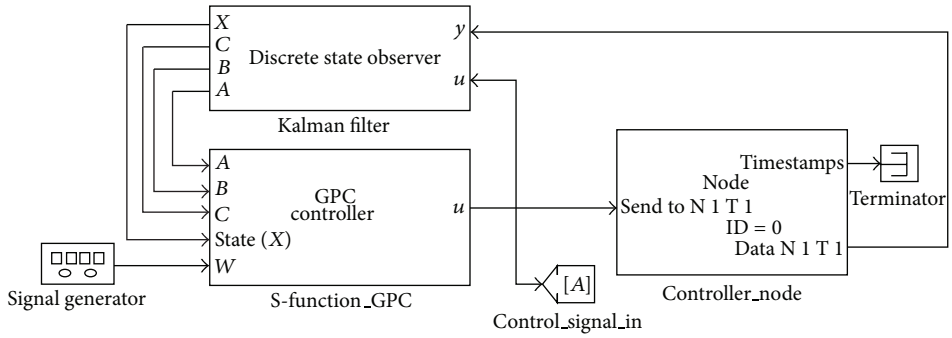



Figure 5: Simulation architecture of controller node.

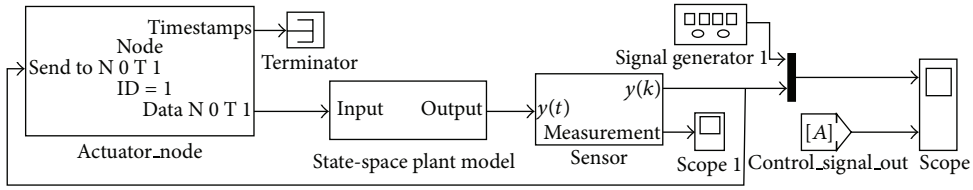



Figure 6: Simulation architecture of actuator/sensor node.

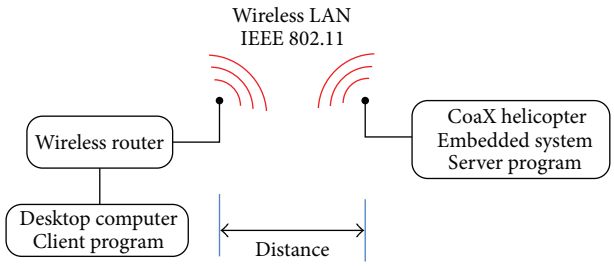



Figure 7: Topology of the wireless network for performance testing.

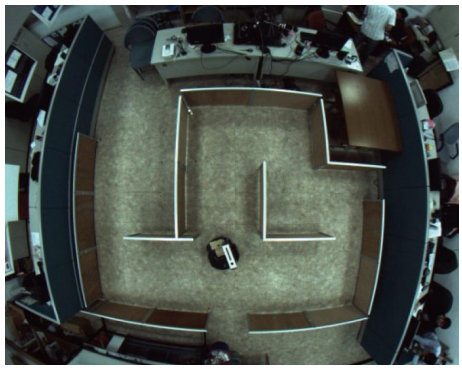


Figure 8: Square-shaped room for wireless network parameter measurements.

Table 1: Controller parameters in GPC.

| Controller parameter | Value |
|---|---|
| Predictive horizon | 6 sampling steps |
| Control horizon | 3 sampling steps |
| Sample time | 0.2 second |
| Penalization matrixes | $Q_y$: 0.8, $Q_u$: 0.01 |

Table 2: Simulation information of nodes with sample time 0.4 sec.

| Simulation specification | Value |
|---|---|
| Number of generated packets | 500 |
| Number of sent packets | 500 |
| Average packet size | 43.4973 |
| Number of sent bytes | 32960 |

listed in Table 3. The system is stable but with higher overshoot and the longer settling time. Different sample times affect the system performance. When the system is with the short sample time, the sender must generate more data packets. It might raise the packets loss rate and shorten the predictive horizon which might cause system unstablility.

Figure 28 shows the system response with sample time 0.2 sec. Figures 29, 30, and 31 show the controller-to-actuator delay, sensor-to-controller delay, and sensor disturbance measurement, respectively. The system response is highly jitter and with longer settling time. The simulation information is listed in Table 4. The system response already cannot follow the reference trajectory.

Figure 9: Long corridor for wireless network parameter measurements.

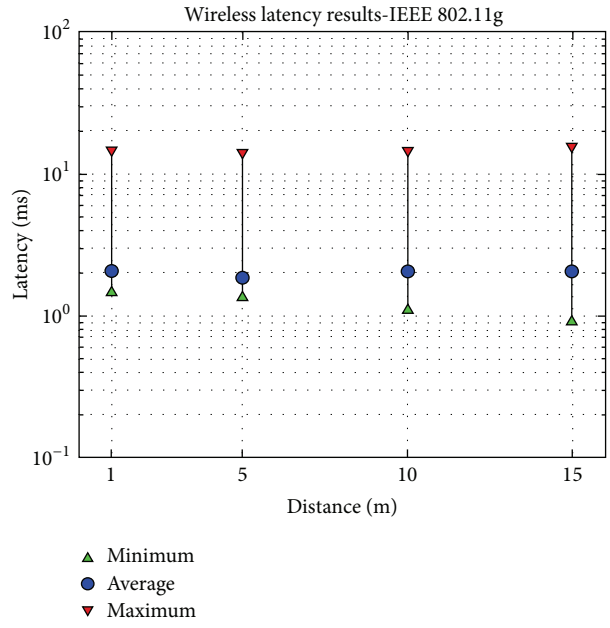



Figure 10: Latency measurements in room—IEEE 802.11 g.

Table 3: Simulation information of nodes with sample time 0.3 sec.

| Simulation information | Value |
|---|---|
| Number of generated packets | 668 |
| Number of sent packets | 668 |
| Average packet size | 43.7701 |
| Number of sent bytes | 44048 |

Figure 32 shows the system response with sample time 0.1 sec. The shorter sample time cause system unstable in WiNCS due to it shorten the predictive horizon.

## 4. Discussion

The WiNCS is implemented and evaluated with random delay, network-induced delay, and packets loss implementation by an analytical model and NS2. Result shows that the

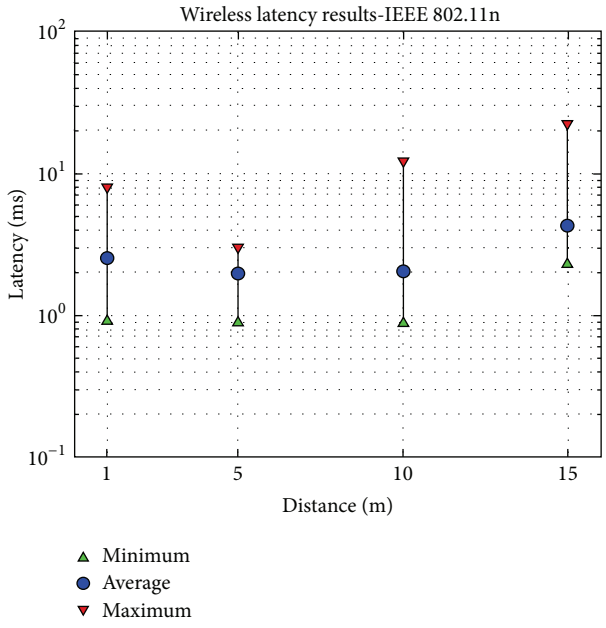



Figure 11: Latency measurements in room—IEEE 802.11n.

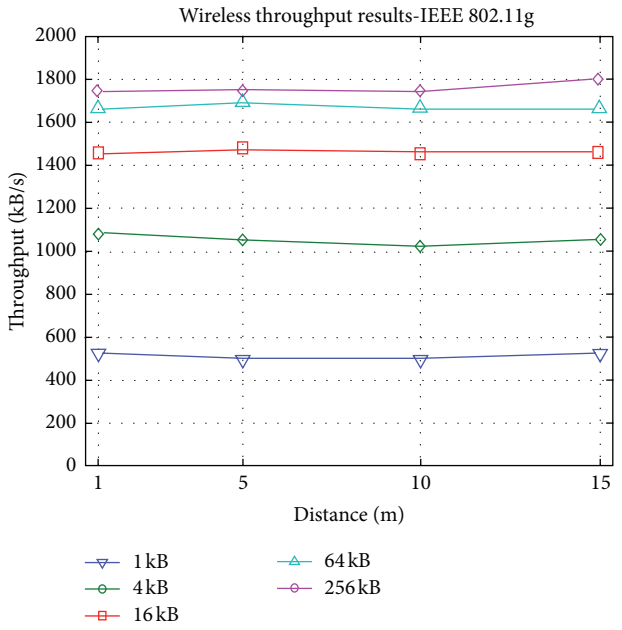



Figure 12: Throughput measurements in room—IEEE 802.11 g.

Table 4: Simulation information of nodes with sample time 0.2 sec.

| Simulation information | Value |
|---|---|
| Number of generated packets | 1002 |
| Number of sent packets | 1000 |
| Number of dropped packets | 2 |
| Average packet size | 44.1737 |
| Number of sent bytes | 65920 |
| Number of dropped bytes | 132 |

system response could follow the reference trajectory with the condition of the time delay under 160 ms and sample

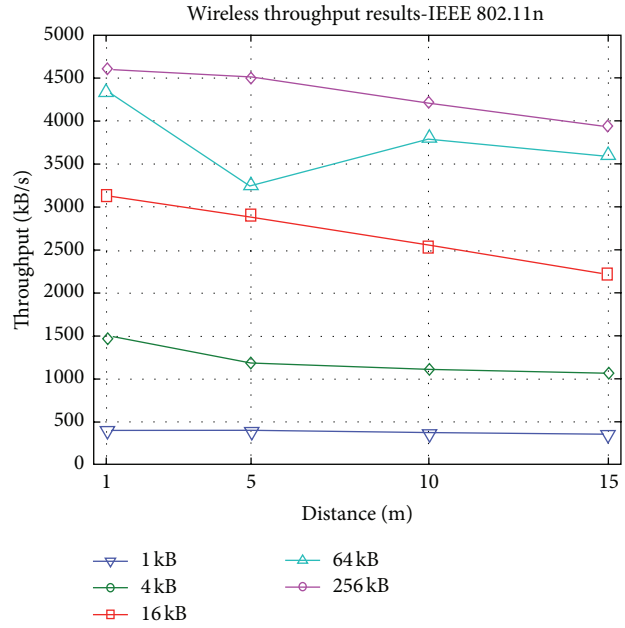



Figure 13: Throughput measurements in room—IEEE 802.11n.

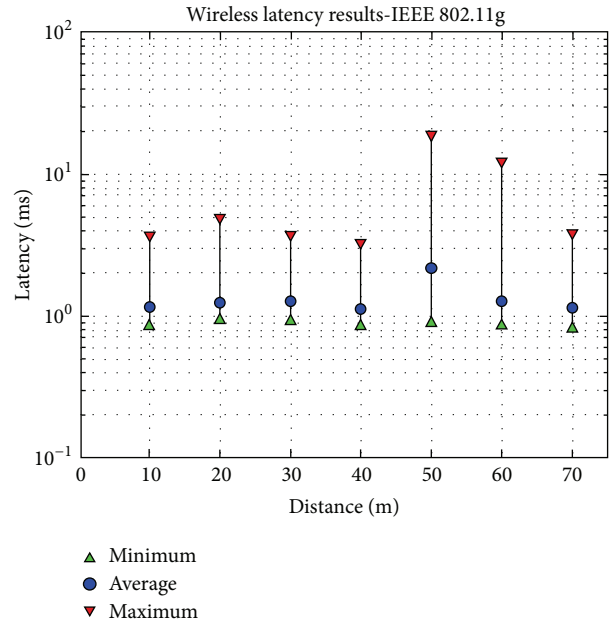



Figure 14: Latency measurements in corridor—IEEE 802.11 g.

time 0.2 s. The system response jittered when packet loss rate exceeded 10%.

The WiNCS is also simulated by NS2 with AOVD protocol. Result shows the system response with sample time = 0.4 sec. and number of transmission data packets = 500 and sample time = 0.3 sec. number of transmission data packets = 668 can follow the reference trajectory. The system response with sample time = 0.2 sec, number of transmission data packets = 1002, and number of packets loss = 2 has high jitter. The different sample times have various

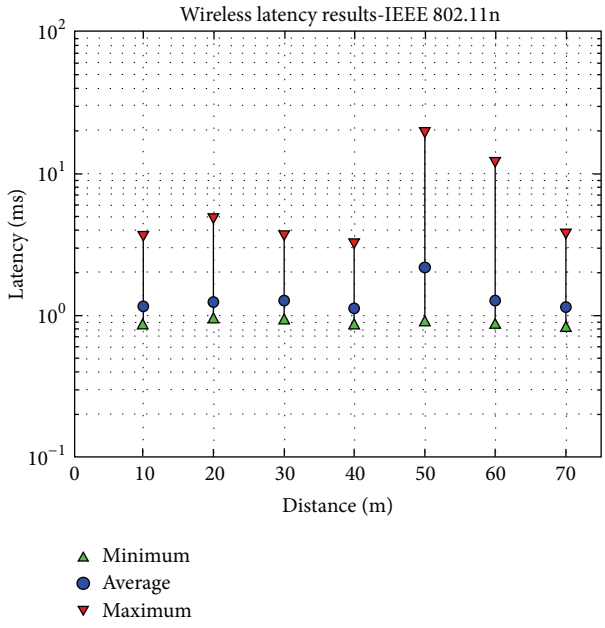



Figure 15: Latency measurements in corridor—IEEE 802.11n.

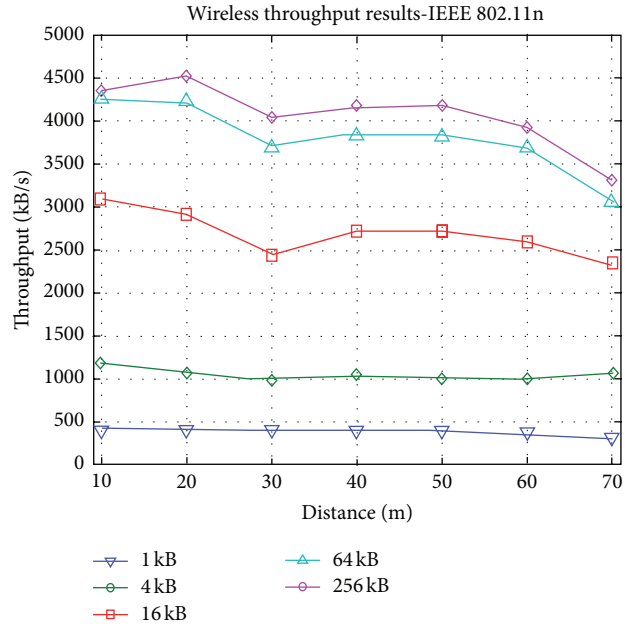



Figure 17: Throughput measurements in corridor—IEEE 802.11n.

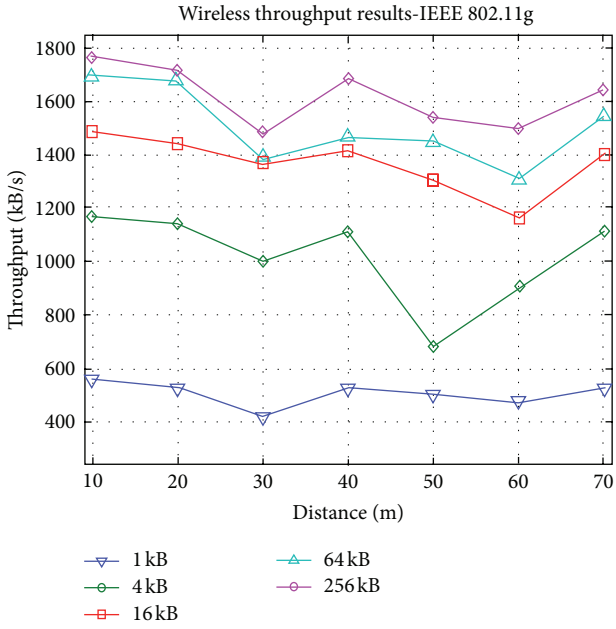



Figure 16: Throughput measurements in corridor—IEEE 802.11 g.

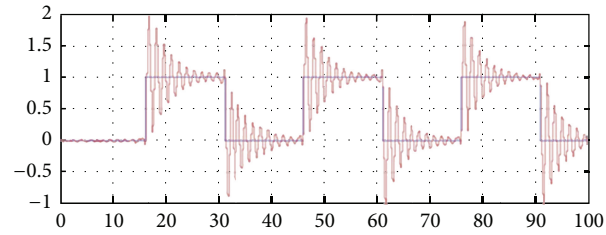


Figure 18: System response with random delay between 160 ms and 200 ms.

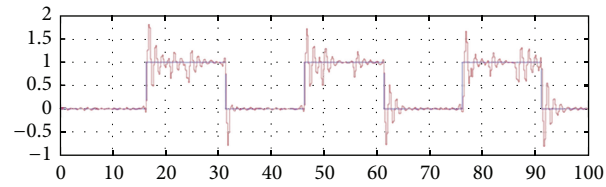


Figure 19: System response with random delay between 120 ms and 160 ms and packets loss rate 5%. $x$-axis: time (sec.) and $y$-axis: magnitude.

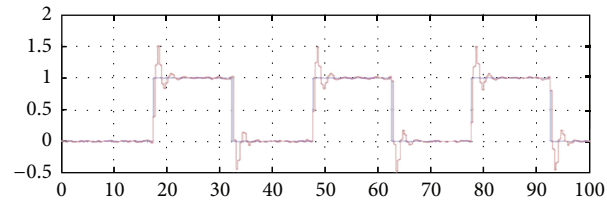


Figure 20: WiNCS response with sample time 0.4 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

numbers of transmission data packets. The larger the number of transmission data packets is, the easier the packets loss occurs.

It is easier to evaluate the performance of GPC in WiNCS when it is simulated via a model because the condition of random delay and packets loss rate can be controlled. WiNCS implemented in NS2 is closer to actual wireless network; therefore, the distance between control node and actuator/sensor node affects the network-induced delay and packets loss. This causes GPC performance to be difficult to

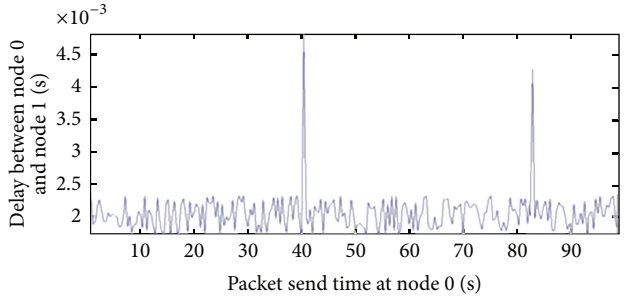



Figure 21: Controller-to-actuator delay with sample time 0.4 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

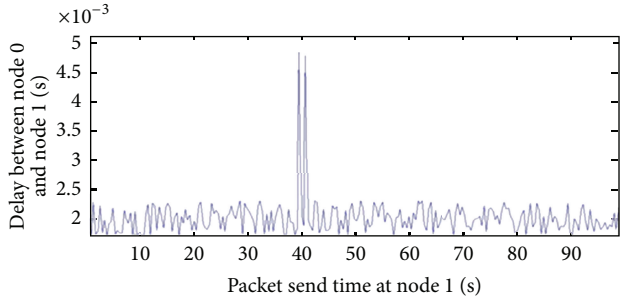



Figure 22: Sensor-to-controller delay with sample time 0.4 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

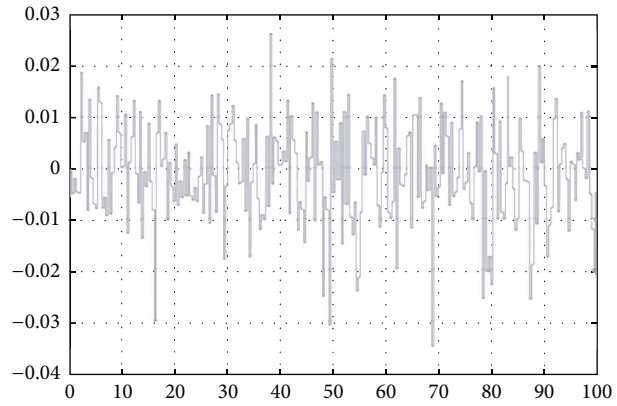


Figure 23: Sensor noise measurement with sample time 0.4 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

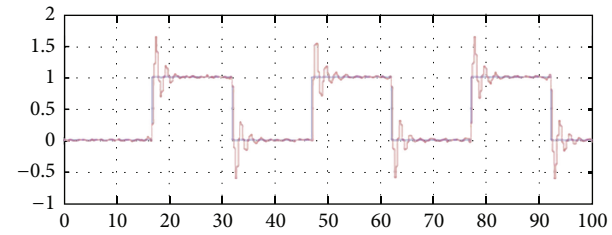


Figure 24: WiNCS response with sample time 0.3 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

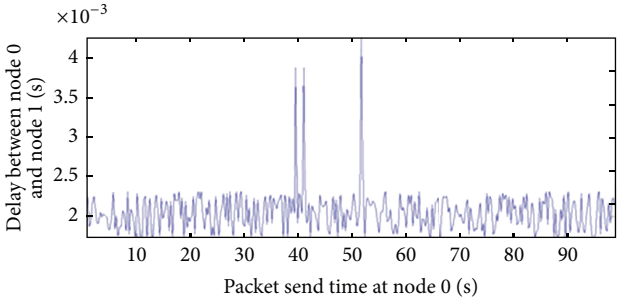



Figure 25: Controller-to-actuator delay with sample time 0.3 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

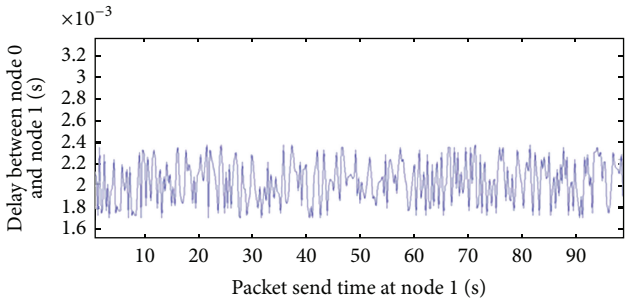



Figure 26: Sensor-to-controller delay with sample time 0.3 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

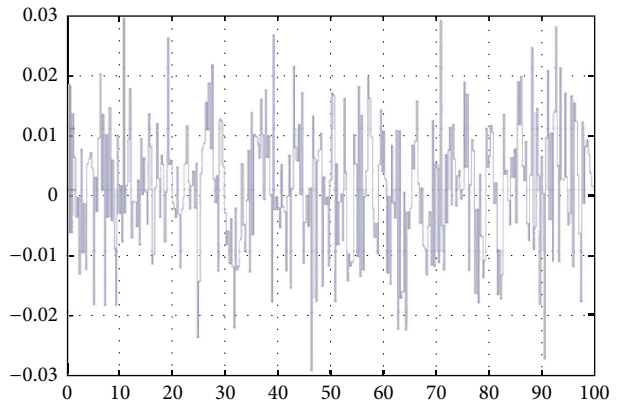


Figure 27: Sensor noise measurement with sample time 0.3 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

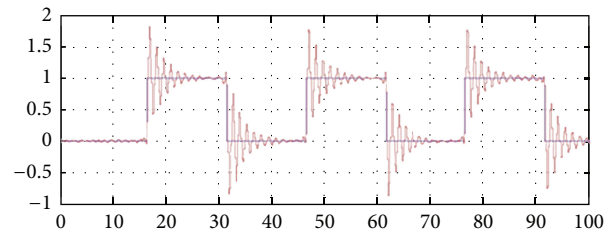


Figure 28: WiNCS response with sample time 0.2 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

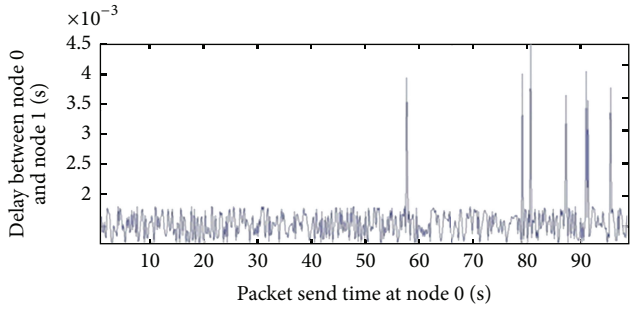



Figure 29: Controller-to-actuator delay with sample time 0.2 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

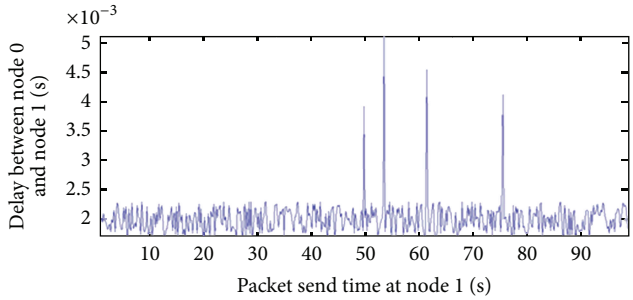



Figure 30: Sensor-to-controller delay with sample time 0.2 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

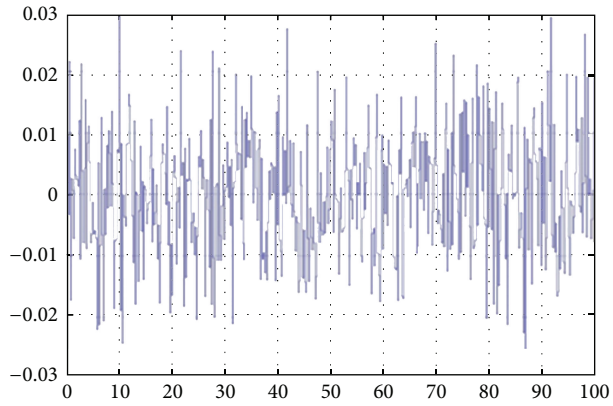



Figure 31: Sensor noise measurement with sample time 0.2 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

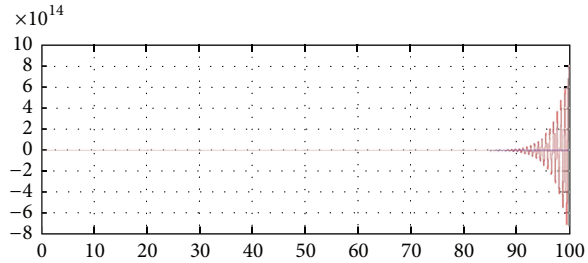



Figure 32: WiNCS response with sample time 0.1 sec. $x$-axis: time (sec.) and $y$-axis: magnitude.

analyse. The parameter in NS2 needs to be reestimated when being in the various wireless coverage environments. This also affects the simulation results when GPC is implemented in WiNCS.

The wireless networked control system results suggest that the latency is not directly related to the distance between sender and receiver. The mean values of the measurements are adequate for a closed-loop control system; however, the maximum values might have to be considered depending on the application. One reason for latency is the property that different wireless networks share the same frequency channel. Therefore, the density of wireless networks and the rate of traffic in close vicinity to the measurement setup determine the latency of the connection.

The throughput of the wireless connection according to standard IEEE 802.11 g is sufficient to transmit compressed images of size 320 × 240 at a rate of 30 frames per second up to a distance of 70 meters. The connection with the faster IEEE 802.11n standard allows transmitting the same images with smaller time delay or images with higher resolution at the same rate.

The measurements suggest that the concept of the wireless networked control system is applicable to autonomous navigation of small UAVs. The latency in a controlled environment is very low and does not inhibit real-time closed-loop control applications. The throughput of either standard IEEE 802.11 g or IEEE 802.11n is sufficient for transmitting compressed images of adequate resolution at a rate of 30 frames per second; however, the standard IEEE 802.11n is preferable for better performance.

The ideal environment for the wireless networked control system approach would be a closed room with strong walls to shield against interference from other networks. The limitations of the proposed system are the high sensitivity to interference from other wireless sources and the necessity of a line of sight connection without massive obstacles like concrete walls.

## 5. Conclusion

This paper proposed the GPC controller with Kalman state estimator in WiNCS based on PiccSIM platform. The packets are exchanged between the controller node and actuator/sensor node via wireless network IEEE 802.11b which is emulated by NS2. Although network-induced delay characteristics in the wireless communication network are difficult to model, this paper describes the main problems which might induce the time delay.

This paper simplifies complex architectures in the wireless communication network for analysis proposed with WiNCS which is simulated in NS2 using the two-ray ground model. First, this study implements WiNCS with the random delay to verify GPC controller capability with the Kalman state estimator to cope with time delay. Then, WiNCS is implemented with NS2 to present the effect of different sample times in the predictive horizon; that is, system performance decreases when sample time decreases. When WiNCS is implemented

with the sample time of 0.2 seconds, the packets start to drop, affecting the system performance.

We also propose the basic WiNCS simulation on a low-level control system. Realizing WiNCS requires not only improving the control algorithm to compensate for time delay, but also improving wireless communication performance. The time delay occurred when the packets are exchanged in the network. The algorithm for optimizing network performance communication is also important. This paper proposes the GPC algorithm for compensating time delay which focuses on improving control algorithm. In future, it is required to have further integration of the automation and control and network communication for realizing WiNCS.

The concept for a wireless networked control system was evaluated with latency and throughput measurements in different environments. The experiment setup conforming to the IEEE 802.11n standard achieves an average latency of 1.3 ms and a data throughput of 3.000 kB/s up to a distance of 70 m. The results demonstrate the feasibility of real-time closed-loop navigation control with the proposed concept.

The only significant limitations of the wireless networked control concept are the relatively short range of the wireless connections and the sensitivity to congest with other wireless devices. However, neither of them inhibits the effectiveness of the concept in the designated application to research in a controlled environment.

Modeling the network-induced delay and packets loss is extremely difficult. Although the NS2 provides the simulated communication network environment, it still simplifies the network condition comparing with real network devices. In this paper, the GPC implemented in WiNCS was verifying that it is a feasibility study.

In future work, a network estimator must be implemented and measure the network-induced delay and round trip time delay for tuning the controller parameters for increasing the performance of WiNCS. In addition, the soft computing technique (e.g., artificial neural network) can be applied to the predictive control algorithm to minimize the predictive error or tune the parameters of network estimator.

## Appendix

## Software Packages

*(i) PiccSIM (Platform for Integrated Communications and Control Design, Simulation, Implementation, and Modeling).* Helsinki University of Technology's PiccSIM is a simulation platform for WiNCS using Matlab/Simulink and NS2. It is a Matlab xPC-based target toolbox that is to transmit user datagram protocol (UDP) packet between Matlab/Simlink and NS2 (http://autsys.tkk.fi/en/Control/PiccSIM).

*(ii) NS2 (Network Simulator Version 2).* NS2 is a discrete event driven network simulator developed by UC Berkeley. NS2 has rich library of networks and protocols such as TCP and UDP, traffic behavior such as file transfer protocol (FTP), Telnet, CBR (constant bit rate), and VBR (variable bit rate), router queue management mechanism, and more (http://www.isi.edu/nsnam/ns/).

*(iii) CMU IPC (Interprocess Communication).* The Carnegie Mellon University's Interprocess Communication library provides flexible, efficient message passing between processes based on the TCP/IP protocol. It can transparently send and receive complex data structures, including lists and variable length arrays, using both anonymous "publish/subscribe" and "client/server" message-passing paradigms. A wide variety of languages and operating systems are supported (http://www.cs.cmu.edu/~ipc/).

*(iv) HrPing Utility.* The hrPing utility provides throughput and round trip delay measurements for computer networks. In contrast to other tools, it achieves higher resolution by timing the round trip delay in microseconds (http://www.cfos.de/ping/ping_e.htm).

## References


[1] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84–99, 2001.

[2] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*, vol. 11, no. 10, pp. 1099–1111, 2003.

[3] F.-Y. Wang and D. Liu, *Networked Control Systems:Theory and Applications*, Springer, London, UK, 2008.

[4] W.-L. D. Leung, R. Vanijjirattikhan, Z. Li et al., "Intelligent space with time sensitive applications," in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM '05)*, pp. 1413–1418, July 2005.

[5] G. C. Walsh, H. Ye, and L. G. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438–446, 2002.

[6] G. C. Walsh, O. Beldiman, and L. Bushnell, "Asymptotic behavior of networked control systems," in *Proceedings of the IEEE International Conference on Control Applications (CCA '99)*, pp. 1448–1453, August 1999.

[7] M.-Y. Chow and Y. Tipsuwan, "Gain adaptation of networked DC motor controllers based on QOS variations," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 5, pp. 936–943, 2003.

[8] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A new method for stabilization of networked control systems with random delays," *IEEE Transactions on Automatic Control*, vol. 50, no. 8, pp. 1177–1181, 2005.

[9] L. A. Montestruque and P. Antsaklis, "Stability of model-based networked control systems with time-varying transmission times," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1562–1572, 2004.

[10] D. Yue, Q.-L. Han, and C. Peng, "State feedback controller design of networked control systems," *IEEE Transactions on Circuits and Systems II*, vol. 51, no. 11, pp. 640–644, 2004.

[11] O. Beldiman, G. C. Walsh, and L. Bushnell, "Predictors for networked control systems," in *Proceedings of the American Control Conference*, vol. 4, pp. 2347–2351, June 2000.

[12] W. Aiguo, W. Dongqing, and L. Jin, "Application of generalized predictive control with neural network error correction in networked control system," in *Proceedings of the 5th World*

*Congress on Intelligent Control and Automation (WCICA '04)*, vol. 2, pp. 1386–1390, 2004.

[13] M. Dejun, F. Lei, and D. Guanzhong, "Research on generalized predictive control algorithm of networked control system," in *Proceedings of the 25th Chinese Control Conference (CCC '06)*, pp. 119–122, August 2006.

[14] P. L. Tang and C. W. de Silva, "Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 707–718, 2006.

[15] E. Witrant, P. G. Park, M. Johansson, C. Fischione, and K. H. Johansson, "Predictive control over wireless multi-hop networks," in *Proceedings of the 16th IEEE International Conference on Control Applications (CCA '07)*, pp. 1037–1042, Singapore, October 2007.

[16] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control. Part I. The basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[17] C. B. E. F. Camacho, *Model Predictive Control*, Springer, Berlin, Germany, 2004.

[18] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control. Part II. Extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160, 1987.

[19] D. Soloway, J. Shi, and A. Kelkar, *GPC-Based Stable Reconfigurable Control*, NASA Ames Research Center, 2004.

[20] K. Belda and J. Bohm, "Range-space predictive control for optimal robot motion," *International Journal of Circuits, Systems and Signal Processing*, vol. 1, no. 1, 2007.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 3rd edition, 2007.

[22] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins University, Baltimore, Md, USA, 3rd edition, 1996.

[23] W. Y. Yang, W. Cao, T. S. Chung, and J. Morris, *Applied Numerical Methods Using MATLAB*, John Wiley & Sons, Hoboken, NJ, USA, 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 505121, 8 pages
http://dx.doi.org/10.1155/2013/505121

*Research Article*

# An Address-Contention Approach Based on a Time-Division Multiplexing Mechanism for ZigBee Networks

**Tu-Liang Lin, Xian-Qun Zeng, and Hong-Yi Chang**

*Department of Management Information Systems, National Chiayi University, Chiayi City 60054, Taiwan*

Correspondence should be addressed to Hong-Yi Chang; alanc68@gmail.com

Received 28 June 2013; Revised 20 November 2013; Accepted 20 November 2013

Academic Editor: Chang Wu Yu



The rise of the Internet of Things promotes adopting wireless sensor networks (WSNs) in daily life. In WSNs, the ZigBee standard has gradually become the dominant communication protocol. ZigBee supports various network topologies, including tree structures. Regarding the address assignment of the tree topology, a distributed address assignment mechanism (DAAM) is specified by the ZigBee standard. Using DAAM yields a simple tree routing method; however, network parameter constraints cause the unpreventable orphan problem. Therefore, an innovative address contention approach was proposed based on a time-division multiplexing address assignment (TDMAA) mechanism, which utilizes the ZigBee beacon intervals for address contention. TDMAA outperforms conventional DAAMs in uneven node distributions, sometimes assigning 20% more addresses.

## 1. Introduction

Rapid advancements in sensors, embedded systems, and wireless communication technologies have fostered increasing research of wireless sensor networks (WSNs). The development of the Internet of Things is expected to ensure that WSNs are prevalent in future life. Numerous WSN-related studies have explored routing [1], localization [2], and sensor deployment and coverage [3]. Moreover, multiple applications have been developed such as those for industrial automation [4], health monitoring and prognosis [5], agricultural environment monitoring [6], and ecological observations [7].

Currently, ZigBee is the dominant communication protocol used in WSNs [8]. ZigBee is a wireless protocol that is designed for low power use and low data transmission rates. The ZigBee Alliance collaborates with the IEEE 802.15.4 committee to specify the complete ZigBee protocol stacks. The IEEE 802.15.4 defines the standard of the lower MAC/PHY layer, and the ZigBee Alliance designs the upper network and application layer standards. The IEEE 802.15.4 belongs to the category of personal area network (PAN) standards. ZigBee supports a large amount of sensor devices to work with flexible network topologies such as peer-to-peer, star, tree, and mesh topologies. Two types of devices, full-function devices (FFDs) and reduced-function devices (RFDs), are currently available in the marketplace. ZigBee networks comprise ZigBee coordinators (ZCs), ZigBee routers (ZRs), and ZigBee end devices (ZEDs). The ZC initiates a PAN and enables the ZRs and ZEDs to connect to the ZC. Because ZRs can route and forward packets, they can accept network join requests. In a tree topology, ZEDs can only function as leaf nodes. The ZC and ZRs are FFDs, whereas ZEDs are typically RFDs; however, FFDs can be converted into RFDs and function as ZEDs when necessary.

Two address assignment schemes, distributed address assignment mechanism (DAAM) and stochastic address assignment mechanism (SAAM), are specified in the ZigBee standard. DAAM works on a hierarchical tree structure; therefore the use of routing tables is not required. On the other hand, devices can independently and randomly select their addresses in SAAM and the network topology is not limited to tree structure in SAAM. Because the addressing in SAAM is not hierarchical, additional efforts are required to detect and resolve address conflicts. The tree-based routing method is not compatible with SAAM; therefore, an ad hoc on-demand distance vector- (AODV-) like protocol is used to find routes.

DAAM has some characteristics which are favorable in a WSN, such as efficient and simple routing algorithm. To achieve simplicity, DAAM preallocates the addresses by using three parameters, the maximal number of child devices ($C_m$), the maximal number of child routers ($R_m$), and the maximal depth of the tree structure ($L_m$). However, because of the restrictions of $C_m$, $R_m$, and $L_m$, new devices can be restricted from connecting to the network even when sufficient unused addresses remain. This phenomenon was termed the orphan problem and was proven to be a NP-complete problem [9].

In this paper, an innovative address contention method was proposed based on time-division multiplexing (TDM). The proposed time-division multiplexing address assignment (TDMAA) mechanism outperforms the standard ZigBee DAAM in the node join ratio. This research yielded two crucial contributions. First, TDM was introduced to the ZigBee address assignment scheme, providing a mechanism for scheduling devices based on the extensibility. Because of the limitations of the network parameters ($C_m$, $R_m$, and $L_m$), the network join order can greatly influence the join ratio. Second, a one-level loss function that can cooperate with TDM was designed to provide an address contention mechanism, allowing nodes that exhibit high extensibility to join the network before the nodes that exhibit low extensibility do. The simulation results showed that the proposed method outperformed DAAM in an uneven node distribution. In certain cases, the TDMAA mechanism outperformed DAAM by more than 20% regarding the node join ratio.

The remaining sections are organized as follows. Section 2 introduces background information and related research. Section 3 presents the proposed approach. Selected experimental results are given in Section 4, and the final section provides the conclusion and a discussion of future work.

## 2. Preliminaries

*2.1. Distributed Address Assignment Mechanism.* In DAAM systems, three parameters ($C_m$, $R_m$, and $L_m$) are assigned to the ZC. ZigBee standards specify that $C_m \geq R_m$, and there are two types of children devices (ZR and ZED); therefore, each ZC and ZR can have at most $R_m$ ZRs and $C_m$-$R_m$ ZEDs. The address assignment mechanism operates in a hierarchical manner and the addresses are assigned from top to bottom until the $L_m$ limit is reached. To illustrate, the ZC divides the entire address space into $R_m + 1$ blocks and the addresses of the first $R_m$ blocks are assigned to the child ZRs of the ZCs. The addresses of the final block are reserved for the ZEDs of the ZCs [8].

Each ZR including ZC can compute Cskip, a parameter that can be used to calculate the addresses of the child devices. The Cskip is defined as follows:

$$\text{Cskip}(d) = \begin{cases} 1 + C_m * (L_m - d - 1), & \text{if } R_m = 1 \\ \dfrac{1 + C_m - R_m - C_m * R_m^{L_m - d - 1}}{1 - R_m}, & \text{otherwise,} \end{cases} \quad (1)$$

where $d$ is the depth of the ZC/ZR.

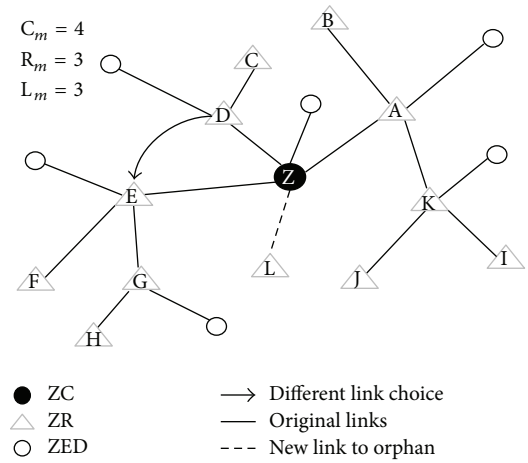



Figure 1: Orphan node example.

The ZC is at depth $d = 0$, and the address of ZC is 0. According to (1), the Cskip($d$) calculates the number of available addresses for each ZR child branch of a ZC/ZR parent at level $d$.

In DAAM, the assignment address can be calculated from the Cskip($d$). Suppose that the address of a ZR is Ap. The address of the $n$th ZR child is $\text{Ap} + (n-1) * \text{Cskip}(d) + 1$ and the address of the $m$th ZED child is $\text{Ap} + R_m * \text{Cskip}(d) + m$. For example, when ($C_m$, $R_m$, $L_m$) are set to (4, 4, 4), the address of the second ZR child of the ZC is $0 + (2-1) * \text{Cskip}(0) + 1 = 86$.

Two kinds of addresses are used in ZigBee, 16-bit short addresses and 64-bit long addresses. The long address is assigned at the time of device manufacture, and the short address is assigned when a device joins a PAN. Because short addresses are only 16 bits long, only $2^{16} = 65536$ addresses are available in a PAN. However, because of the limitations of $C_m$, $R_m$, and $L_m$, connecting to a network may be restricted before the exhaustion of these 65 536 addresses. For example, when ($C_m$, $R_m$, $L_m$) was set to (5, 5, 7), the maximal number of addresses was $\text{Cskip}(0) * R_m + C_m - R_m = 19\,531 * 5 = 97\,655$, which is greater than 65 535; however, when 65 535 devices were randomly deployed, there are still plenty of orphan nodes despite sufficient address space.

Figure 1 illustrates an orphan node example. The ($C_m$, $R_m$, $L_m$) was set to (4, 3, 3), and ZR D had two choices: to connect to ZC Z or to ZR E. If ZR D chose to connect to ZC Z, ZR D became an orphan node because of the limitation of $C_m = 4$; therefore, ZC Z lacks space for a new device. However, if ZR D chose to connect to ZR E, then ZC Z can have space to accept ZR L, so ZR L will not be an orphan node in this case.

*2.2. Tree Routing Protocol.* An advantage of using DAAM is the simplicity of its tree routing mechanism. The ZCs and ZRs can forward the packets along the tree without using a routing table. When a packet is received, a ZC or ZR first verifies whether the destination is itself or one of its child ZEDs. If the ZC or ZR is the destination, it accepts the packet. If one of its child ZEDs is the destination, the ZC or ZR forwards

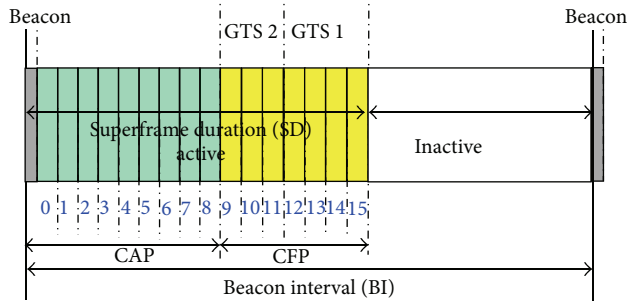



Figure 2: Superframe structure.

the packet to the child accordingly. Otherwise, the ZC or ZR determines the address of the relay ZR, Ar, according to (2). Assume that the address of ZC or ZR is Ac and its depth is $d$:

$$\mathrm{Ar} = \begin{cases} \mathrm{Ac}+1+\left\lfloor \dfrac{\mathrm{Ad}-(\mathrm{Ac}+1)}{\mathrm{Cskip}\,(d)} \right\rfloor & \\ \quad *\mathrm{Cskip}\,(d) & \text{if } \mathrm{Ac} < \mathrm{Ad} < \mathrm{Ac}+\mathrm{Cskip}\,(d-1) \\ \mathrm{Ap} & \text{otherwise,} \end{cases} \tag{2}$$

where Ad is the destination address and Ap is the address of the parent node.

*2.3. Beacon Interval.* The IEEE 802.15.4 can operate in two modes, (1) the beacon enabled mode and (2) non beacon-enabled mode. In nonbeacon-enabled mode, the medium is ruled by non-slotted CSMA/CA, and on the other hand the beacons are periodically broadcasted by the ZC to synchronize devices connecting to the PAN. In the paper, we utilized the time-division multiplexing mechanism to WSN address assignment, so we mainly focus on the beacon-enabled mode network in which the time can be divided into a sequence of beacon intervals.

In beacon-enabled network, a superframe structure is defined (see Figure 2) and the time between two subsequent beacons is the beacon interval (BI) [10]. The superframe duration (SD) is the active portion of the BI and active portion can be divided into two periods, contention access period (CAP) and contention free period (CFP). The time slots in CFP can be reserved and form guaranteed time slots (GTS). The BI and SD are determined by two parameters, the Beacon Order (BO) and superframe order (SO), and thus the low duty cycles can be configured by setting a small value of SO. In this research, BIs are utilized in a Time-Division Multiplexing manner to perform the address assignment.

*2.4. Related Work.* Since the orphan problem has been identified and has been proved to be a NP-complete problem, several heuristic solutions have been proposed for different purposes. After identifying the orphan problem as a NP-complete problem, the orphan problem was divided into two subproblems, the bounded degree and depth tree formation (BDDTF) problem and end device maximum matching (EDMM) problem [9]. A two-stage algorithm which spans the tree first and then prunes the tree to fit the ZigBee parameter definition was proposed for the BDDTF problem and a bipartite maximum matching algorithm for EDMM problem [9]. Although the proposed algorithm in [9] can reduce the orphan nodes, the approach is purely based on the graph theory and to make the approach comply with ZigBee standard is difficult.

Some solutions are based on the idea of address borrowing [11–14] to reduce the orphan nodes, but all these address borrowing-based approaches need to maintain extra information for the routing or extra overhead for tree reorganization. In [14], a single level address reorganization (SLAR) was proposed, and when a parent node did not have sufficient address space, the maximum depth is increased by one level to adjust the asymmetric nature of the tree to increase its capacity for new nodes. A hybrid address configuration (HAC) was proposed by [11], and a parent node can apply for extra addresses from the PAN coordinator when it does not have enough room for the new nodes. Although the address borrowing-based methods can increase the node join ration, they also increase the complexity in the routing protocol design since the borrowed addresses might need to be treated differently.

To the best of our knowledge, only [15] focused on the tree formation instead of tree adjustment. In [15], three mechanisms, 2DAAM (2 layer DAAM), LDAAM (Location aware DAAM), and (RSSI DAAM) received signal strength indicator DAAM, were proposed for address assignment and, among the tree mechanisms, RSSI DAAM which uses the received signal strength to calculate the distance between two nodes has good performance with least cost and thus is the most recommended mechanism by the authors. Although RSSI DAAM showed better performance than the original DAAM, a scheduling approach complying with ZigBee is missing. In this paper, we compared our approach TDAAM with the original DAAM and RSSI DAAM [15].

There is some research which is focused on special types of network topology. In [16, 17], the authors showed that the original DAAM performs poorly in the long-thin topology in which a number of linear paths of nodes connect to each other. A cluster-based approach was proposed for the address assignment of long-thin topology in [17] and the nodes are divided into 4 types, coordinator, cluster head node, bridge node, and network node. Therefore, it is not uncommon to design a new address assignment mechanism that can work properly in a special type of topology.

*2.5. Problem Statement.* In the standard ZigBee DAAM, a ZC or ZR periodically broadcasts beacon frames when operating in the beacon-enabled mode. The beacon frame contains the information regarding the PAN to which the ZC or ZR belongs. A nearby device can scan for beacons and discover the PAN. To join the PAN, the device sends association requests to the ZC or ZR. If the association request is approved, the device receives an association response frame that contains the assigned address from the ZC or ZR. A device can join the PAN at any time if it can obtain an address from a ZC or ZR.

The standard DAAM device-joining procedure was not designed to prioritize certain devices. Therefore, devices

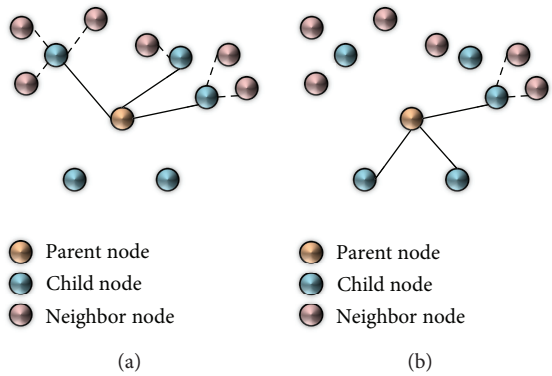



Figure 3: Two different join scenarios.

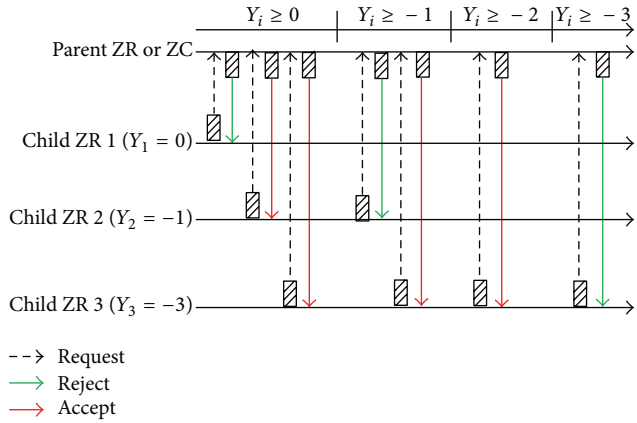



Figure 4: TDM address assignment (TDMAA) example.

that exhibit low extensibility (e.g., devices that comprise no surrounding neighbors) may join the PAN, thereby impeding devices that exhibit high extensibility (e.g., devices that comprise sufficient surrounding neighbors) from joining the network. Figure 3 shows two device joining scenarios. Suppose that the $(C_m, R_m, L_m)$ is set to $(3, 3, 2)$. Few orphan nodes remain if the three most extensible nodes are occupied first (Figure 3(a)); however, more orphan nodes remain if the three least extensible nodes are occupied first (Figure 3(b)).

Since the standard ZigBee DAAM is not designed to prioritize the device join order according to the extensibility, two problems of the original DAAM are identified as follows.

*Problem 1.* Given a set of ZigBee devices, how is the extensibility of each device quantified?

*Problem 2.* Given a set of neighbor devices with quantified extensible scores, how is the join order scheduled?

## 3. Time-Division Multiplexing Address Assignment

A formal definition of the time-division multiplexing address assignment (TDMAA) mechanism consisted of two parts, an address-loss function and a scheduling approach. The address-loss function is for quantifying the extensibility of a device and the scheduling approach is for selecting the highly extensible devices. To the best of our knowledge, this is the first time that time-division multiplexing mechanism is applied to WSN address assignment.

Given the network parameter $R_m$ and a node with the number of surrounding neighbors $X_i$, the address-loss function should be able to calculate the possible address loss. Therefore, in order to quantify the extensibility of the nodes, the following one-level address-loss function was designed:

$$Y_i = \begin{cases} 0 & \text{if } X_i - 1 - R_m \geq 0 \\ X_i - 1 - R_m & \text{otherwise,} \end{cases} \quad (3)$$

where $X_i$ is the number of surrounding neighbors of the $i$th device.

When $Y_i = 0$, there are enough children for the next level, so no address loss exists in the next level if this node is allowed to connect to the network. If $Y_i = -n$, $n$ addresses are lost in the next level. Supposing that the depth of the parent node is $d$, if the parent node allows a child node of $Y_i = k$ to connect to the network, the total address loss would be $|k| * \text{Cskip}(d + 1)$. In this research, only one level of address loss was explored, but (3) has the potential to be extended to calculate multiple levels of address losses.

The algorithm for calculating the $Y_i$ in each node is simple. Suppose that each node can broadcast a Hello message to the neighbors within the communication distance. Given a warmup time for Hello message exchange, each node can simply calculate the number of Hello messages it received to determine the number of neighbors $X_i$ and $Y_i$ can be calculated from $X_i$. When a node wants to join the PAN, the node must send the joining request with $Y_i$ value to the prospective parents asking to join the PAN. The prospective parents maintain a currently allowed $Y_i$ value, and if the $Y_i$ value sent by the prospective child is greater than the currently allowed $Y_i$, the prospective child is allowed to join the PAN.

To preserve tree extensibility, devices on the tree must first allow highly extensible nodes to connect. Therefore, the study proposes using a TDMAA approach at the beginning stage of ZigBee network formation. The range of currently allowed $Y_i$ value is calculated. For example, when $R_m = 3$, the range of currently allowed $Y_i$ value is from $-3$ to 0. Regarding the first beacon interval, only devices where $Y_i \geq 0$ are allowed to connect to the prospective parent ZC or ZR. The currently allowed $Y_i$ value decreases by 1 after a beacon interval elapses until reaching the minimal value. The devices where $Y_i \geq -1$ are allowed to connect to the network at the second beacon interval. The rules are applied until the minimal $Y_i$ value is reached and the ZC or ZR subsequently enables all neighboring devices to join the network (this is how the original ZigBee standard DAAM operates). Figure 4 shows how the TDMAA works for $Y_i$ from 0 to $-3$.

The TDM (time-division multiplexing) mechanism which can operate in accordance with the beacon interval forms the second part of our TDMAA. The currently allowed $Y_i$ value decreases by 1 after a beacon interval elapses until reaching the minimal value. When the currently allowed $Y_i$

```
Algorithm TDMAA
Input Network Parameters C_m, R_m, L_m
Begin
        Integer X_i, Y_i;
        Send_Message(Hello);
        X_i = 0;
        While (Still Warm Up)
        {
           If Receive_Message(Hello);
           X_i = X_i + 1
        }
        If ((X_i − 1 − R_m) > 0)
           Y_i = 0;
        Else
           Y_i = X_i − 1 − R_m;
        While (Not Connected to PAN)
        {
           Send_Message(Join_Request(Y_i));
        }
        Currently_allowed_Y_i = 0;
        While (Connected to PAN)
        {
           If (Receive_Message(Join_Request(Y_i))
                 If (Currently_allowed_Y_i > Y_i && C_m, R_m, L_m still have capacity)
                    Send_Message(Accept);
                 Else
                    Send_Message(Reject);
           If (Receive_Message(Beacon) && Currently_allowed_Y_i is not minimum)
                 Currently_allowed_Y_i = Currently_allowed_Y_i − 1;
           If (The network parameters C_m, R_m, L_m indicate no capacity)
              Exit;
        }
End
```

Algorithm 1: The algorithm of time-division multiplexing address assignment.

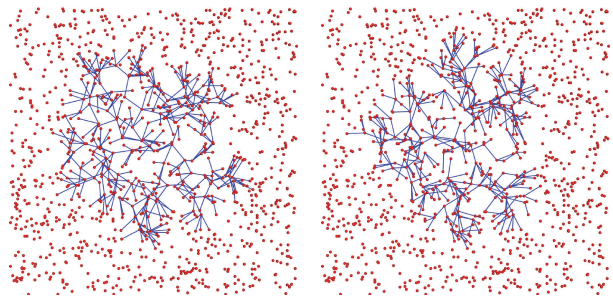
reaches the minimal value, it works as the original DAAM. Algorithm 1 is the algorithm for TDMAA.

In this paper, two performance metrics, number of orphan nodes and the join ratio, are employed to measure the address a preassignment schemes. The join ratio is calculated using the following formula:

$$\text{Join ratio} = \frac{\text{Number of ZC} + \text{Number of ZRs joined} + \text{Number of ZEDs joined}}{\text{Total number of devices}}. \quad (4)$$

## 4. Experimental Results

A simulator was implemented using JAVA and a molecular structure visualization tool called PyMOL [18] was used to render the output results. The experiments were performed on two node distribution setups, random and uneven node distribution setups. The nodes were placed in an area of 400 $m^2$. In the random distribution setup, the coordinators of the nodes were randomly assigned. In the uneven distribution setup, two high-node-density squares were added to the top right and bottom left areas. The communication range was set as 30 m in all experiments. For simplicity, let $C_m$ be equal to $R_m$ in all network parameter configurations, and the values of $L_m$ were set according to the $C_m$ and $R_m$ values, achieving equivalent total node numbers in both experimental setups.



(a) (b)

Figure 5: The tree structures of DAAM (a) and TDMAA (b) from random node distribution.

The TDMAA did not substantially improve in the random node distribution setup. Figure 5 shows the tree structures of the DAAM and TDMAA obtained using the random node distribution setup. The spreading areas of the two trees are fairly similar. This is probably because of the ineffectiveness

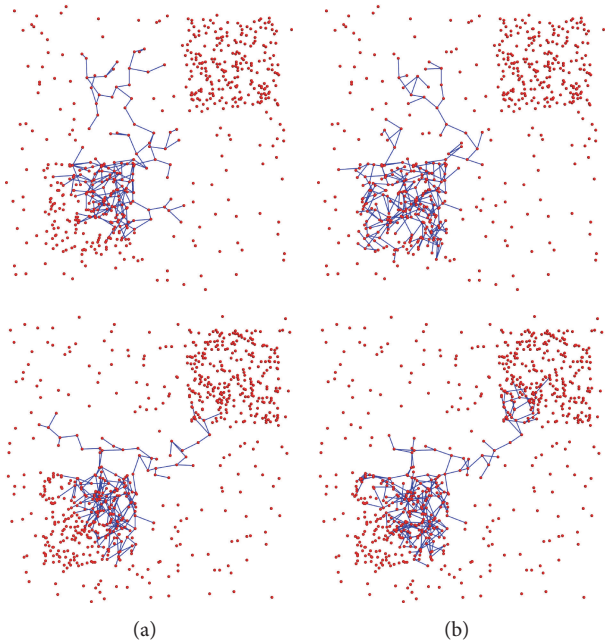



Figure 6: The tree structures of DAAM (a) and TDMAA (b) from unevenly distributed nodes.

of the one-level address-loss function in randomness. Future studies should pursue improvements of the address-loss function.

Although the TDMAA did not outperform the DAAM in the random distribution setup, it performed much more satisfactorily than the DAAM did in the uneven distribution setup. The TDMAA located larger high density areas than the DAAM did, eventually yielding higher address assignment ratios. Figure 6 shows the results of the DAAM and the TDMAA in the uneven node distribution setup. The network parameter (2, 2, 14) was used to perform the simulations.

The RSSI DAAM from [15] was also implemented to provide some comparisons with our method, and the name of RSSI DAAM is shortened as RSSI in the remaining text. Figure 7 shows the number of orphan nodes in the DAAM, TDMAA, and RSSI results, using various network parameter configurations in the uneven node distribution setup. The low $C_m$ and $R_m$ configurations tended to yield more satisfactory levels of performances than the high $C_m$ and $R_m$ configurations did. Since it is crucial to choose wisely when $C_m/R_m$ is small and TDMAA can choose high extensible nodes first, so fewer orphans remained. RSSI only performs slightly better than TDMAA in two configurations, (3, 3, 9) and (5, 5, 6), and RSSI has the worst performance in both the lowest and highest $C_m$ and $R_m$ configurations.

Figure 8 shows the trends in the join ratios of the DAAM, TDMAA, and RSSI compared with the number of nodes in the uneven distribution setup. The join ratios were calculated using formula (4). A substantial difference was observed between the DAAM and TDMAA results when the number of total nodes reached 1200 and the join ratio of the DAAM dropped to less than 10%, whereas the join ratio of the TDMAA remained at 38%. Because the heuristic of

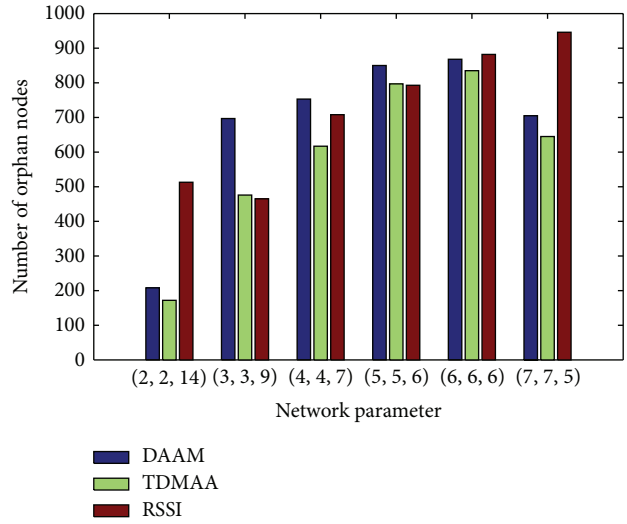



Figure 7: The number of orphan nodes of DAAM, TDMAA, and RSSI with different network parameters.

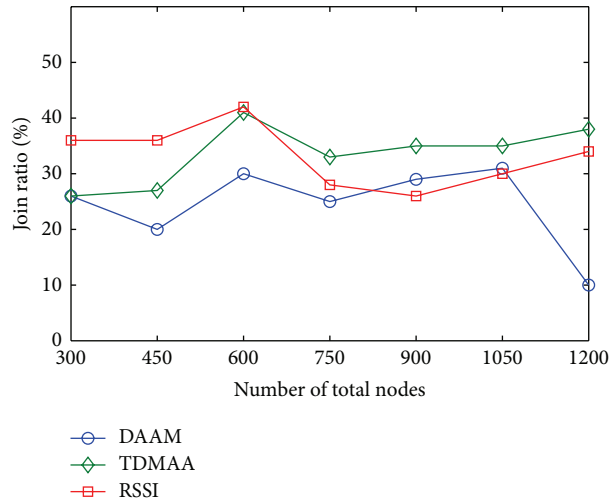



Figure 8: The join ratio comparison of the DAAM, TDMAA, and RSSI.

the TDMAA was to choose nodes that possessed numerous surrounding neighbors, the TDMAA tended to locate high density areas. Therefore, when the node density increased, the TDMAA performed steadily; however, the DAAM formed linear chains around the neighbors and $L_m$ was quickly consumed; thus, the new nodes were blocked from entering the network.

The RSSI approach performs better than TDMAA when the number of total nodes is small and performs worse than TDMAA when the number of total nodes is larges in Figure 8. Given the same 400 $m^2$ area for all settings, the large number of total nodes means dense node coverage and the small number of total nodes means sparse node coverage. Therefore, RSSI is more suitable for spare node distribution and TDMAA is more suitable for dense node distribution.

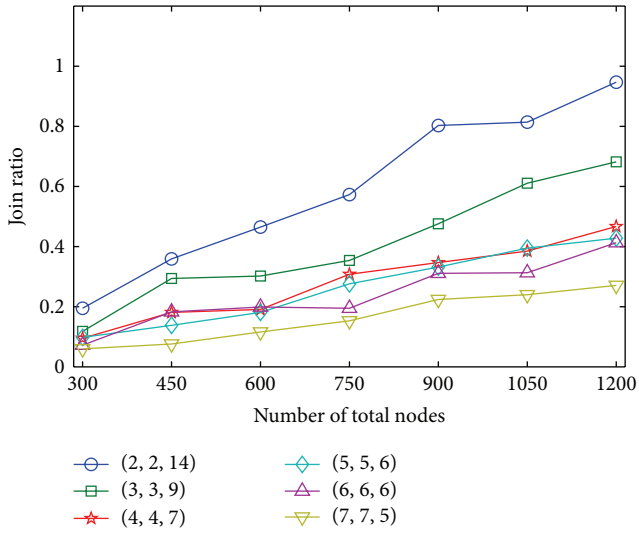



FIGURE 9: The join ratio of the TDMAA method with different network parameters.

The performance of TDMAA and RSSI is nearly the same when the total number of nodes equals 600.

Figure 9 displays the join ratios of the TDMAA, using various network parameter configurations and numbers of total nodes. The join ratios of all network parameter configurations steadily increased as the number of total nodes increased. The (2, 2, 14) configuration exhibited superior performance level among all the network parameter configurations. Therefore, the performance level of TDMAA was improved when the density of the sensor nodes was high and the number of branches ($R_m$) was small in the uneven distribution setup.

## 5. Conclusion and Future Work

DAAM, the standard ZigBee address assignment scheme, provides a simple means to assign addresses to tree-connected devices and enables the forwarding of packets along the tree structure without using a routing table. However, because of the constraints of the network parameters ($C_m$, $R_m$, and $L_m$), the existence of orphan nodes is unavoidable. The standard device-joining procedure was not designed to prioritize specific devices; therefore, certain devices that exhibit low extensibility may join the PAN first, preventing devices that exhibit high extensibility from joining the network. In this paper, a one-level address-loss function was designed and a TDMAA approach was proposed to schedule the new device joining sequence according to the values of the address-loss function. The TDMAA uses periodic beacon intervals to determine the acceptance of a new device. Each new device computes the address-loss function and proposes an association request to a ZC or ZR with the value of the address-loss function. The ZC or ZR can only accept certain values at certain beacon intervals; therefore, nodes that exhibit high extensibilities can join the network before nodes that exhibit low extensibilities can do so. The address-loss function can be extended and applied to multiple layers.

Two types of node distributions were conducted, random and uneven. The TDMAA did not outperform DAAM in the random distribution setup; this was attributed to the ineffectiveness of the one-level address-loss function. Future research exploring effective address-loss functions should be conducted. The TDMAA performs more satisfactorily than the DAAM does in the uneven distribution setup. The simulation results show that the join ratio of DAAM rapidly drops to less than 10% when the number of nodes increases to 1200 in the uneven distribution setup, but the join ratio of TDMAA remains steady at 38%.

## Acknowledgment

This research was supported by funding from the National Science Council of Taiwan (NSC101-2218-E-415-002-MY2 and NSC102-2221-E-415-012).

## References

[1] A. N. Eghbali, N. T. Javan, and M. Dehghan, "EDAP: an efficient data-gathering protocol for wireless sensor networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 7, no. 1, pp. 12–24, 2011.

[2] G. Mao, B. Fidan, and B. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[3] T.-W. Sung and C.-S. Yang, "A cell-based sensor deployment strategy with improved coverage for mobility-assisted hybrid wireless sensor networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 5, no. 3, pp. 189–198, 2010.

[4] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.

[5] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *Systems, Man, and Cybernetics C*, vol. 40, no. 1, pp. 1–12, 2010.

[6] B. Cai, Q. Bi, F. Li, D. Wang, Y. Yang, and C. Yuan, "Research and design of agricultural environment monitoring system based on ZigBee sensor network," *Acta Agriculturae Jiangxi*, vol. 11, p. 050, 2010.

[7] H. Yamamoto, S. Uchiyama, M. Yamamoto, K. Nakamura, and K. Yamazaki, "Development of sensor network for ecology observation of seabirds," *IEICE Transactions on Information and Systems*, vol. 95, no. 2, pp. 532–539, 2012.

[8] Z. Alliance, *Zigbee Specification*, 2008.

[9] M.-S. Pan, C.-H. Tsai, and Y.-C. Tseng, "The orphan problem in ZigBee networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 11, pp. 1573–1584, 2009.

[10] S. C. Ergen, *ZigBee/IEEE 802.15.4 Summary*, vol. 10, UC Berkeley, 2004.

[11] L.-H. Yen and W.-T. Tsai, "The room shortage problem of tree-based ZigBee/IEEE 802.15.4 wireless networks," *Computer Communications*, vol. 33, no. 4, pp. 454–462, 2010.

[12] Z. Ren, P. Li, J. Fang, H. Li, and Q. Chen, "SBA: an efficient algorithm for address assignment in ZigBee networks," *Wireless Personal Communications*, vol. 71, no. 1, pp. 719–734, 2012.

[13] Y.-R. Li, H.-B. Shi, and B.-Y. Tang, "Address assignment and routing protocol for large-scale uneven wireless sensor networks," in *Proceedings of the International Symposium on Computer Network and Multimedia Technology (CNMT '09)*, pp. 1–4, Wuhan, China, 2009.

[14] D. Giri and U. K. Roy, "Single level address reorganization in wireless personal area network," in *Proceedings of the 4th International Conference on Computers and Devices for Communication (CODEC '09)*, pp. 1–4, 2009.

[15] K. H. Tsai, W. Hung, and S. J. Sun, "On the tree formation of wireless sensor networks," in *Proceedings of the International Conference on Systems and Informatics (ICSAI '12)*, pp. 891–894, 2012.

[16] M.-S. Pan, H.-W. Fang, Y.-C. Liu, and Y.-C. Tseng, "Address assignment and routing schemes for ZigBee-based long-thin wireless sensor networks," in *Proceedings of the IEEE Vehicular Technology Conference (VTC '08)*, pp. 173–177, 2008.

[17] M. S. Pan and Y. C. Tseng, "ZigBee-based long-thin wireless sensor networks: address assignment and routing schemes," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 12, pp. 147–156, 2013.

[18] Schrodinger LLC, *The PyMOL Molecular Graphics System, Version 1. 5. 0. 4*, Schrodinger, 2010.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 790794, 11 pages
http://dx.doi.org/10.1155/2013/790794

*Research Article*

# A Coverage Hole Healing Strategy with Awareness of Data Delivery Time in Wireless Sensor Networks

**Fu-Tian Lin,[1,2] Tien-Wen Sung,[1,3] and Chu-Sing Yang[1]**

[1] *Institute of Computer and Communication Engineering, Department of Electrical Engineering, National Cheng Kung University, Tainan 70101, Taiwan*
[2] *Department of Electrical Engineering, Tung Fang Design University, Kaohsiung 82941, Taiwan*
[3] *Department of Network Multimedia Design, Hsing Kuo University, Tainan 70963, Taiwan*

Correspondence should be addressed to Chu-Sing Yang; csyang@ee.ncku.edu.tw

Received 24 June 2013; Accepted 10 November 2013

Academic Editor: Naveen Chilamkurti



An efficient wireless sensor network (WSN) should maintain full sensing coverage and topology connectivity within its sensing field. Once holes occur due to failed sensors, the functionality and performance of the WSN will be affected. In this work, the proposed hole healing strategy aims to shorten the delivery hop count by a dynamic hole healing process in order to improve the data delivery time while maintaining the coverage ratio and topology connectivity. The criteria used to determine which hole should have the highest priority for healing in the next round include the weighted distance, angle magnitude, and depth of the hole. This study proposes a mobile robot, operating within the WSN, which carries redundant sensors to patch the holes by an optimum healing path. This path is determined based on the proposed EDPS (equally divided path selection) algorithm. Simulation results show the superiority of the proposed hole healing scheme over other general methods.

## 1. Introduction

A wireless sensor network (WSN) [1] consists of many sensing devices; however, each device only has limited resources. Under the constraint of these limited resources, these wireless devices have limited abilities, such as environment information sensing around itself, simplified computation capacity, and shorter communication distances [2, 3]. Even so, by cooperating with neighboring devices, nodes can form a network topology for WSN applications, which is important for guaranteeing the integrity and reliability of the environmental information in sensor network applications. Therefore, WSN applications usually deploy a large number of devices to guarantee full coverage and topology connectivity [4, 5]. After the wireless devices have been deployed, a network topology will be constructed via the devices' autonomous exchange of information with neighboring devices. WSN applications always require complete coverage [6], but as the abilities of wireless devices and their battery energy are limited, device failures are unavoidable. A single failed device is called a hole, which induces a coverage hole and reduces the system efficiency. Additionally, when the conveyance path is established, environment information will be sent along this path to data receiver center. In order to prevent the data transmission being interrupted because of node failures, two methods for ensuring data transmission are to reroute, or to patch nodes. The latter method (patching nodes for hole healing) is chosen in this study. The hole-healing approach was chosen over a rerouting approach for two main reasons. The first is that once the sensors near the sink, or a considerable number of sensors malfunction, either the rerouted path will become much longer, or it will not be possible to find a path for data transmission by a rerouting approach. The second reason is that a rerouting approach cannot avoid or solve the problem of coverage ratio decreases. A hole healing approach, however, will be able to keep routing paths and coverage ratios properly maintained for data transmission and the sensing tasks in a WSN. This study therefore used a hole healing approach and focused on the design of the healing strategy and mechanism. In addition, it was assumed that the WSN would always have enough reserved backup sensors for the hole healing process.

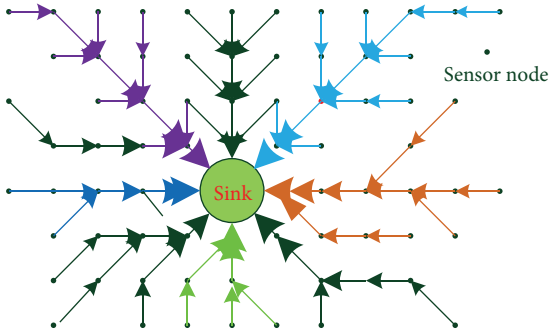



Figure 1: The multi-to-one data flow concept.

This study proposes a healing scheme that is not only able to solve the above problems but is also able to improve the delay time for data transmission. Some WSN applications require a real time response, for example, the protection of bridges, railway track security, traffic surveillance, rescue operations, or battlefield reconstruction. When an event occurs, the information must be sent to the data center as soon as possible. Therefore, the propagation time of event response in the surveillance field is an important factor in WSN applications. Data is conveyed directly or indirectly to a data receiver center, and if a node fails during this process, it will either induce data loss or postpone the propagation delay time. Therefore, failed nodes must be healed in order to increase the coverage rate and to rebuild the topology connectivity, the latter of which is more important. Before the hole is healed, carefully calculating which hole must be selected for healing, with special attention paid to the data propagation time.

This paper considers an asymmetrical WSN topology. This consists of a data collection center called a sink, and many wireless sensor devices called nodes. Nodes are responsible for collecting information on their surrounding environment and forwarding data from neighbor nodes. The information is transmitted to the data center by one-hop or multihop methods. The information flow structure is called multi-to-one topology, which differs from peer-to-peer architecture. In the multi-to-one architecture, node loading is very unbalanced, because sensors need to sense environmental information and forward others' data to the sink. Therefore, a node closer to the sink will have a heavier traffic load. As shown in Figure 1, the direction and thickness of the arrows represent the data flow direction and volume, respectively. Consequently, coverage holes occur due to node failures owing to the limited node energy being exhausted, or other accidental events. As a result, data will be lost, the conveyance time will be increased, and the topology may be separated. Because of the lack of infrastructure backbone, WSN applications usually use an ad hoc network mode. When nodes do not work together, however, it will be difficult to maintain a topology connection. To make matters worse, when they do communicate with each other, they use up their limited energy, causing node failure. As a result, there will be communication problems.

The remainder of this paper is organized as follows. Section 2 introduces relevant works. Section 3 makes system model statements concerning assumptions made in this study. Section 4 presents the proposed healing scheme algorithm. Section 5 summarizes the simulation results, and Section 6 draws conclusions.

## 2. Related Works

In WSN applications, connectivity and coverage are two major problems. Specifically, increasing the coverage rate and maintaining the topology connectivity can reduce node energy consumption and prolong the network lifetime. Based on limited energy resources and the nature of wireless networks, node failure is unavoidable. Therefore, reactivating failed nodes is an effective method to keep WSN applications working continuously. To do this, it is necessary to know the positions of failed nodes or holes. The Voronoi diagram concept [7] is used to detect a coverage hole and to calculate the hole size [8, 9] or improve the coverage [10, 11]. The article in [12] proposed a triangular oriented diagram to detect holes and calculate the hole size and then determine which mobile sensor should have priority for healing. Additionally, articles in [13–15] proposed a method for discovering failed nodes and estimating hole positions by periodically collecting and updating the surveillance information at the data receiver center.

In terms of hole deployment, authors in [16] utilize redundant dynamic nodes that have been distributed within the field and use dynamic scheme technology to fix the holes, based on a centralized computer to drive the appropriate dynamic nodes to holes. In [17] a TSP-Delaunay redeployment method was proposed for repairing holes. Also, a BFNP (best fit node policy) method [18] was proposed for fixing holes by activating inactive nodes around holes. In [19], the bidding contention protocol was proposed, which enables redundant nodes in dense deployment areas to move to sparsely covered areas during the contention process. Unlike redundant nodes, static nodes are able to identify field holes. Active redundant nodes must derive hole information and calculate the healing efficiency before moving, a method which may induce erroneous movement and result in wasted energy. Besides the problem of energy consumption, the speed of the mobile nodes' movement is a critical issue which will affect the efficiency of the healing scheme; see [20] for further details. Authors in [21] considered a small group of mobile robots operating in WSNs and proposed the randomized robot-assisted relocation of static sensors (R3S2) and a grid-based variant (G-R3S2) algorithm for hole identification and coverage repair. However, these studies did not consider the delay time induced by data transmission in the hole healing process.

## 3. Preliminary

This study considers a centralized WSN architecture, with nodes uniformly deployed within the $W * D$ field, and with prior knowledge of the boundary nodes [22]. The field can be

imagined as an $r \times r$ virtual grid, each grid possibly placed by some sensor nodes. If no nodes are located within a grid it is called a vacant grid. The coordinate of a grid is defined at the grid center. This study also assumes that all nodes are homogeneous; that is to say, all sensors have the same hardware and software features, so all nodes in a field have the same protocol stack. In addition, it was assumed that the WSN would always have enough reserved backup sensors for the hole healing process.

Vacant grids are areas uncovered by sensors; adjacent and continuously vacant grids are called coverage holes (*HOLEs*, which also include single vacant grids), and each *HOLE* is assigned a serial number $i$. For ease of reference, $H_i$ stands for the $i$th *HOLE*, where $i \leq m$, $m$ is the total number of *HOLEs* within the field of interest. This paper also uses $n_i$ to represent the total number of boundary nodes in the $i$th *HOLE*, $B_{ij}$ ($j \leq n_i$), the boundary nodes of $H_i$. $R$ is the communication radius of a node with an omni-directional radio communication mode. In order to guarantee that neighboring nodes are within communication range of each other, the relationship between $r$ and $R$ is as shown in Figure 2, where $r$ is the grid edge, and its length is equal to $R/\sqrt{2}$.

With the centralized WSN architecture, the sink knows the *HOLE* distribution in the surveillance field. To explain the node locations, Cartesian coordinates $(x, y)$ are used. For example, $H_i(x, y)$ are the coordinates of $H_i$'s position, which is the gravitational mass center of the *HOLE*. $x$ and $y$ are computed as follows:

$$x = \frac{1}{n_i}\left(\sum_{j=1}^{n_i} B_{ij}x_j\right), \qquad y = \frac{1}{n_i}\left(\sum_{j=1}^{n_i} B_{ij}y_j\right). \tag{1}$$

To make the paper more readable, the term "*HOLE* angle" is defined as $\Theta(H_i)$, which is from the sink viewpoint and is the minimum angle of $H_i$. A *HOLE* is presented as a graph topology with virtual grids, denoted as $G_i = (V_i, E_i)$, where $V_i$ is the set of vertices, which include the vacant grids and the boundary nodes on the $i$th *HOLE*, and $E_i$ is the set of edges within *HOLE* $H_i$. Specifically, the edge is the connection of the adjacent vertices with a distance less than or equal to $R$.

To maintain the topology connectivity and ensure the integrity of the environment sensing information with a shortened postponement time for data transmission, a *HOLE* healing scheme is proposed. Specifically, this scheme has two phases, the first is to select one *HOLE* while at the same time calculating a healing path, which is based on the three properties of the *HOLE*, and the second phase is to drive a robot to patch the *HOLE* along the healing path decided by the first phase. The robot is equipped with a positioning component and has the ability to move to the appointed location. Terrain obstacles are not considered, and the mobile robot only carries out and finishes an established healing path each time; the shortest route is determined by the Dijkstra algorithm.

## 4. Hole Healing Scheme

The purpose of the proposed scheme is to select a *HOLE* and to include one optimized healing path for this *HOLE*. In

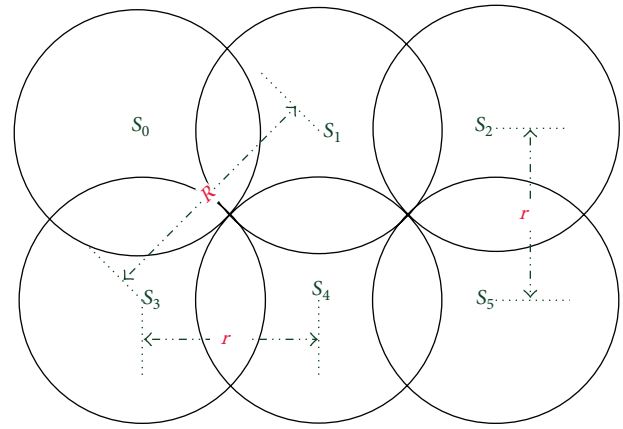



Figure 2: The distance relationship of nodes under grid topology.

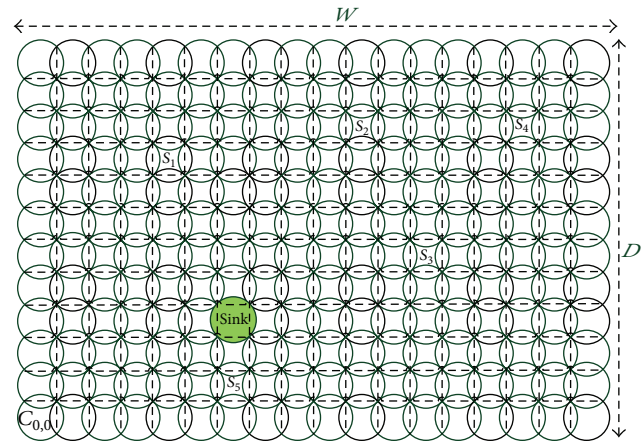



Figure 3: The ruled grid structure with Cartesian coordinates.

WSN applications, when the connected topology is disrupted, the connection must be restored for data transmission to continue; otherwise the environment information will either be lost or delayed at the sink. The results of the hole healing scheme are not only expected to increase the coverage rate but to also improve the data propagation time. In fact, when a *HOLE* is healed, the sequence will affect the system efficiency, especially used in real time applications.

Optimizing the area coverage and topology connectivity is a very complicated process. First, the Cartesian coordinates are used to describe the location of the nodes as described above. The position of the node is expressed as $C(x, y)$, abbreviated to $C_{x,y}$, and the origin $C_{0,0}$ is located at the lower left hand corner. Figure 3 illustrates the relative positions. A sink is responsible for collecting the environmental information, for example, the locations $S1$, $S2$, $S3$, $S4$, and $S5$, representing the sensor nodes with the coordinates $C_{8,4}$, $C_{10,9}$, $C_{12,5}$, $C_{15,9}$, and $C_{6,1}$, respectively. Also, according to the earlier definition, the node position is always located at the center of the grid.

To calculate the sequence of *HOLEs* to be healed, three weighted metrics are considered, which are the properties of a *HOLE*. The first metric is the *HOLE* angle, $W(\Theta(H_i))$, the size of the angle indicating the degree to which data transmission

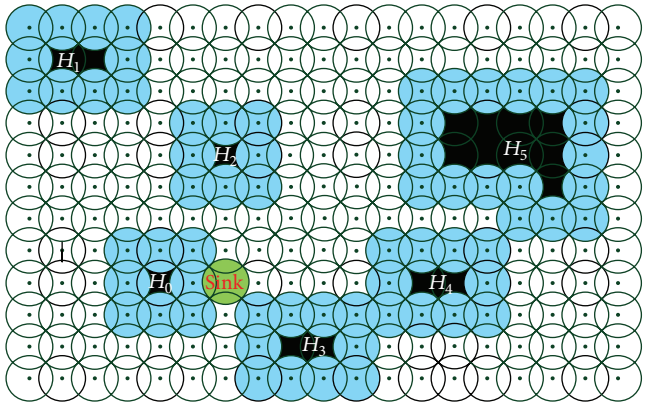



Figure 4: The *HOLE* distribution in the surveillance field.

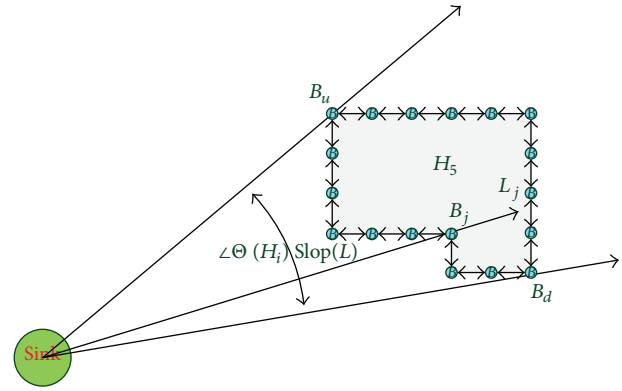



Figure 5: Illustration of the relationship between $B_u$, $B_d$, and $\Theta(H_i)$.

is hindered by the *HOLE*; the second metric is the distance between the sink and the *HOLE*, denoted as $W(len(H_i))$; and the last metric is the depth of the *HOLE*, expressed as $W(deep(H_i))$, where $i = 1, 2, 3, \ldots, m$. The details of these three weighted metrics are introduced in Sections 4.1, 4.2, and 4.3, respectively.

*4.1. Discussion of the HOLE Angle.* The authors of [23] proposed the boundary node detection scheme, which uses the LVP (localized Voronoi polygons) algorithm to find border nodes. By using the algorithm, the nodes will know whether they are a border node or not by neighboring nodes' information only, and these nodes will transmit their status information to the sink. Figure 4 shows a typical example of *HOLE* distribution in the field, which results from the sink collecting information in the surveillance field.

However, the angle can provide information on how the data flow will be hindered by a *HOLE*. Specifically, if a hole occurs in the middle of a transmission path, this will cause a conveyance interruption, and more time for rerouting will be required in order to maintain the data transmission path. The data are conveyed to the sink from all sides, and the data transmission influence is directly proportional to the *HOLE* width; that is to say, a larger *HOLE* width will mean more interference. In this study, the *HOLE* angle is considered equivalent to the *HOLE* width.

Before computing $\Theta(H_i)$, a ray expressed by [Sink, $B_i$) which starts at the sink and extends toward border node $B_i$ is denoted by $\overrightarrow{L_i}$. According to the property of the trigonometric function, the method proposed in article [24] is used to solve $\Theta(H_i)$ over 180 degree problems. Here it must be noted that all angles are measured either in a clockwise direction, or a counter-clockwise direction.

The algorithm applies the slope of the ray to find the upper border node $B_u$ and the lower border node $B_d$ in $H_i$, as well as the angle $\Theta(H_i)$. The algorithm discussed in Algorithm 1 is used in a typical example shown in Figure 5. The metric of the *HOLE* angle $W(\Theta(H_i))$ can be calculated by

$$W\left(\Theta\left(H_i\right)\right) = \frac{\Theta\left(H_i\right)}{180}. \tag{2}$$

*4.2. The Distance Metric $W(len(H_i))$.* The proposed hole healing scheme is based on a centralized topology. Obviously, the environmental information is conveyed to the sink, and if any *HOLE* exits within the field, then the data transmission will be hindered. Influence on the fixed size of a *HOLE* owing to the different influences of various positions depends on how close the sink is. If the *HOLE* position is closer to the sink, then the interference will be more serious, and if the *HOLE* is far away from the sink, the interference will be much less. Therefore, *HOLE* healing must first consider the influence intensity of data transmission. Figure 6 illustrates the *HOLE* distribution by Euclidean distance in the field, where $d_i$ denotes the Euclidean distance between the sink and the *HOLE*:

$$d_i = \sqrt{\left(H_i \cdot x - S_0 \cdot x\right)^2 + \left(H_i \cdot y - S_0 \cdot y\right)^2}. \tag{3}$$

The metric of the *HOLE* distance is denoted as $W(len(H_i))$. If the efficiency of data propagation time that considers the *HOLE* distance only is discussed, it is easy to see that the shorter the distance is, the shorter the propagation time will be. Conversely, if the *HOLE* position is far away from the sink, or even located at the field border, then the influence on the data transmission will be less, most of which is the coverage rate problem. So let this metric be inversely proportional to the reciprocal of the *HOLE* distance between the *HOLE* and the sink:

$$W\left(len\left(H_i\right)\right) = \frac{1}{d_i}. \tag{4}$$

*4.3. The Metric of the HOLE Depth, $W(deep(H_i))$.* When the *HOLE* area is small, it is clear which *HOLE* must be selected, with the path healing determined by the metrics $W(\Theta(H_i))$ and $W(len(H_i))$ only. However, when the *HOLE* area is large, the *HOLE* area effects must be considered. Under such circumstances, not only the shorter delay time for the data transmission must be taken into account, but topology disconnection must also be avoided. To do this, the third factor should be considered—the *HOLE* depth.

To determine the depth of the *HOLE*, the nearest border nodes $B_{\text{near}}$ and farthest border nodes $B_{\text{far}}$ must first be calculated, based on the distance from the sink. In order to

**Input:** $H_i$, $B_{ij}$
**Output**: $\angle\Theta(H_i)$, $B_u$ and $B_d$
**Initial**:
Let $B_u \leftarrow H_i(x, y)$, $B_d \leftarrow H_i(x, y)$
$\text{slope}_{\min} \leftarrow \text{slope}(H_i)$, $\text{slope}_{\max} \leftarrow \text{slope}(H_i)$
(1) For $j$ from 1 to $n$ do
(2) $\overrightarrow{L_J} \leftarrow [\text{Sink}, B_{ij})$.
(3) if $\text{slope}(\overrightarrow{L_J}) < \text{slope}_{\min}$
(4) $B_d = B_{ij}$, $\text{slope}_{\min} = \text{slope}(\overrightarrow{L_J})$
(5) else if $\text{slope}(\overrightarrow{L_J}) > \text{slope}_{\max}$
(6) $B_u = B_{ij}$, $\text{slope}_{\max} = \text{slope}(\overrightarrow{L_J})$
(7) End For

$$(8)\ \emptyset(H_i) = \begin{cases} 2\sin^{-1}\left(\dfrac{R/2}{\left|\overrightarrow{\text{Sink}H_i}\right|}\right); & \text{if } \text{slope}_{\min} = \text{slope}_{\max} \\ \cos^{-1}\left(\dfrac{\overrightarrow{\text{Sink}B_u} \; n\overrightarrow{\text{Sink}B_d}}{\left|\overrightarrow{\text{Sink}B_u}\right| \times \left|\overrightarrow{\text{Sink}B_d}\right|}\right); & \text{otherwise} \end{cases}$$

$$(9)\ \Theta(H_i) = \begin{cases} \emptyset(H_i); & \text{if } \emptyset(H_i) \in [0^\circ, 180^\circ] \\ 360^\circ - \emptyset(H_i); & \text{if } \emptyset(H_i) \in [180^\circ, 360^\circ] \end{cases}$$

Algorithm 1: The algorithm of *HOLE* angle calculation.

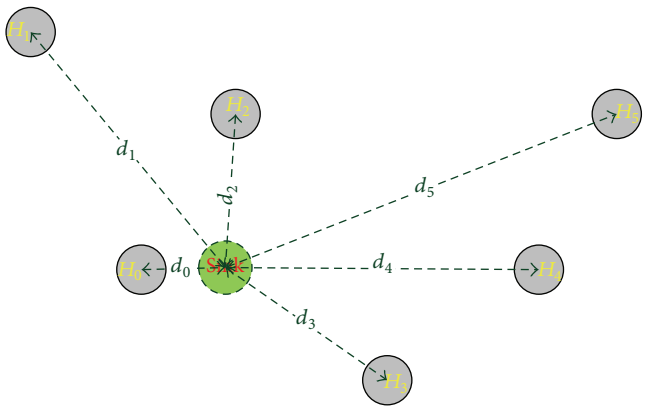



Figure 6: The *HOLE* distribution by Euclidean distance.

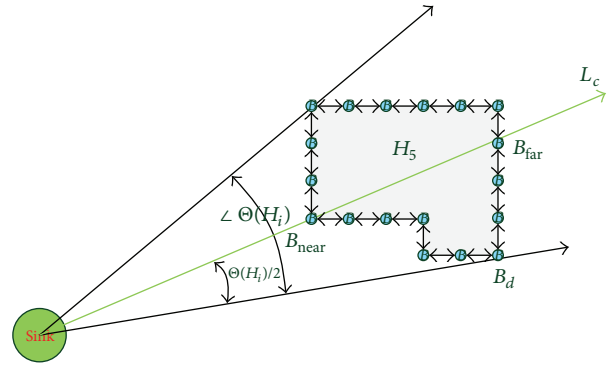



Figure 7: Illustration of the relationship between $B_{\text{near}}$, $B_{\text{far}}$, and $\Theta(H_i)$.

choose the optimum healing path and simplify the problem of irregular *HOLEs*, a bisector is used to divide the *HOLE* angle. The length of the line within the *HOLE* is considered the depth of the *HOLE*. Here, the definition of the *HOLE* depth $W(deep(H_i))$ is the Euclidean distance between $B_{\text{near}}$ and $B_{\text{far}}$. In order to find the two boundary nodes, the following definition is used.

*Definition 1.* A ray [Sink, $B_i$) divides $\Theta(H_i)$ equally and passes through $H_i$, which intersects the nearest border node on $H_i$, $B_{\text{near}}$ and the farthest border node on $H_i$, $B_{\text{far}}$. The method of finding the path from $B_{\text{near}}$ to $B_{\text{far}}$ is called the equally divided path selection method, abbreviated as EDPS.

To calculate both nodes $B_{\text{near}}$ and $B_{\text{far}}$, known conditions like the sink, $B_d$ and $\Theta(H_i)$ are used to establish a ray $\overrightarrow{SB_t}$, denoted as $L_c$, which is based on the $SB_d$ side and starts at the sink position with $\Theta(H_i)/2$. $L_c$ intersects with the *HOLE* at two boundary nodes called $B_{\text{near}}$ and $B_{\text{far}}$, shown in Figure 7. It is worth noting that the term "intersect" means that $L_c$ passes through the transmission range of a node. So the value of the boundary nodes $B_{\text{near}}$, $B_{\text{far}}$ and the metric of the *HOLE* depth $W(deep(H_i))$ can be described as follows:

$$B_{\text{near}} = \left\{B_j \mid d\left(L_c, B_j\right) \leqq R \;\&\&\; \min d\left(S, B_j\right), B_j \in H_i\right\},$$

$$B_{\text{far}} = \left\{B_j \mid d\left(L_c, B_j\right) \leqq R \;\&\&\; \max d\left(S, B_j\right), B_j \in H_i\right\},$$

$$W\left(deep\left(H_i\right)\right) = \sqrt{\left(B_{\text{far}} \cdot x - B_{\text{near}} \cdot x\right)^2 + \left(B_{\text{far}} \cdot y - B_{\text{near}} \cdot y\right)^2}. \quad (5)$$

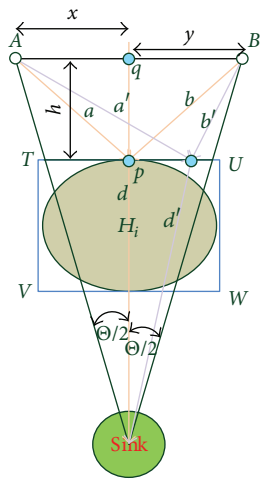



Figure 8: Illustration of the path selection by EDPS.

As discussed above, an established path from $B_{\text{near}}$ to $B_{\text{far}}$ is the shortest distance to the farthest node that crosses the *HOLE* and arrives at the sink. This paper will demonstrate that the decided path is the shortest distance for data transmission. Before Theorem 2, which calculates the shortest distance for node data transmission, is carried out, facing the sink first gives a rectangle parallelogram *TUWV* surrounding the *HOLE*, as shown in Figure 8.

**Theorem 2.** *The shortest healing path in $H_i$ can be determined by EDPS.*

*Proof.* In Figure 8, the parallelogram *TUWV* contains a *HOLE* $H_i$, and $p$ is an arbitrary point on segment *TU*, under uniform node distribution. *A* and *B* represent two nodes in symmetrical positions from the perspective of the sink. The proposed method attempts to find the shortest path that passes through the region of *TUWV*. It summarizes those distances from both points *A* and *B* to go through the same path and reach the sink. Points *A* and *B* are two symmetric points based on the stochastic random process.

Point $q$ is one point on segment *AB*. Let the length of *AB* be $L$, and let $\overline{Aq} = x$ and $\overline{Bq} = y$. As a result, $x + y = L$. In addition, let $h$ be the distance between $\overline{AB}$ and $p$, assuming that $\overline{Ap} = a$ and $\overline{Bp} = b$. According to the triangular theory, the following equations are obtained:

$$a^2 = h^2 + x^2, \tag{6}$$

$$b^2 = h^2 + y^2. \tag{7}$$

Here the function $f(x)$ is defined, and let $f(x) = a + b$.

Both (6) and (7) substitute for $f(x)$, so

$$f(x) = \sqrt{x^2 + h^2} + \sqrt{(L-x)^2 + h^2}. \tag{8}$$

Take (8) to derive by once and twice differential operations. This yields the first and second differential equations, respectively:

$$f'(x) = \frac{x}{\left(x^2 + h^2\right)^{1/2}} - \frac{L-x}{\left[(L-x)^2 + h^2\right]^{1/2}}, \tag{9}$$

$$\begin{aligned} f''(x) = {} & \frac{\left(x^2 + h^2\right)^{1/2} + x \times (1/2)\left(x^2 + h^2\right)^{-1/2}}{\left(x^2 + h^2\right)} \\ & - \left(\left[(L-x)^2 + h^2\right]^{1/2} \times (-1) + (L-x)\right. \\ & \left. \times \frac{1}{2}\left(x^2 + h^2\right)^{-1/2} \times (-2)(L-x)\right) \\ & \times \left[(L-x)^2 + h^2\right]^{-1}. \end{aligned} \tag{10}$$

According to the extreme value theorem, it is known that the function $f(x)$ has a minimum value when $x = L/2$. In other words, $a + b$ has a minimum value when the point $q$ is located at the center of $L$. In short, it is proved that the path found by EDPS from nodes *A* and *B* to point $p$ arriving at the sink is the shortest possible distance. □

*4.4. The Selection of the Optimal Healing Path.* The healing path will be selected according to the three factors described above, that is, the angle, the distance, and the depth of the *HOLE*. Here, it is worth noting to patch one node in the *HOLE*, if we pay attention to the position needed to be healed. Even though the coverage increase is fixed with a maximum size of the sensing range of a node, the delay time for the data transmission may vary with different healing positions. To extend the analysis of the proposed scheme to various applications, three parameters, $\alpha$, $\beta$, and $\gamma$ weighting on the three *HOLE* metrics, respectively, are used. The formula of the total healing metric is denoted as $W(H_i)$, shown as follows:

$$\begin{aligned} W(H_i) = {} & \alpha W(\Theta(H_i)) \times \beta W(len(H_i)) \\ & \times \gamma W(deep(H_i)), \end{aligned} \tag{11}$$

where $\alpha$, $\beta$, and $\gamma$ are positive harmonic coefficients. Let $\alpha + \beta + \gamma = 1$. Changing the parameters will induce different results. Based on (11), one *HOLE* will be selected. However, because the EDPS method selected healing path from $B_{\text{near}}$ to $B_{\text{far}}$ is a straight line, the patch nodes will be located in a virtual grid, as assumed earlier. As the straight line patch nodes may not be the real node positions, it is necessary to modify the straight line healing. To find the real path, the vertices in the *HOLE* which are close to $L_c$ are used. According to the virtual vertices computed in Algorithm 2, the vertices' accordance with the healing sequence should be arranged in the set of *Path*{}.

In Algorithm 2, $L_c$ is determined as in Figure 7. The virtual vertices set $V_i$ are the grid locations in $H_i$. The Steiner point problem layout method [25] is used with the proposed grid topology to determine the actual vertices' positions. As shown in Figure 9, the blue ray is calculated by the EDPS

**Input:** $L_c, B_{\text{near}}, B_{\text{far}}, V_i$
**Output:** $Path\{\}$
**Initial:** $Path\{\} = \varphi$
(1) $P \leftarrow B_{\text{near}}$
(2) Adding $B_{\text{near}}$ to $Path\{\}$
(3) $V_s \leftarrow P$
(4) $P = \{V_i \mid d(V_s, V_i) \le R_c \;\&\&\; \min\ d(L_c, V_i)\}$
(5) Add the vertex $P$ to $Path\{\}$
(6) $V_s \leftarrow P, \{V_i\} = \{V_i\} - P$
(7) Repeat step 4 until $P = B_{\text{far}}$
(8) Adding $B_{\text{far}}$ to $Path\{\}$

Algorithm 2: The algorithm for the modification from straight healing line to real positions.

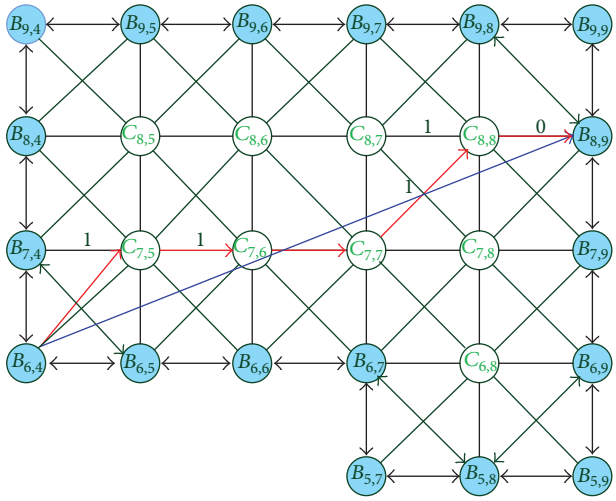



Figure 9: Illustration of the real healing path.

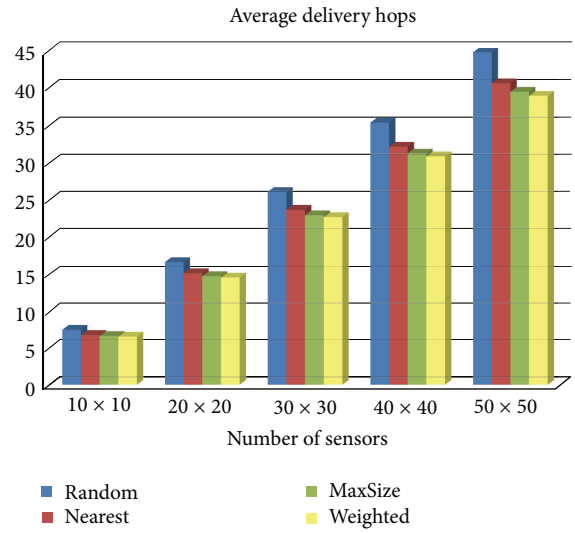



Figure 10: ADH (random $HOLE = 15$, $\alpha = \beta = \gamma = 1/3$).

method in $H_5$. The optimum healing path, which consists of the real healed positions, is denoted by the red arrows. In fact, these positions must also be sequentially recorded in $Path\{\}$. The elements of the set are sequenced locations of the vertices which will be healed. For example, Figure 9 has two nodes, $B_{6,4}$ and $B_{8,9}$, which are the border nodes $B_{\text{near}}$ and $B_{\text{far}}$, respectively. In addition, there are four vertices, $C_{7,5}$, $C_{7,6}$, $C_{7,7}$, and $C_{8,8}$. That is to say, the elements of $Path\{\}$ are $\{B_{6,4}, C_{7,5}, C_{7,6}, C_{7,7}, C_{8,8}, B_{8,9}\}$.

## 5. Simulation Results

The hole healing scheme proposed in this paper considers three weighted metrics, which are the angle, $W(\Theta(H_i))$, the depth, $W(deep(H_i))$, and the distance, $W(len(H_i))$, of the *HOLE*. The abbreviations $p_{ia}$, $p_{id}$, and $p_{il}$ are used to represent the normalized values between 0 and 1 of these three metrics. The delivery hop count was used to measure the data propagation delay. Therefore, if the number of hops can be reduced during data transmission, the data transmission will have a shorter delay time. This also has the advantages of reducing energy consumption and prolonging the lifetime of the WSN applications. The experiment in this paper was carried out by simulations under different environmental parameters to illustrate the variety of the average delivery path length for data transmission. The results show that the proposed weighted method has less delivery hops than do other methods, listed as follows:

(1) random selection for healing hole,

(2) nearest distance selection for healing hole,

(3) maximum size selection for healing hole.

First, in this simulation let the three weight values of $\alpha$, $\beta$, and $\gamma$ be equal; in other words, all of them are equal to 1/3. The average delivery hops (ADH) of the proposed weighted method were calculated and compared with those of the other methods. In the simulations, sensor nodes were initially deployed in the ruled fields of 10 × 10, 20 × 20, 30 × 30, 40 × 40, and 50 × 50, respectively. The sizes and locations of the *HOLEs* were randomly generated, and the maximum number of *HOLEs* is limited to 15. Each ADH result is the average of 1000 iterations for each case in the simulation, as shown in Table 1 and Figure 10.

Table 1: ADH (random *HOLE* = 15, $\alpha = \beta = \gamma = 1/3$).

| Node count | Method | | | |
|---|---|---|---|---|
| | Random | Nearest | MaxSize | Weighted |
| 10 × 10 | 7.34 | 6.64 | 6.52 | 6.41 |
| 20 × 20 | 16.44 | 14.96 | 14.55 | 14.37 |
| 30 × 30 | 25.89 | 23.48 | 22.75 | 22.49 |
| 40 × 40 | 35.2 | 31.97 | 31.05 | 30.66 |
| 50 × 50 | 44.66 | 40.5 | 39.35 | 38.81 |

$W(H_i)_{\max}$ denotes the largest *HOLE* score and is calculated by (12). $H_i$ denotes the $i$th *HOLE* with the maximum score, which will be selected for healing:

$$W(H_i)_{\max} = \underset{\forall i}{\operatorname{argmax}} (\alpha \cdot p_{ia} + \beta \cdot p_{id} + \gamma \cdot p_{il}), \quad \alpha + \beta + \gamma = 1. \tag{12}$$

Obviously, the longest average transmitted path was obtained by random selection; the second longest path was achieved by nearest distance selection; the third longest path was achieved by maximum size selection; and the best result (the shortest route) was obtained by the proposed weighted selection scheme. It also can be observed from the results that the number of ADH increased proportionally according to the number of nodes, and the result differences between the four methods were more conspicuous in larger areas.

To discuss the decrement rate ($D_r$) of ADH in the various methods, the $D_r$ of ADH is calculated by formula (13), and the results are shown in Table 2 and Figure 11. This roughly illustrates that the weighted method is reduced by 12.9% more than the random method, by 4% more than the nearest method, and by 1.3% more than the MaxSize method:

$$D_r = \frac{\text{result} - \text{result}_w}{\text{result}} \times 100\%. \tag{13}$$

The above simulation results were based on a fixed number of *HOLEs* and various numbers of deployed nodes. The next simulations were conducted with a fixed number of 30 × 30 nodes, and both the position and size of the *HOLEs* were produced by randomization. However, the numbers of *HOLEs* were given as 5, 10, 15, 20, or 25 for different cases. After carrying out 1000 simulations for each case, the ADH results were obtained as shown in Table 3 and Figure 12. The results show that the longest distance of the average transmission path was obtained by the random selection method; the second longest path was the nearest distance first selection; and the third longest path was the maximum size first selection method. However the best result (the shortest route) was obtained by the proposed weighted method. It also can be observed that when the number of *HOLEs* increases, the results of the different methods differ more significantly.

Similarly, a fixed number of 30 × 30 nodes was used, and both the position and size of the *HOLEs* were produced by randomization, and the number of *HOLEs* were given as 5, 10, 15, 20, or 25, respectively. By comparing the proposed weighted method with the random, nearest, and MaxSize methods, the results of the decrement rate of ADH were obtained, as shown in Table 4 and Figure 13, which roughly illustrate that the weighted method was reduced by about 9.75%~14.35% more than the random method, by about 1.94%~5.87% more than the nearest method, and by about 0.91%~1.21% more than the MaxSize method.

The simulation results shown above are compared with the different methods. However, Table 5 and Figure 14 show the varied ADH results based on the simulations of simultaneously changing both the number of nodes and *HOLEs* in the weighted method. It is clear that if the number of nodes

Table 2: Decrement rate of ADH (random *HOLE* = 15, $\alpha = \beta = \gamma = 1/3$).

| Node count | Method | | |
|---|---|---|---|
| | Random → Weighted | Nearest → Weighted | MaxSize → Weighted |
| 10 × 10 | 12.67% | 3.46% | 1.69% |
| 20 × 20 | 12.59% | 3.94% | 1.24% |
| 30 × 30 | 13.13% | 4.22% | 1.14% |
| 40 × 40 | 12.9% | 4.1% | 1.26% |
| 50 × 50 | 13.1% | 4.17% | 1.37% |

Table 3: ADH (node count = 30 × 30, $\alpha = \beta = \gamma = 1/3$).

| *HOLE* count | Method | | | |
|---|---|---|---|---|
| | Random | Nearest | MaxSize | Weighted |
| 5 | 22.97 | 21.14 | 20.92 | 20.73 |
| 10 | 24.83 | 22.77 | 22.3 | 21.98 |
| 15 | 25.89 | 23.48 | 22.75 | 22.49 |
| 20 | 26.37 | 23.92 | 23.09 | 22.77 |
| 25 | 26.76 | 24.35 | 23.2 | 22.92 |

Table 4: Decrement of ADH (node count = 30 × 30, $\alpha = \beta = \gamma = 1/3$).

| *HOLE* count | Method | | |
|---|---|---|---|
| | Random → Weighted | Nearest → Weighted | MaxSize → Weighted |
| 5 | 9.75% | 1.94% | 0.91% |
| 10 | 11.48% | 3.47% | 1.43% |
| 15 | 13.13% | 4.22% | 1.14% |
| 20 | 13.65% | 4.81% | 1.39% |
| 25 | 14.35% | 5.87% | 1.21% |

Table 5: ADH for proposed weighted *HOLE* selection method ($\alpha = \beta = \gamma = 1/3$).

| Node count | *HOLE* count | | | | |
|---|---|---|---|---|---|
| | 5 | 10 | 15 | 20 | 25 |
| 10 × 10 | 6.15 | 6.31 | 6.41 | 6.46 | 6.48 |
| 20 × 20 | 13.38 | 14.08 | 14.37 | 14.53 | 14.56 |
| 30 × 30 | 20.73 | 21.98 | 22.49 | 22.77 | 22.92 |
| 40 × 40 | 28.15 | 29.85 | 30.66 | 31.07 | 31.34 |
| 50 × 50 | 35.5 | 37.82 | 38.81 | 39.36 | 39.59 |

Table 6: ADH count (node count = 30 × 30, random *HOLE* = 15).

| $\alpha$ | $\beta$ | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0.1 | 0.2 | **0.3** | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
| 0 | 23.49 | 23.02 | 22.77 | 22.62 | 22.66 | 22.67 | 22.82 | 22.87 | 22.94 | 22.94 | 22.99 |
| 0.1 | 23.31 | 22.83 | 22.64 | 22.5 | 22.59 | 22.64 | 22.79 | 22.89 | 22.86 | 22.84 | |
| 0.2 | 23.05 | 22.71 | 22.54 | 22.54 | 22.58 | 22.63 | 22.71 | 22.79 | 22.88 | | |
| 0.3 | 22.87 | 22.67 | 22.51 | 22.51 | 22.55 | 22.59 | 22.69 | 22.79 | | | |
| 0.4 | 22.84 | 22.62 | 22.49 | 22.5 | 22.51 | 22.59 | 22.64 | | | | |
| 0.5 | 22.83 | 22.55 | 22.5 | 22.52 | 22.47 | 22.64 | | | | | |
| **0.6** | 22.67 | 22.54 | 22.5 | **22.38** | 22.53 | | | | | | |
| 0.7 | 22.75 | 22.57 | 22.5 | 22.53 | | | | | | | |
| 0.8 | 22.66 | 22.59 | 22.45 | | | | | | | | |
| 0.9 | 22.69 | 22.61 | | | | | | | | | |
| 1 | 22.7 | | | | | | | | | | |

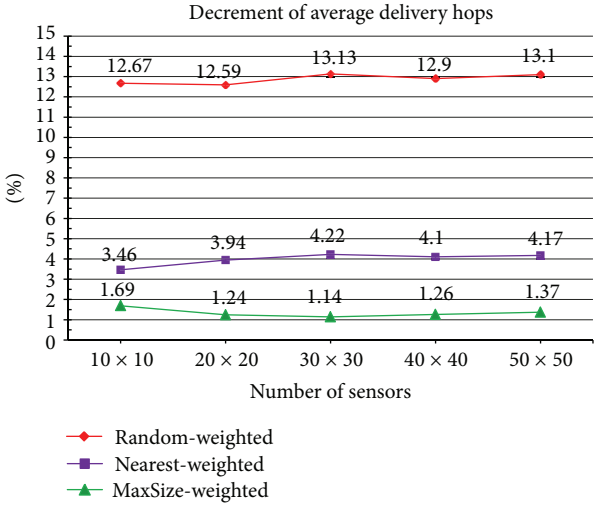



Figure 11: $D_r$ of ADH (random *HOLE* = 15, $\alpha = \beta = \gamma = 1/3$).

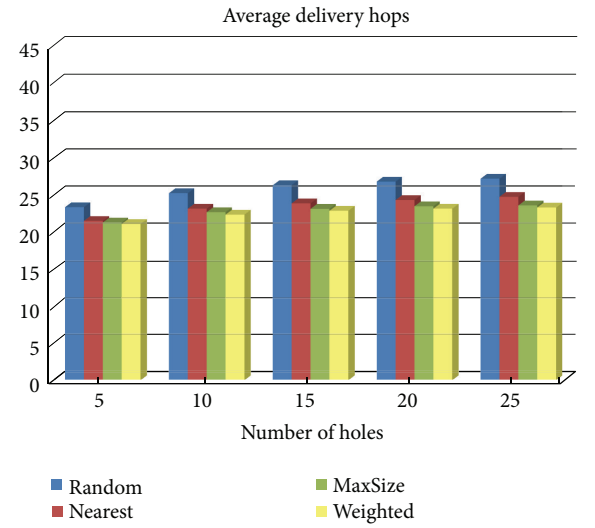



Figure 12: Average delivery hops (node count = 30 × 30, $\alpha = \beta = \gamma = 1/3$).

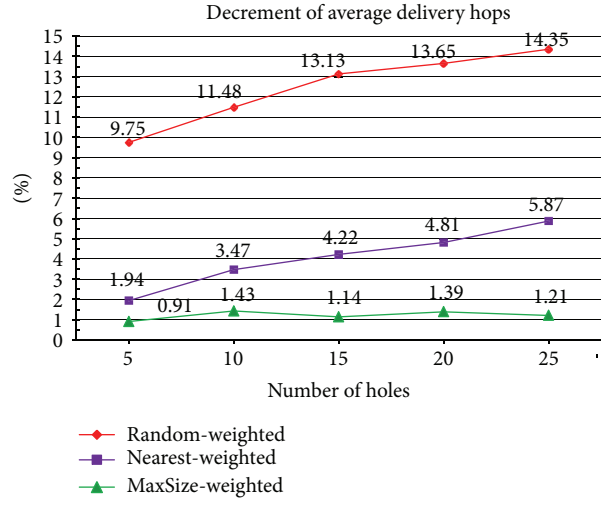



Figure 13: Decrement of ADH (node count = 30 × 30, $\alpha = \beta = \gamma = 1/3$).

or *HOLEs* increases, the ADH count will also increases. It is also clear that an increased number of nodes results in a more significant increase of ADH than does an increased number of *HOLEs*.

The results discussed above were under the condition of three equal weighting coefficients, $\alpha = 1/3$ for the *HOLE* angle, $\beta = 1/3$ for the *HOLE* depth, and $\gamma = 1/3$ for the *HOLE* distance parameters. Next, the various values of the coefficients $\alpha$, $\beta$, and $\gamma$ were considered, and the changes of the ADH in the weighted method were observed. Because the values of the weights $\alpha$, $\beta$, and $\gamma$ are between 0 and 1 and $\alpha + \beta + \gamma = 1$, in Table 6, the oblique line illustrates that the cases will never happen (the total value of weights exceeds 1). The value of $\gamma$ is related to $\alpha$ and $\beta$ and is calculated according to the following formula:

$$\gamma = 1 - \alpha - \beta. \tag{14}$$

Table 6 and Figure 15 illustrate the change of the ADH count with variously changed weighting values of $\alpha$, $\beta$, and

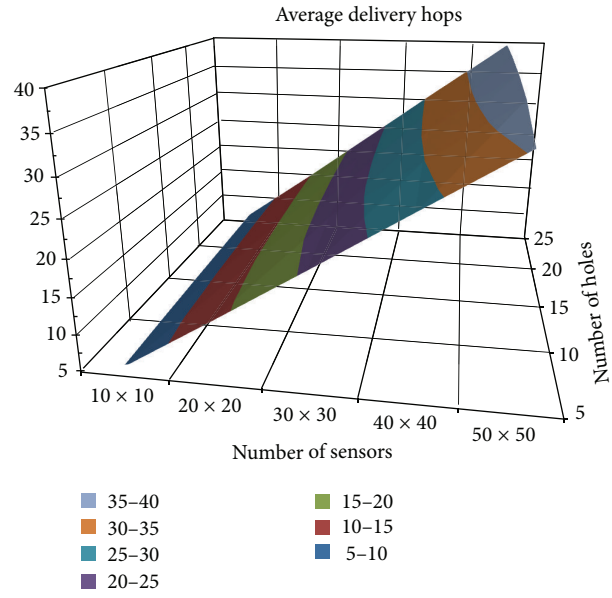



Figure 14: ADH for proposed weighted *HOLE* selection method ($\alpha = \beta = \gamma = 1/3$).

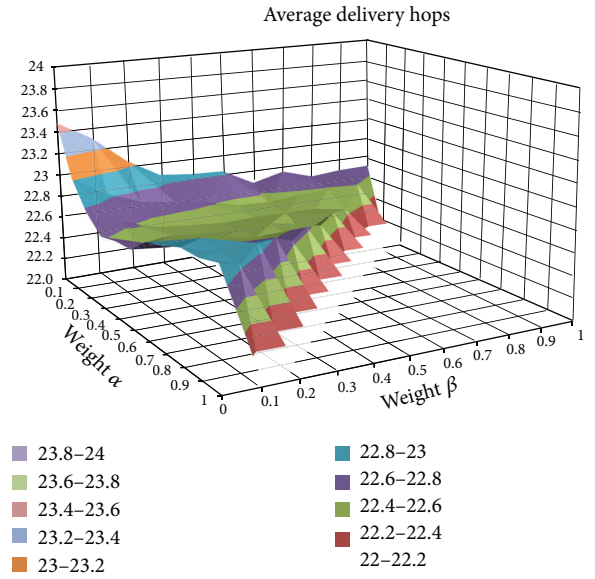



Figure 15: ADH (node count = 30 × 30, random *HOLE* = 15).

$\gamma$. The result is also the average of 1000 iterations in each simulation case. The numbers of nodes and *HOLEs* are 30×30 and 15, respectively. As the results show, when both weighting values $\alpha$ and $\beta$ are lower, the delivery hop count will become larger. When $\alpha = 0.6$, $\beta = 0.3$, and $\gamma = 0.1$, the lowest ADH count (= 22.38) is obtained, which means higher data transmission efficiency and lower energy consumption than the other permutations of the weighting coefficients.

According to the above simulation results, it clear that the proposed weighted method has demonstrates significantly better performance than the other methods. In addition, the simulations with adjusted of weighting values were able to obtain the optimum values for $\alpha$, $\beta$, and $\gamma$ in order to obtain the shortest ADH and apply the proposed weighted *HOLE* healing strategy to practical applications.

## 6. Conclusion

The focus of this paper is the rebuilding and maintenance of network topology connections. Reducing the time of postponement in the data transmission process is the purpose of the proposed hole healing scheme. Three properties of *HOLEs* for *HOLE* selection are proposed to improve the data conveyance time. They consider factors such as the angle of a *HOLE*, the distance between the sink and a *HOLE*, and the depth of the *HOLE*. In addition, the proposed EDPS algorithm is able to find a shorter path for healing the selected *HOLE*. This idea is quite useful for the healing process, especially when large *HOLEs* occur in the field. To ensure the accuracy of node location calculation, the field is imagined as a virtual ruled grid, and a robot is utilized to patch sensors in order to rebuild the topology connection. The performance of this proposed strategy is evaluated according to the average delivery hops. By comparing the proposed method with the random selection, nearest selection, and maximum size selection methods, the simulation results show that the proposed strategy outperforms the others. We believe that our proposed method of hole healing with data delivery awareness (HHDDA) is a valuable maintenance method for WSN applications.

## Acknowledgment

The authors would like to thank the Research Center for Energy Technology and Strategy (RCETS) of National Cheng Kung University for financially supporting this research under Grants no. D102-23015 and no. D102-15103.

## References

[1] G. J. Pottie, "Wireless Sensor Networks," in *Proceedings of the IEEE Information Theory Workshop*, pp. 139–140, Killarney, Ireland, 1998.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[3] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," in *Proceedings of the Annual ACM International Conference on Mobile Computing and Networking MobiCom*, pp. 263–270, 1999.

[4] T.-W. Sung and C.-S. Yang, "A cell-based sensor deployment strategy with improved coverage for mobility-assisted hybrid wireless sensor networks," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 5, no. 3, pp. 189–198, 2010.

[5] T.-W. Sung and C.-S. Yang, "An adaptive joining mechanism for improving the connection ratio of ZigBee wireless sensor networks," *International Journal of Communication Systems*, vol. 23, no. 2, pp. 231–251, 2010.

[6] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proceedings of the 20th Annual Joint Conference of the*

*IEEE Computer and Communications Societies*, pp. 1380–1387, April 2001.

[7] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 4, pp. 345–406, 1991.

[8] A. Ghosh, "Estimating coverage holes and enhancing coverage in mixed sensor networks," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 68–76, November 2004.

[9] G. Wang, G. Cao, P. Berman, and T. F. La Porta, "Bidding protocols for deploying mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 515–528, 2007.

[10] T.-W. Sung and C.-S. Yang, "Voronoi-based coverage improvement approach for wireless directional sensor networks," *Journal of Network and Computer Applications*, 2013.

[11] T.-W. Sung and C.-S. Yang, "Distributed voronoi-based self-redeployment for coverage enhancement in a mobile directional sensor network," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 165498, 15 pages, 2013.

[12] S. Babaie and S. S. Pirahesh, "Hole detection for increasing coverage in wireless sensor network using triangular structure," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 1, article 2, 2012.

[13] C. -F. Huang and Y. C. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of the 2nd ACM international conference on Wireless Networks and Applications (WSNA '03)*, 2003.

[14] X. Li, D. K. Hunter, and K. Yang, "Distributed coordinate-free hole detection and recovery," in *Proceedings of the IEEE Global Telecommunications Conference (Globecom '06)*, December 2006.

[15] N. P. Le, B. T. Quan, N. T. Hieu, and N. K. Van, "Efficient approximation of routing holes in wireless sensor networks," in *Proceedings of the 2nd Symposium on Information and Communication Technology (SoICT '11)*, pp. 72–79, October 2011.

[16] S. Zhou, M.-Y. Wu, and W. Shu, "Finding optimal placements for mobile sensors: wireless sensor network topology adjustment," in *Proceedings of the IEEE 6th Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication*, pp. 529–532, June 2004.

[17] P. Si, C. Wu, Y. Zhang, Z. Xia, and L. Cheng, "A hole detection and redeployment strategy in wireless sensor network," *Journal of Information and Computational Science*, vol. 8, no. 13, pp. 2577–2585, 2011.

[18] X.-H. Deng, C.-G. Xu, F.-Y. Zhao, and Y. Liu, "Repair policies of coverage holes based dynamic node activation in Wireless Sensor Networks," in *Proceedings of the 8th International Conference on Embedded and Ubiquitous Computing (EUC '10)*, pp. 368–371, December 2010.

[19] G. Wang, G. Cao, P. Berman, and T. F. La Porta, "Bidding protocols for deploying mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 515–528, 2007.

[20] Fu-Tian Lin, Liang-Cheng Shiu, Chao-Yang Lee, and Chu-Sing Yang, "A method to analyze the effectiveness of the holes healing scheme in wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 725452, 10 pages, 2013.

[21] B. Hao, J. Tang, and G. Xue, "Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation," in *Proceedings of the Workshop on High Perfomance Switching and Routing (HPSR '04)*, pp. 246–250, April 2004.

[22] S. Funke, "Topological hole detection in wireless sensor networks and its applications," in *Proceedings of the joint workshop on Foundations of mobile computing (DIALMPOMC '05)*, September 2005.

[23] C. Zhang, Y. Zhang, and Y. Fang, "A coverage inference protocol for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 850–864, 2010.

[24] Y. Bejerano, "Coverage verification without location information," *IEEE Transactions on Mobile Computing*, no. 4, pp. 11631–11643, 2011.

[25] G.-H. Lin and G. Xue, "Steiner tree problem with minimum number of Steiner points and bounded edge-length," *Information Processing Letters*, vol. 69, no. 2, pp. 53–57, 1999.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 125145, 20 pages
http://dx.doi.org/10.1155/2013/125145

*Research Article*

# Distributed Continuous *k* Nearest Neighbors Search over Moving Objects on Wireless Sensor Networks

**Chuan-Ming Liu and Chuan-Chi Lai**

*Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 10608, Taiwan*

Correspondence should be addressed to Chuan-Ming Liu; cmliu@csie.ntut.edu.tw

Received 29 June 2013; Accepted 20 October 2013

Academic Editor: Chang Wu Yu



Continuous *k*-nearest neighbor (C*k*NN) search is a variation of *k*NN search that the system persistently reports *k* nearest moving objects to a user. For example, system continuously returns 3 nearest moving sensors to the user. Many query processing approaches for C*k*NN search have been proposed in traditional environments. However, the existing client-server approaches for C*k*NN search are sensitive to the number of moving objects. When the moving objects quickly move, the processing load on the server will be heavy due to the overwhelming data. In this thesis, we propose a distributed C*k*NN search algorithm (DC*k*NN) on wireless sensor networks (WSNs) based on the Voronoi diagram. There are four features about DC*k*NN: (1) each moving object constructs a local Voronoi cell and keeps the local information; (2) in order to keep the reliability of system, the query message will be propagated to related objects; (3) using the idea of safe time, the number of updates is reduced; (4) an equation to estimate a more accurate safe time is provided. Last, we present our findings through intensive experiments and the results validate our proposed approach, DC*k*NN.

## 1. Introduction

The mobile sensors have been widely used in recent years. For example, the smartphone, Samsung Galaxy S4, includes a barometer, thermometer, and hygrometer (to measure humidity)—the first major smartphone to do so. By using these sensors, people can easily use their mobile devices to obtain environmental information. On the other hand, the popularization of global positioning system (GPS) and the miniaturization of mobile devices, equipped with a wireless network module, make location-based services (LBSs) no longer expensive. Thus, with the combination of the mobile sensors and the GPS, more and more interesting applications and issues gradually emerged.

In general, one can regard each mobile device (sensor) as a moving object in wireless mobile sensor networks. Each user (object) can use the GPS and sensors to provide the environmental information, such as location, temperature, and humidity, to the information systems which are built by the manufacturers. These systems can sink user's information and send the user's location through the mobile networks. Mobile users can immediately access the information related to the location and geographic information. Many manufacturers even provide different telecommunication services to the users, The multiple types of information services make life more convenient. For example, if a user issues a query and wants to know the temperature at a specific location, the neighboring mobile objects of this location will return the measured temperature information to the user. Another example is the temperature monitoring system where a mobile sink may move around to collect the temperature around it. In these cases, the query may move.

As the above examples, the mobile query often selects *k* nearest neighbors (*k*NN) [1] around itself to obtain the average of the measured temperatures. Moreover, the mobile query may monitor the environmental information for a period of time. Each moving object thus needs to continuously reply to the mobile query with measured environmental information and location. Such an application actually is the *continuous k nearest neighbors* (C*k*NN) query [2, 3], which is used to derive the *k* nearest neighbor sensors and then get the average of the received information continuously. In this

paper, we consider the mobile environment where the data objects and queries can move.

Spatial queries, such as $k$ nearest neighbors query and continuous $k$ nearest neighbors query, have attracted many researchers' interest and also have been discussed on many different applications. Most of the existing methods about the $k$NN or C$k$NN search in the considered mobile environment still use the client-server architecture and rely on a central server for processing queries. Each moving object needs to continuously obtain the location information of itself by GPS and send the information back to the server for updating the database. Thus, if we use the existing methods under a specific scenario that the position status of objects changes frequently, the server must receive a huge amount of information stream for updating each object's location in a short time, thus causing the server system to be overloaded. Besides, the large amount of messages for updates occupies quite a lot of bandwidth. We hence propose an effective and noncentralized approach for supporting C$k$NN query in a mobile environment.

In this work, we consider the continuous $k$ nearest neighbors query in WSNs or some other distributed and mobile environments like WSNs, which consist of a number of mobile devices (sensors) and propose a new distributed approach, DC$k$NN, for processing C$k$NN query efficiently. From the object's point of view, the distributed architecture using local computation can effectively reduce the updating cost since the update messages in the client-server architecture may need more hops to arrive to the server. Even worse, the long delay of the update messages can make the answer of the C$k$NN inaccurate. On the other hand, the distributed architecture can avoid the long response time caused by the overload of the server. In our considered environments, each mobile device, also called moving sensor or *moving object*, can obtain its location information by the GPS equipment. Besides, each moving object is equipped with enough storage and a cpu for storing the local information and processing simple 2D functions, respectively.

In our proposed approach, each moving object will send messages to its one-hop neighbors when necessary to avoid the wrong information for processing spatial queries. When finding the $k$NN, our method uses the incremental strategy on the number of hops based on the Voronoi diagram (VD) [4]. In order to estimate the movement of a moving object accurately, we consider the speed of each moving object and adapt the concept of *safe region* [5] to our proposed approach, distributed continuous $k$ nearest neighbors (DC$k$NN).

In this paper, we address the problem of efficiently processing continuous $k$ nearest neighbors over moving objects with updates and make the following contributions.

(i) We identify the continuous $k$ nearest neighbors search problem on mobile ad hoc networks and categorize some detailed existing works about this issue.

(ii) We provide a novel approach, using local Voronoi diagram, for efficiently and precisely collecting the location of each object in a distributed way.

(iii) With the combination of local Voronoi diagram, we propose a model for accurately estimating the safe-time (safe-region) of continuous $k$ nearest neighbors query, which can lead to a short response time, fewer number of messages, less update cost, and high accuracy.

(iv) We perform a simulated experimental evaluation and the results show the superiority of our solution over the adaptation of state-of-the-art solutions [5] given to the similar issues.

The rest of this paper is organized as follows. In Section 2, we review the related work. The performance metrics and considered issues are introduced in Section 3. Section 4 presents the proposed solution using localized Voronoi diagram and how to derive the accurate safe time. Section 5 gives some analysis and comparisons between the proposed DC$k$NN method and different existing methods. The discussion on the experimental simulation is in Section 6. Finally, we give the conclusion remarks in Section 7.

## 2. Related Work

The existing works for C$k$NN search can be categorized into the following categories according to the ways for accessing or managing data: (1) *pull-based* approach, (2) *push-based* approach, and (3) distributed and mobile approach. The pull-based approach uses the traditional client-server architecture [6–8]. A given central server is responsible for processing the spatial queries from the user devices and all the data are stored in the server. Whenever a client wants some information, the client will send a query to the server. In many applications, the amount of spatial data needed to be processed could be very large and hence the spatial data is usually saved in the external storages, such as disks. Thus, the existing approaches for C$k$NN search using the pull-based approach focus on optimizing the number of I/O's. Of course, the computation time (CPU time) is still an important performance metric when the size of spatial data needed to be processed is not large. As a result, the above two measurements, the number of I/O's and CPU time, are usually used to validate the effectiveness of the proposed algorithms for processing $k$NN and C$k$NN queries using pull-based approach. However, the bandwidth of communication between the clients and servers is *asymmetric*. In other words, the uplink bandwidth is limited and much smaller than the bandwidth of downlink. This phenomenon in wireless communication will cause the bottleneck [9] when the number of queries increases if the pull-based approach is applied for accessing data.

In order to overcome the bottleneck problem caused by the pull-based approach in wireless environment, the push-based approach is proposed and also referred to as *data broadcasting* approach. Data broadcasting has attracted a lot of research attention in the past decades and many approaches or protocols for different types of spatial queries have been proposed [10–13]. In the data broadcasting environment, the data are broadcast in the air by the server and the clients execute the query process by listening to

the broadcast channel. The load for processing data on the server is dramatically decreased and relieves the bottleneck problem. Instead, the computation load on the clients is increased. From the user's point of view, the fewer battery power each device consumes, the better experience each user has. Thus the existing works [10, 11, 13–15], in general, discuss the performance of data broadcasting protocol in terms of *latency* and *tuning time*. Latency is the time a client experiences from issuing query to receiving the complete answer. The tuning time is the amount of time actually spent on listening to the broadcast. The latency indicates the quality of service (QoS) provided by the system and the tuning time can represent the power consumption of mobile clients.

Many works [16–18] have discussed the $k$NN and C$k$NN queries on the distributed and mobile environments like mobile ad hoc NETworks (MANETs) and wireless sensors networks (WSN). MANETs and WSNs are self-configuring infrastructure-less networks consisting of mobile devices and sensors connected by wireless communication. Each device in such networks is free to move independently in any direction and will thus change its links to other devices frequently. Each device can forward the traffic unrelated to its own use and act as a router. In contrast to the client-server and data broadcasting environments, there is no central server that handles the spatial queries or broadcasts spatial data in MANETs or WSNs. Accordingly, the information system based on the above infrastructure-less networks has to process the queries in a distributed way. Each mobile node (or device) in such a system cooperates and exchanges spatial data with each other and then derives the answer for the spatial queries.

In spatiotemporal data applications, the datasets consist of data objects and the data as well as queries might move over time. In order to process a great volume of spatial data and queries efficiently, some decomposition methods, such as grid and Voronoi diagram, have been used in the existing works [7, 19, 20]. Xiong et al. [20] focused on multiple $k$NN queries and proposed an incremental search technique based on hashing objects into a regular grid and keeping CPU time in mind. The main objective of this work on disk-resident data is to minimize disk I/O operations. The CPU time is considered only as a secondary objective. Zhao et al. [21] proposed a Voronoi-based approach for C$k$NN query. Although this work provided a navigation service to mobile devices, the query processing is still operated on a central server.

Mouratidis et al. [22] proposed a threshold-based algorithm to help the server processing continuous nearest neighbor queries from geographically distributed clients. Within a given threshold, the $k$ nearest neighbors of the query point will not change. However, this work still needs a central server for monitoring the $k$NN objects, maintaining the huge spatial data, and broadcasting each object's information. Chatzimilioudis et al. [16] proposed a proximity algorithm to answer all $k$-nearest neighbor queries continuously. In such an environment, many stations, called query processors, are placed to monitor all the objects in the radio region. Then, different query processors can exchange the monitored spatial information to calculate the $k$ nearest neighbor answer to the user (query point).

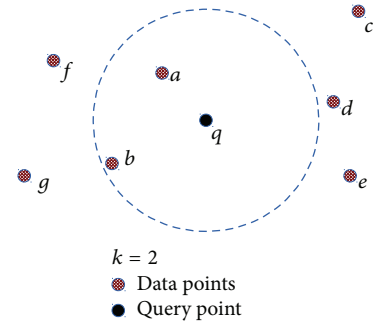



Figure 1: A 2NN search with query point $q$, where 2NN($q$) = $\{a, b\}$.

In contrast to most of the above approaches, Kim et al. [5] proposed a decentralized method to support continuous $k$ nearest neighbor query. Each query object predicts a safe time as a threshold for reducing the cost of information update. This method assumes that there is a maximum speed, *MaxSpeed*, of each object in the environment and use, DI$k$NN algorithm [23] to find the initial $k$ nearest neighbors. By monitoring the $k$th and $(k + 1)$th NN, the mobile query can know the influence region and use this region to filter moving objects, thus reducing a great amount of update cost. However, this work only considered the stationary query, not a moving query. We will discuss the accuracy of this method later in details. Before introducing the proposed algorithm, we first give some background and terminologies.

## 3. Preliminaries

In this section, we present some concepts and techniques that are generally used when solving the spatial data queries on moving objects. The metrics generally used to evaluate the query process will also be introduced.

*3.1. Spatial Query.* The $k$ nearest neighbor search ($k$NN search) is one of the important types of spatial queries. Suppose that the distance between two points $v$ and $u$ is dist$(v, u)$ and assume that there is a given dataset $D$ with a query point $q$. The $k$ nearest neighbor search is to find the $k$ nearest data objects in dataset $D$. A simple example in Figure 1 illustrates the 2NN of the given query point $q$ and data objects $a$ and $b$ are the two data objects closest to $q$. In this case, we will say 2NN($q$) = $\{a, b\}$. The definition of the $k$ nearest neighbor search problem thus can be defined as follow.

*Definition 1* ($k$NN search problem). Given a query point $q$ and an integer $k$, the $k$ NN search problem is to find a subset $k$NN($q$) $\subseteq D$ with $|k\text{NN}(q)| = k$, such that any data object $p \in k\text{NN}(q)$ and dist$(p, q) \leq \min_o \text{dist}(q, o)$, where $o \in (D\text{-}k\text{NN}(q))$.

The continuous $k$ nearest neighbors search (C$k$NN Search) [3] is a variation of $k$NN query. C$k$NN search is extended from $k$NN search and the system will continuously return the $k$NN answer in a time interval. In a mobile environment, the mobile devices are regarded as moving objects. Thus, the result of a C$k$NN search may change over time due to the movement of the queries and data objects. We can

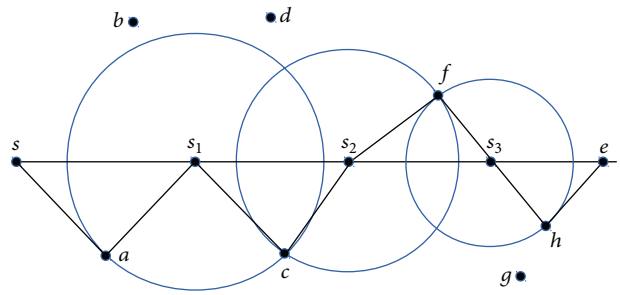



Figure 2: An example of C$k$NN, $k = 1$, where the output is $\{\langle\{a\}, [s, s_1]\rangle, \langle\{c\}, [s_1, s_2]\rangle, \langle\{f\}, [s_2, s_3]\rangle, \langle\{h\}, [s_3, e]\rangle\}$ for $q = [s, e]$.

model the moving pattern of the query as a path consisting of consecutive line segments and then consider the line segments for C$k$NN search. Let $D$ be a dataset of data objects and $q = [s, e]$ be a query line segment. A continuous $k$ nearest neighbors (C$k$NN) query retrieves $k$ nearest neighbors of every point in the line segment $q = [s, e]$. The result contains a set of $\langle R, T\rangle$ tuples, where $R$ is the resulting $k$NN within the time interval $T$ on $[s, e]$. Consider Figure 2 as an example, where $D = \{a, b, c, d, e, f, g, h\}$ and $k = 1$. The output of the query is the set of tuples $\langle\{a\}, [s, s_1]\rangle$, $\langle\{c\}, [s_1, s_2]\rangle$, $\langle\{f\}, [s_2, s_3]\rangle$, and $\langle\{h\}, [s_3, e]\rangle$. Tuple $\langle\{a\}, [s, s_1]\rangle$ indicates that $\{a\}$ is the $k$NN($q$) ($k = 1$) for interval $[s, s_1]$ and so on. We then can define the continuous $k$ nearest neighbor search problem as below.

*Definition 2* (C$k$NN search problem). Given a line segment $[s, e]$ and an integer $k$, for any data object $p$, if $o_k \in$ C$k$NN($[s, e]$) is one of the $k$ nearest neighbor, then for any object $o' \in (D\text{-}Ck\text{NN}([s, e]))$, $\text{dist}(p, o') \geq \text{dist}(p, o_k)$.

*3.2. Voronoi Diagram.* Using Voronoi diagram (VD) for C$k$NN search is proposed in the pull-based approach [7]. Consider a set $P$ of $n$ distinct data points, called *sites*, in the plane. The Voronoi diagram of $P$, denoted by VD($P$), is defined as the subdivision of the plane into $n$ cells, one for each site in $P$, and each cell corresponding to a site $p \in P$ is defined to be the area of all points in the plane closer to $p$ than to any other point in $P$. Figure 3 shows the Voronoi diagram of 16 data points $p_1, p_2, \ldots, p_{16}$ in the plane. Actually, the boundary between two points is the perpendicular bisector of those two points. The method proposed in [7] uses the characteristics of the perpendicular bisector (the distances from any point on this line to both neighboring points are the same) to determine the nearest neighbors of the current query point. Let us use Figure 3 as an example. Then, each solid line is the perpendicular bisector between two neighboring data objects. Suppose that the solid line with arrows denotes a path for the query point from position $S$ to position $D_1$ and the dotted line with arrows is the path from $S$ to $D_2$. For the path from $S$ to $D_1$, it is very easy to determine the answer set of the continuous nearest neighbors using Voronoi diagram and $\{p_9, p_8, p_{11}, p_{12}, p_{10}\}$ is the result. On the other hand, for the dotted path from $S$ to $D_2$, since the query does not move to another Voronoi cell, the answer set of continuous nearest neighbor is always $\{p_9\}$.

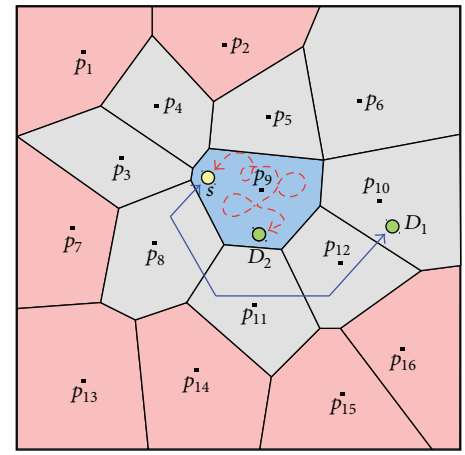



Figure 3: A continuous nearest neighbor search example using Voronoi diagram.

Using the Voronoi diagram, one can easily find the NN for a query point. To find the $k$NN for an arbitrary $k > 1$, we can extend the concept of Voronoi diagram to have the order-$k$ Voronoi diagram [24]. Using the order-$k$ Voronoi diagram, one can find the $k$NN for a query point by locating the query point in some order-$k$ Voronoi cell. Patroumpas et al. proposed a method [25], which approximates the order-$k$ Voronoi cells by limited data objects to support continuous $k$ nearest neighbor query. Figure 4 shows the process of constructing an approximate order-3 Voronoi cell. Before the construction, all the $k$ nearest neighbors are known. Initially, the system assumes that approximate cell VC$'$ containing $q$ is equal to the entire universe of interest where any possible location is being recorded. The system regards query point $q$ as the center point, divides the whole region into eight quadrants, and finds the 3NN($q$) = $\{n_1, n_2, n_3\}$, as shown in Figure 4(a). For each closest point $m_i$ to $q$ in each quadrant, the system delineates all its $k$ bisectors with the $k$NN points and then produces the approximate order-$k$ Voronoi cell of the $k$NN point with respect to $q$. But, for reducing processing cost even further, the system chooses bisector $b_i$ that is closest to $q$ and thus bounds the cell most tightly (Figure 4(b)). Accordingly, by taking the most restrictive bisector $b_i$ for each quadrant in counter-clockwise order, the system can gradually crop entire universe and obtain the approximate order-$k$ Voronoi cell VC$'$ as shown from Figures 4(b) to 4(i). The gray region in Figure 4(i) is the resulting approximate order-3 Voronoi cell. No matter the query moves in this region, 3NN($q$) = $\{n_1, n_2, n_3\}$.

*3.3. Safe-Time.* In the wireless sensor network, the idea of *safe-time* is proposed by Kim et al [5]. This method first assumes that there is a maximum speed, *MaxSpeed*, of each object in the environment and then uses DI$k$NN algorithm [23] to find the initial $k$ nearest neighbors. As Figure 5 shows, the user located at $S$ issues a query with $q$ as the center point of this query. The system will first inquiry object $n_p$, the nearest neighbor of the query point $q$. Then, the process searches the initial $k$NN counter-clockwise on the concentric

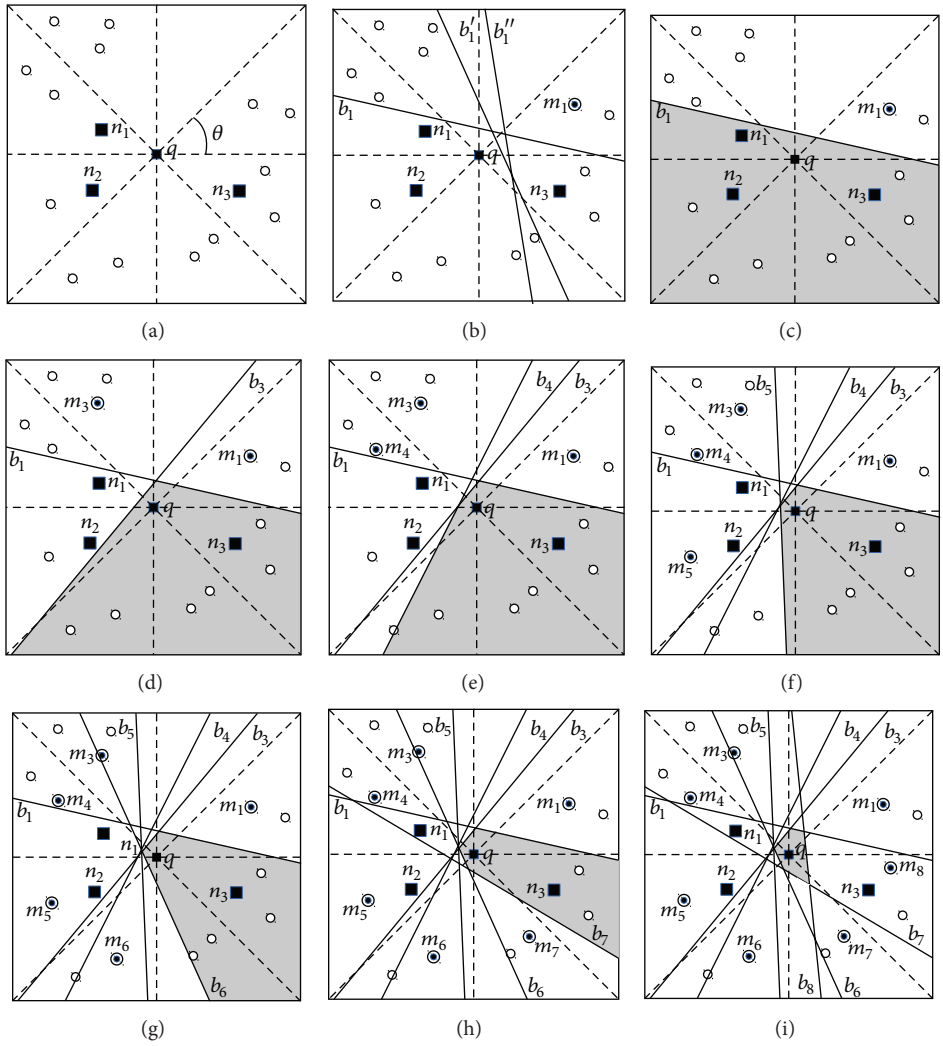



FIGURE 4: The construction of an approximate order-3 Voronoi cell, where $q$ is the query point.

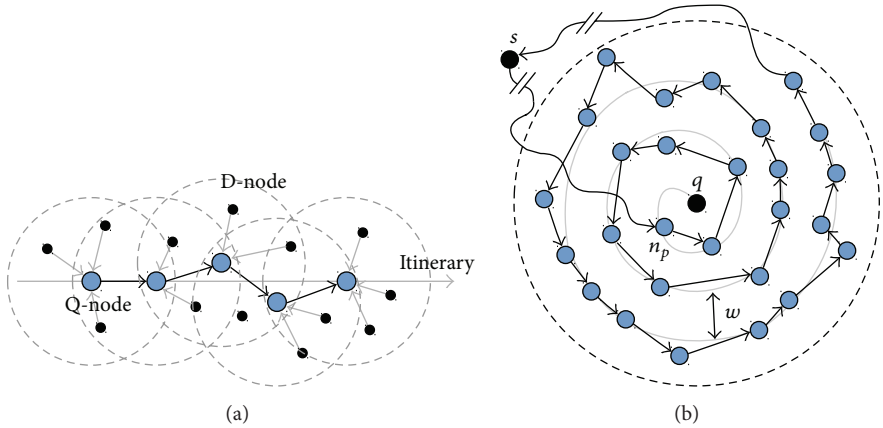



FIGURE 5: The $k$NN search on WSNs: (a) Q-node is the first object receiving the query and D-node is the sink node which is responsible for collecting data and (b) presents the multihops routing path for the $k$NN process.

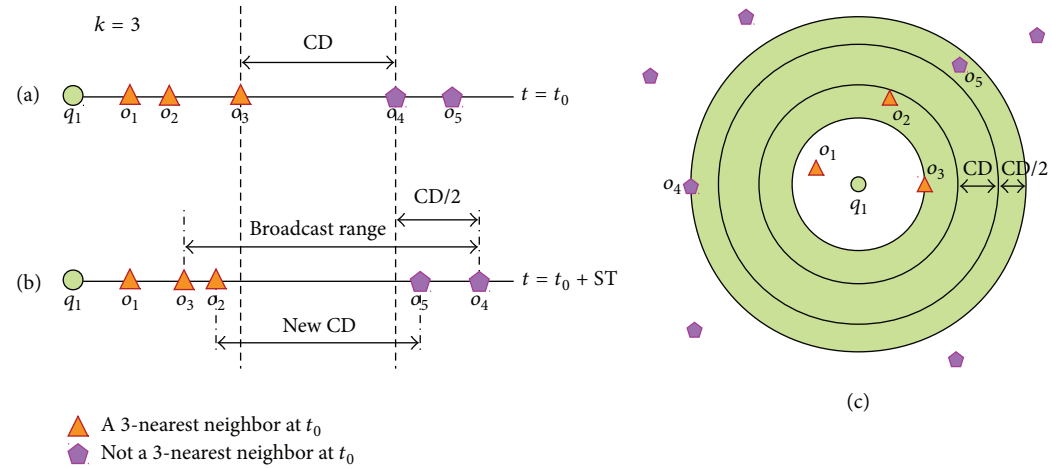



FIGURE 6: Update process using safe-time for 3NN search on WSNs: (a) shows the CD between the third and fourth closest data points; (b) presents the maximum influenced distance for update using broadcast; (c) indicates the area for broadcasting to update.

circles incremental with an distance $w$. The query message is forwarded in hop by hop manner as shown in Figure 5(b). Finally, the system can obtain the initial $k$NN from the last object which finished the query processing and derive the distance, *cross-distance* (CD), between the $k$th and $(k + 1)$th nearest neighbors. As Figure 6(a) shows, the system then can compute a safe time = CD/(2 ∗ MaxSpeed) by the cross-distance between the $k$th and $(k + 1)$th nearest neighbors. This indicates that the system ensures that the order of $k$th and $(k + 1)$th nearest neighbors will not exchange in the safe-time period. Thus, the resulting $k$NN does not need to update. In addition, the maximum moving distance of each object is limited in the safe-time, so we can predict the maximum influenced region after the movement of $k$th and $(k + 1)$th nearest neighbors as shown in Figures 6(b) and 6(c) to obtain the safe region (the dark rings in the figure). The system will ignore the objects which locate outside the safe region, thus reducing the cost of updating each object's information.

*3.4. Performance Metrics.* In order to evaluate the proposed DC$k$NN approach, we consider the following performance metrics, including *update cost*, *number of messages*, *accuracy*, and *response time*.

*3.4.1. Update Cost.* The update cost is to measure the number of updates during the query process. We sometimes use the frequency to denote it. Since the sensors (objects) and even the query will move, the status of the sensors should be maintained in order to have the accurate results. However, the cost for one update is high in the distributed environment. Hence, one of the objectives of the designed algorithm, DC$k$NN, is to minimize the update cost for the C$k$NN query.

*3.4.2. Number of Messages.* Except for the number of updates, there are some other kinds of messages for the query process in the distributed and mobile environment, including the messages for query and information passing. The total number of messages used during the query process can be used to indicate the throughput over the networks and the energy consumption. So the proposed DC$k$NN is towards minimizing the total number of messages for the query.

*3.4.3. Accuracy.* By using the pull-based approach, the time interval for routing update messages to the server depends on the distance between the data object and the server. It thus can not ensure that the location information of each object saved in the server is the latest. Hence, the results may be timely incorrect when deriving the answers for a query due to the mobility. This situation becomes worse in the distributed and mobile environments. In our design, the proposed approach can avoid redundant updates and keep the accuracy of the results.

*3.4.4. Response Time.* The complicated query process, the delay time for message communication, and the number of users will affect the response time of the system to process the query. Long response time is unbearable and may return obsoletely invalid query results. Even worse, the long response time may cause important information missing or financial loss. Thus the system response time is also one of the important issues that must be considered when designing the method for continuous queries.

## 4. Distributed Continuous $k$ Nearest Neighbors Search

In the mobile and distributed environments, like WSNs or MANETS, the pull-based approach may not be a good match for C$k$NN search due to the problems discussed in Section 2. We thus propose a distributed algorithm, DC$k$NN, for C$k$NN search in the mobile and distributed environments. The distributed approach using a better safe-time can reduce the update cost, relieve the load for server, shorten the response time, and make the result more accurate. There are three phases in DC$k$NN: initialization phase, query processing phase, and information update phase. In the initialization

```
query_DCkNN_search()
 (1) broadcast a query q(k, hopc);        // hopc is initial to 1
 (2) get the Voronoi cells or object's information;
 (3) if (k = 1) then
 (4)       show LVC;
 (5)       find safetime of DC1NN (with LVC);
 (6) elseif (k > 1) then
 (7)       while the received information is insufficient do
 (8)             broadcast the query q(k, hopc + 1) to ask more hops of neighbor nodes;
 (9)       run normal kNNSearch to find kNN set;
(10)       show LVC according to kNNset and Info;
(11)       find safetime of DCkNN (with LVC)
(12) while (safetime ≠ 0)
(13)       wait;
```

Algorithm 1: The DC$k$NN algorithm on the query (sink) node.

phase, each moving object exchanges the location information and builds a local Voronoi cell (LVC) as well as calculates the safe time for the LVC. The query processing phase collects sufficient location and environmental information of neighboring neighbors for exploring the result of the given C$k$NN query. The last information update phase is to maintain the latest information for the moving objects and the result of the continuous $k$ nearest neighbor query. The following will explain each phase in details.

*4.1. Phase 1: Initialization* ($k$=1). During system initialization, each mobile sensor (moving object) exchanges location information with its neighbors, derives the local Voronoi cell (LVC), calculates the safe time of LVC, and then stores the LVC and safe time in the memory.

When a user issues an NN query $Q$ and broadcasts the query message $q(k, hopc)$ (case $k = 1$), where $k$ is the number of the nearest neighbors the query wants to find and $hopc$ is the hop counter and sets to one initially. When the query point $Q$'s neighboring objects receive the query message $q(k, hopc)$, they will determine whether $Q$ is located in their Voronoi Cells (VCs). If a neighboring object replies true, it means that this neighboring object is the nearest neighbor of the query point $Q$ and this neighboring object will transmit the Voronoi cell (VC) information of itself to the query point $Q$, including the neighboring object boundaries, location, direction, and speed of movement, and some environmental information. On the other hand, if a neighboring object returns false, this neighboring object will check whether query point $Q$ moves towards it or not. If the query point $Q$ is moving towards a neighboring object, then this neighboring object also will send its VC information to the $Q$; otherwise, the query message will be ignored. After the initialization phase, each moving object will maintain the LVC information and safe time. The update process will be executed for the latest LVC information only when the safe time decreases to zero. Assume that each object has $d$ neighbors in average. Since the bisector between two objects costs $O(1)$ time, the LVC construction (Initialization phase) can be finished in $O(d)$ time.

*4.2. Phase 2: CkNN Query Processing.* The pseudocode, Query_DC$k$NN_Search(), in Algorithm 1 shows the whole query process on the query point. The operations from Line 6 to the end of Query_DC$k$NN_Search() do the query processing of C$k$NN ($k > 1$). When a query point $Q$ wants to find the $k$ nearest neighbors where $k > 1$, $Q$ first broadcasts messages $q(k, hopc)$ (case $k > 1$) and gets the local Voronoi cell (LVC) information from the nearest neighboring (NN) object. The nearest neighboring object of $Q$ is responsible for routing messages to its neighboring objects, collecting the local Voronoi cell (LVC) information of the neighboring objects, and sending information back to query point $Q$. Basically, if the received information is insufficient to derive the $k$NN answer, the query point $Q$ will repeatedly sends the query message $q$ with one more hop (i.e., $hopc = hopc + 1$) to the specific neighboring objects to obtain more neighboring VC information until the received information is enough to get $k$NN answer.

When the received information is enough to derive the results of $k$NN, the query point $Q$ will use the approximate method we modified from [25] to calculate the approximate order-$k$ LVC in which the query point $Q$ locates. By using the modified method, the query point $Q$ has to partition the plane into $\lambda$ equal sectors and selects an additional nearest neighboring object in each sector, excluded from $k$NN objects. Thus, the query point $Q$ needs to collect at least $k + \lambda$ neighboring objects' information to build the approximate order-$k$ LVC. Note that the information of the additional $\lambda$ objects can be learned from the information of the received neighboring objects. In comparison with the centralized approach for constructing the approximate order-$k$ Voronoi cell, the modified approximate method only needs to consider the $k$ nearest neighboring objects and $\lambda$ additional nearest neighboring objects instead of considering all objects in the wireless environment. The computation time is thus much better than the traditional centralized approach and can be derived as $O(\lambda) + O(\lambda) \times O(k) = O(\lambda(k + 1))$, where $O(\lambda)$ is the time for selecting the $\lambda$ additional neighboring objects. In the end, $O(\lambda k)$ is the time for determining the boundary of the approximate order-$k$ LVC. For more details about the analysis of the computation time, please refer to

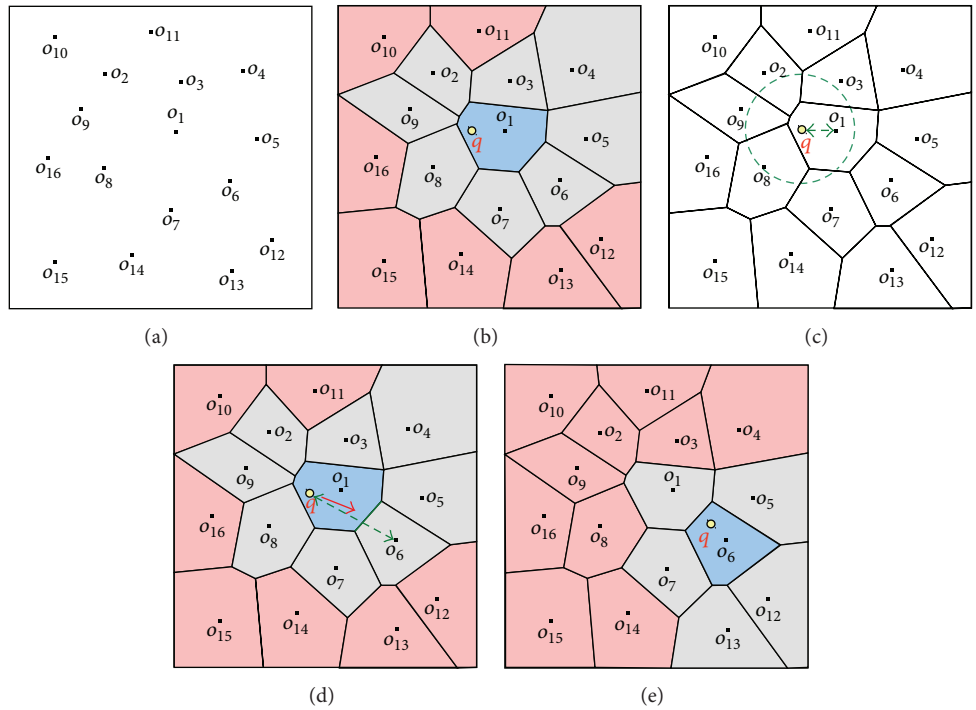



FIGURE 7: An example of DCNN where (a) each sensor exchanges information in the initial step; (b) each sensor builds its LVC; (c) a query at object $q$ is issued and sensor $o_1$ returns the information of its LVC to $q$ since $q$ is located in the Voronoi cell associated with $o_1$; (d) object $q$ knows which sensor will be reached next and can derive the safe time by the LVC information of $o_1$; and (e) the query result to $q$ can be updated seamlessly when it moves to $o_6$.

[25]. After the query point $Q$ calculates the approximate order-$k$ LVC, a safe time of the query $q$ can be derived. The safe time of the query $q$ is regarded as the time limit for starting the next location information updating process and updating the C$k$NN answer.

Figure 7 gives an example for the case of $k = 1$. As shown in Figure 7(a), there are 16 moving sensors, $o_1, \ldots, o_{16}$, in the system and each sensor has the location information about itself. In the initial phase, each sensor broadcasts the message with its location information and receives the location information from its neighbors. The local Voronoi cell (LVC) of each sensor then can be calculated and stored. The overall calculated Voronoi diagram is depicted in Figure 7(b). Afterwards, if the locations of the sensors change as moving, each sensor will broadcast new location information to its neighbors and each sensor can derive a new LVC by the received updating messages. In Figure 7(c), a continuous NN query is issued by some query object $q$ with its location. In order to know the Voronoi cell where $q$ locates, $q$ first broadcasts the query message to the neighboring sensors (in dashed circle). After receiving the query message, the neighboring sensor will check whether the query $q$ is located in its associated cell. If $q$ is not in the cell, the sensor will ignore the query message. In Figure 7(c), sensor $o_1$ will return the result to the query $q$, saying that object $o_1$ is the nearest neighbor of $q$ for the time being. If $q$ moves in the direction as indicated by the solid arrow in Figure 7(d) to object $o_6$, $q$ can know that it moves towards the boundary of $o_1$ and $o_6$ by the LVC stored in $o_6$ and then derive the safe time. When approaching the boundary (i.e., the safe time is almost passed), a query message to $o_6$ will be issued by $q$ in advance. Sensor $o_6$ will reply to $q$ with the cell information to maintain the accuracy of the query result. Figure 7(e) shows the result after $q$ moves into the associated cell of sensor $o_6$.

Figure 8 demonstrates an example of DC$k$NN for $k > 1$. The initial phase is the same as the previous DCNN example. Each sensor derives the corresponding LVC with the information from the neighboring sensors. The query is issued by a moving object $q$ with its location. Object $q$ locates itself and knows that it is in the VC of $o_1$ after sensor $o_1$ replies to $q$ with its LVC information. Suppose $k = 3$. After the initialization, $q$ is located in $o_1$ and has the LVC information of $o_1$. Having such information, $q$ can derive the NN but 3NN cannot be decided yet. So, $q$ will broadcast the query message to all of its one-hop neighboring sensors, say $o_1$, $o_2$, $o_3$, $o_8$, and $o_9$ in Figure 8(a), to request information. The neighboring sensors will reply to $q$ the information about their LVCs. Object $q$ then can derive the resulting 3NN for the time being, say $o_1$, $o_8$, and $o_9$ in the example. With these replied information, $q$ can build the approximate order-3 Voronoi cell, as shown in Figure 8(b). Then, $q$ uses the approximate order-3 Voronoi cell to derive the safe time in order to maintain the continuous query result. Now, if $k = 10$, the information received from the one-hop neighboring sensors is not enough to derive the result. Then, the query message will be broadcast to $q$'s two-hop neighboring sensors as Figure 8(c) shows. If the information is not enough to derive the resulting $k$NN, the query message will be broadcast

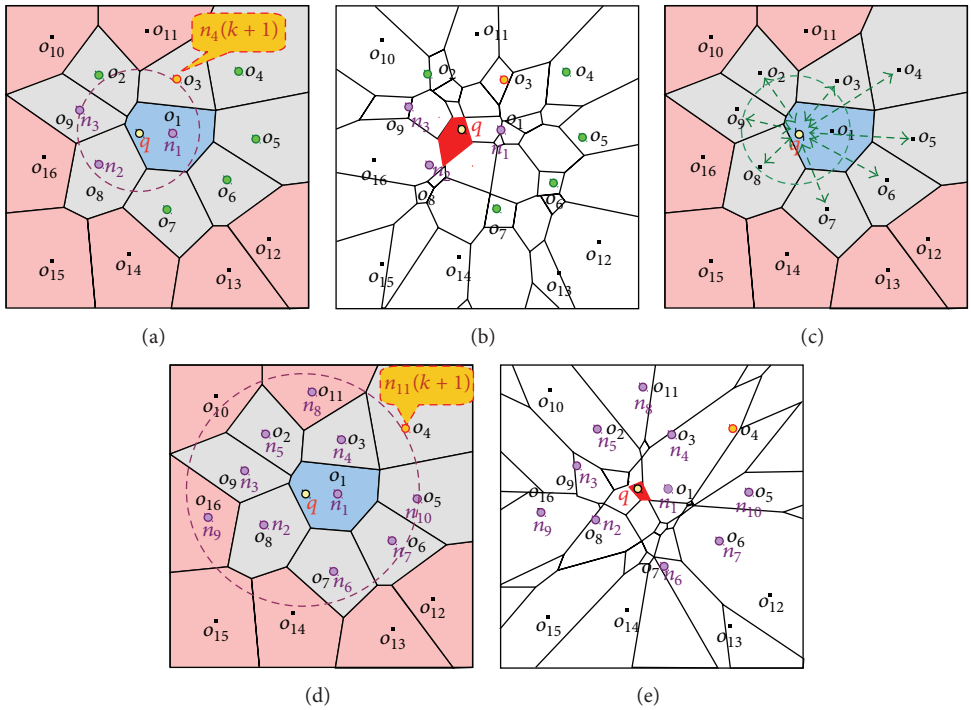



FIGURE 8: An example of DC$k$NN for $k > 1$, where the initialization phase is the same as previous DCNN example and (a) $o_1$, $o_8$, and $o_9$ are the 3NN to the query object $q$, (b) the approximate order-3 Voronoi cell can be built with enough information received, (c) the information is not enough for 10NN query and two-hop neighboring sensors are requested, (d) the resulting 10NN can be derived, and (e) the approximate order-10 Voronoi cell can be calculated with information received.

to one more hop neighboring sensor. Such process will be repeated with an increment on the hop count. After having enough information, $q$ can calculate the resulting 10NN and the approximate order-10 Voronoi cell as shown in Figures 8(d) and 8(e), respectively. Using the approximate order-10 Voronoi cell, the safe time can be derived. In the following, we will discuss how to derive the safe time for maintaining the result of the continuous $k$NN query.

*4.3. Phase 3: Information Updates.* In the distributed and mobile environments, it is challenging for a mobile node to maintain the location information of the other mobile nodes effectively. If a mobile node sends the update messages frequently, it will consume much network resource and power. Furthermore, the result of the moving query should be effectively maintained. For different roles, nodes, and query, we propose effective ways, respectively, in our DC$k$NN query process to update the location information and maintain the query result.

*4.3.1. The Safe Time of Each Object.* In the proposed approach, DC$k$NN, each object cooperates with each other and exchanges information to update the location information of each object. In order to monitor and obtain the continuous answer of a continuous $k$ nearest neighbors query, we use a *safe distance* between each object and query. Then we can derive the safe time of each object for location updates by estimating the maximum speed of that object. Suppose that the sensing area is partitioned into a grid for location management. There are three possible methods for deriving the safe distance. The first one is the *grid-based safe distance*. Each object directly uses the minimum distance between itself and the boundary of the grid cell where the object locates. Simply using the grid-based safe distance may cause some problems about the imprecise updates. The second method, *VD-based safe distance*, uses VD to obtain the safe distance. In order to predict the safe distance more precisely, we can consider the previous two distances simultaneously and then select the smaller one. So the last method, *proper safe distance*, will return a better safe distance by comparisons. Having the safe distance, the safe time then can be derived by estimating the maximum speed of the considered object.

In Figure 9, the sensing area is divided into a grid and the Voronoi diagram formed by the sensors is also shown. For sensor $o_1$ in Figure 9(a), the grid-based safe distance and VD-based safe distance are presented, respectively, and the proper safe distance will be the VD-based safe distance after the comparison. On the contrary, for sensor $o_3$ in Figure 9(b), the proper safe distance will be the grid-based safe distance.

*4.3.2. The Safe Time of Each Query.* For the moving query, it is important to know when the result will change. To derive the safe time, during which the result will not change, we refer to the characteristics of Voronoi diagram. In the traditional environment with static objects, the system only uses the minimum distance from the query to the boundary of the Voronoi cell where the query locates as the safe distance. The the safe time can be obtained by the estimated maximum

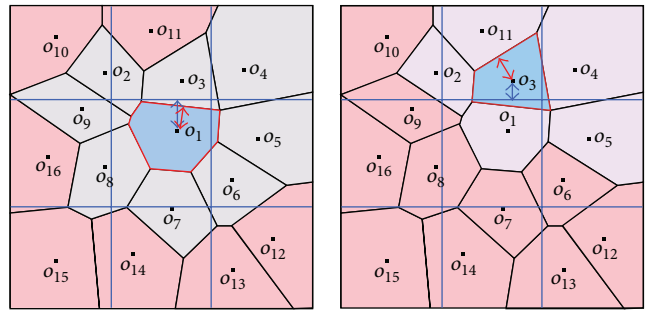



FIGURE 9: An illustration for different safe distances: (a) sensor $o_1$ uses the VD-based safe distance as the proper safe distance and (b) sensor $o_3$ uses the grid-based safe distance as the proper safe distance.

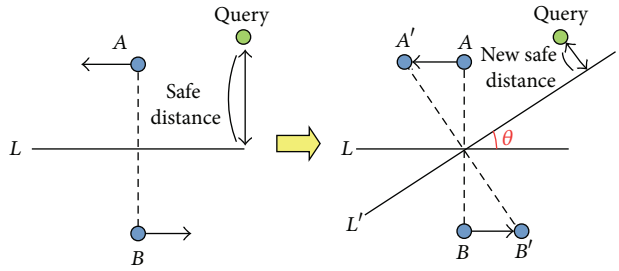



FIGURE 10: An example showing the impact on the safe distance caused by object movement.

speed of that query. During the safe time, no update is necessary, thus saving the update cost.

Unfortunately, in the moving object environment, the above method cannot obtain the right safe time. As shown in Figure 10, the left part shows the static case, where $L$ is the Voronoi cell boundary between objects $A$ and $B$, and the safe distance of the query can be calculated as the figure shows. However, for the moving objects, the original safe distance between the $L$ and query cannot be directly used to calculate the safe time because the object movement will cause the distance form query to $L$ to be nonlinearly increasing or decreasing. As the right part of Figure 10 shows, after the movement of object $A$ and object $B$, $L$ becomes $L'$ and the safe distance changes. Thus, if we want to use the boundaries of Voronoi cell to estimate more accurate safe time, we need to consider the relative movements between the objects and query.

In order to derive the precise safe distance, we consider the movements of objects and query simultaneously and use Figure 11 to illustrate. Initially, $\overline{QA}$ and $\overline{QB}$ are the distances from the query to objects $A$ and $B$, respectively. The line between $A$ and $B$ is the boundary of the Voronoi cells associated with $A$ and $B$ and the query is located in the cell of $A$ as Figure 11(a) shows. When objects and the query move as in Figure 11(b), the query may touch the boundary. This indicates that the query will enter another cell and the result of the query should be updated. The time from the beginning to the moment of touching the boundary is the precise safe time for the query. According to the properties of Voronoi diagram, when the query is on the boundary, $\overline{QA} = \overline{QB}$. We hence can use the condition of $\overline{QA} = \overline{QB}$ to decide whether the query is on the boundary. With this property, even though the objects and query move frequently, the query only needs the speeds, directions, and locations of the neighboring objects to predict the safe time easily. Within the safe time, the query will not touch the boundary. In Figure 11(c), the query meets the boundary at $t_1$ and the safe time is $t_1$.

In the following, we present the way to calculate the safe time when $\overline{QA} = \overline{QB}$. Assume that the initial location of the moving query is $(x_q, y_q)$ and the speed is $(V_{x_q}, V_{y_q})$. Suppose the initial location of object $A(B)$ is $(x_A, y_A)((x_B, y_B))$ with the speed being $(V_{x_A}, V_{y_A})((V_{x_B}, V_{y_B}))$. The equation to find when the query meets the boundary can be derived according to the property of perpendicular bisector. Since $\overline{QA} = \overline{QB}$, we can have

$$\begin{aligned}
&\left(\left[\left(x_q + V_{x_q} * t\right) - \left(x_A + V_{x_A} * t\right)\right]^2\right. \\
&\quad \left.+\left[\left(y_q + V_{y_q} * t\right) - \left(y_A + V_{y_A} * t\right)\right]^2\right)^{1/2} \\
&= \left(\left[\left(x_q + V_{x_q} * t\right) - \left(x_B + V_{x_B} * t\right)\right]^2\right. \\
&\quad \left.+\left[\left(y_q + V_{y_q} * t\right) - \left(y_B + V_{y_B} * t\right)\right]^2\right)^{1/2}.
\end{aligned} \tag{1}$$

Then,

$$\begin{aligned}
0 = &\left[V_{x_A}^2 + V_{y_A}^2 - V_{x_B}^2 - V_{y_B}^2 - 2 * V_{x_q} V_{x_A}\right. \\
&\left.-2 * V_{y_q} V_{y_A} + 2 * V_{x_q} V_{x_B} + 2 * V_{y_q} V_{y_B}\right] * t^2 \\
&+ 2\left[\left(x_q - x_A\right)\left(V_{x_q} - V_{x_A}\right) + \left(y_q - y_A\right)\left(V_{y_q} - V_{y_A}\right)\right. \\
&\left.-\left(x_q - x_B\right)\left(V_{x_q} - V_{x_B}\right) - \left(y_q - y_B\right)\left(V_{y_q} - V_{y_B}\right)\right] * t \\
&+ \left[\left(x_q - x_A\right)^2 + \left(y_q - y_A\right)^2 - \left(x_q - x_B\right)^2 - \left(y_q - y_B\right)^2\right].
\end{aligned} \tag{2}$$

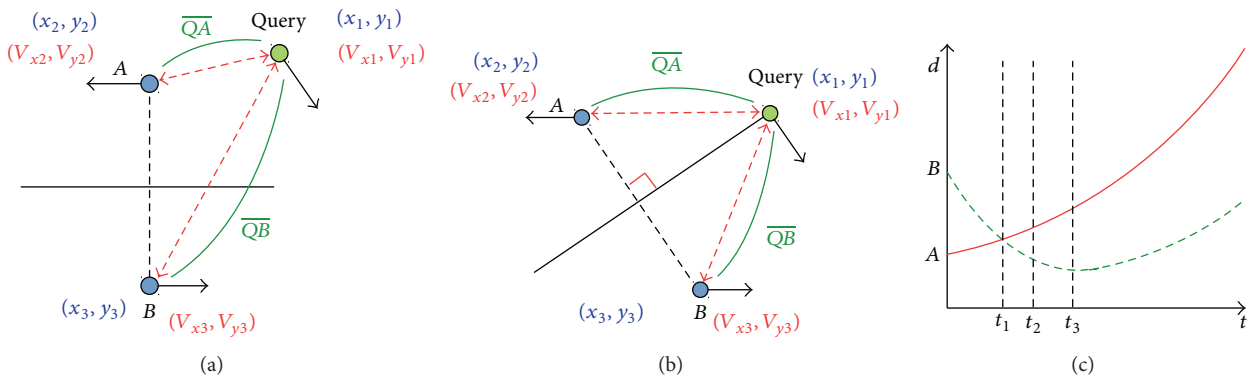



FIGURE 11: The precise safe-time for moving objects $A$ and $B$ and the moving query, where (a) shows the beginning state, (b) presents the moment when the moving query meets the boundary, and (c) gives the diagram for the change on the safe distance with time $t$.

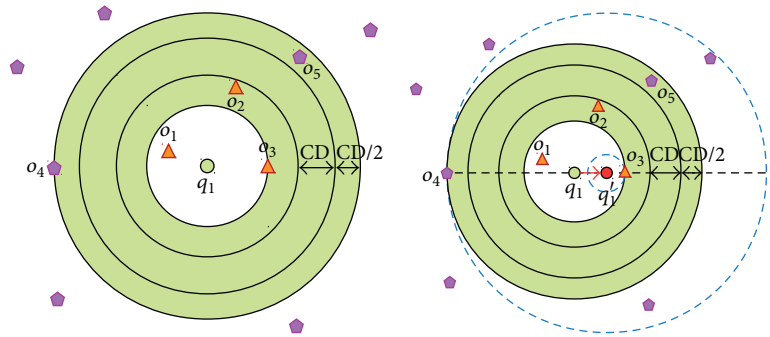



FIGURE 12: The phenomenon when using the original safe-time methods to (a) stationary query and (b) moving query.

Equation (2) is a quadratic equation. Since a general quadratic equation can be written in the following form: $ax^2 + bx + c = 0$, recall that the discriminant of a quadratic equation is $\Delta = b^2 - 4ac$. If $\Delta > 0$, there are two distinct real roots; if $\Delta = 0$, there is exactly one real root; if $\Delta < 0$, there are two distinct (non-real) complex roots. In (2),

$$\begin{aligned} a &= V_{x_A}^2 + V_{y_A}^2 - V_{x_B}^2 - V_{y_B}^2 - 2 * V_{x_q} V_{x_A} \\ &\quad - 2 * V_{y_q} V_{y_A} + 2 * V_{x_q} V_{x_B} + 2 * V_{y_q} V_{y_B}, \\ b &= 2\left[\left(x_q - x_A\right)\left(V_{x_q} - V_{x_A}\right) + \left(y_q - y_A\right)\left(V_{y_q} - V_{y_A}\right)\right. \\ &\quad \left. - \left(x_q - x_B\right)\left(V_{x_q} - V_{x_B}\right) - \left(y_q - y_B\right)\left(V_{y_q} - V_{y_B}\right)\right], \\ c &= \left(x_q - x_A\right)^2 + \left(y_q - y_A\right)^2 - \left(x_q - x_B\right)^2 - \left(y_q - y_B\right)^2, \end{aligned} \tag{3}$$

and then we can derive the safe time $t$ according to the discriminant:

$$t = \frac{\left[-b + \left(b^2 - 4ac\right)^{1/2}\right]}{2a} \quad \text{or} \quad \frac{\left[-b - \left(b^2 - 4ac\right)^{1/2}\right]}{2a}. \tag{4}$$

Finally, we select the minimum value of $t > 0$ as the safe time.

Additionally, assuming that $d$ is the distance between the object and the query and $t$ is the time, $\overline{QA} = \overline{QB}$ can be converted into a function of $t$ and $d$, as shown in Figure 11(c). Then we can obtain the safe time $t_1$ which is the time when $\overline{QA} = \overline{QB}$. Furthermore, the object $B$ will be closest to the query at the time $t_3$.

## 5. Comparisons with Previous Approaches

In this section, we compare the proposed DC$k$NN approach, the Safe-time with stationary query approach [5], and the naïve way used pull-based approach, in terms of safe time, space, and accuracy.

*5.1. Safe Time.* The original safe-time with stationary query [5] (safe-time(S)) does not support moving queries. An example of C3NN is shown in Figure 12, where $q_1$ is the query point, triangles are the 3NN objects, the pentagons represent the other individual objects, and $o_4$ is 4th NN. Figure 12(a) shows the original result for safe-time(S). After obtaining the $k$th and $(k+1)$th objects, ($o_2$ and $o_4$ in the figure), the *cross-distance* (CD) of $k$th and $(k+1)$th objects can be calculated. Recall that the cross distance of two data objects $A$ and $B$ is the difference on the distances of $A$ and $B$ to the center (or, the query point). For the next update (after a safe-time period), all the objects in this region should be checked.

We adapt the safe-time(S) method for moving query. The adapted approach will result in a higher update cost.

Figure 12(b) presents a case that the query moves toward object $o_3$ ($k$th NN) and away from object $o_4$ (($k+1$)th NN). When the query $q_1$ moves to the point $q_1'$, the most outside concentric circle increases and becomes as large as the most outside dotted concentric circle. On the contrary, the most inside concentric circle decreases and becomes as small as the most inside dotted concentric circle. Thus, the cross-distance increases and more number of objects will be examined.

In addition, according to the Figure 12(b), we can know that the original Safe-time method, safe-time(S), is not appropriate for moving queries since the calculation of Safe-time(S) method does not consider the speed of query. safe-time(S) directly uses twofold of the maximum speed of the $k$th and ($k$+1)th NN objects to divide the CD to derive the safe time. The cost for updates increases when the CD becomes larger since the average checked area becomes bigger. Besides, the obtained $k$NN answer cannot be guaranteed to be correct. On the contrary, the cross-distance when the query moves towards object $o_4$ (($k+1$)th NN) and away from object $o_3$ ($k$th NN), in this case, the order of the $k$th and ($k+1$)th NN object may be changed, but the update does not perform, thus resulting in incorrect results.

We modify the Safe-time to fully support the moving query. In order to support and maintain the accuracy of moving query, the maximum speed (*Maxspeed*) used in the safe time calculation is required to be modified and replaced by the maximum relative speed between the query and the objects. Because of using the relative speed, the adapted safe-time method, *safe-time*($M$), needs twofold maximum relative speed ($2 * Maxspeed$) to calculate the safe time for guaranteeing the accurate results. The safe time derived is less than the one derived in safe-time(S) due to the relative speed. Hence, the update frequency of safe-time(M) is higher than the update frequency of safe-time(S).

*5.2. Space.* In DC$k$NN, each object and query only need to maintain the information of LVC. The information to be stored includes the location of neighboring objects and the object information of order-$k$'s local Voronoi diagram. The total amount of stored information is not significant on each object. Thus, the use of storage space does not need to be very substantial.

*5.3. Accuracy.* The safe-time (S) [5] considers the stationary query. When applying it to the moving query, some adaption should be made. According to our experimental results, when the value of $k$ increases with a moving query, the accuracy of the result will be reduced. In our simulation, we modify the original Safe-time(S) as Safe time(M) for moving query and consider the relative distance in detail caused by the moving query and objects. Although the correctness of the answer can be increased, the update frequency increases, thus increasing the number of delivered messages.

DC$k$NN uses Voronoi and grid cell boundary to calculate the safe time and updates the information of related objects before the end of the safe time. In fact, the boundary of the Voronoi cell represents the boundary of moving $k$NN query. After the safe time period, the query will collect the information of related neighboring objects. Having such real-time information, the query can have the result more accurate in time. Similar to safe-time(M), DC$k$NN uses the $k$th and ($k+1$)th nearest moving objects to derive the safe time. So, it is able to maintain the correctness of the answer.

## 6. Simulation Experiments

In the simulation, we compare the following methods: Naïve, DC$k$NN, safe-time (S), and safe-time (M). The Naïve method is centralized and used as a benchmark for the comparison. All the mobile sensors will send the location information to the central server to update the information. The other methods, DC$k$NN, Safe-time(S), and Safe-time(M), are distributed and used in the considered mobile wireless sensor network, where each object knows its position, has the ability to perform some calculation, and stores the information of localized Voronoi cell (LVC). We assume that each object can transmit a message to its one-hop neighbor so that each object can communicate with at least one object. For Safe-time(S) and the safe-time(M), a maximum speed (*MaxSpeed*) is set in order to estimate relative speed and distance for safe-time calculation.

In the simulation, all the compared approaches are performed under the same environmental conditions for fairness. The initial location and speed of moving objects are uniformly randomly distributed. Since we focus on the performance of the query process of the proposed DC$k$NN approach, we further assume that the transmission quality is ideal and no congestion and communication delays occur. The metrics measured include the update frequency for a query, the total number of messages, response time, and accuracy. The update frequency for a query is the average number of update messages sent by the data objects related to the query per minute. The total number of messages is all the amount of messages sent from all objects during the query process. The accuracy of each method is measured by comparing the results to the real answers. Since the query is continuous and the update occurs subsequently, the measured response time is mainly the time required from sending a query to obtaining the first result.

All the experiments are simulated, so the results about the response time may not show the practical cases exactly but can reflect the trends on the performance among all the compared approaches relatively. In order to have the response time in our simulation, we thus use the total number of hops for deriving the result to multiply the average time for one hop in the simulation system to present the response time. Thus, the response time is proportional to the total number of hops in our simulation program. In addition, because the main difference between Safe-time(S) and the Safe-time(M) is the way for deriving the safe time, the measured response time of these two methods is the same. The simulation program is implemented with Java. The setting of our simulation is shown in Table 1 with a default value for each parameter. In all the simulation experiments, the final results are reported with the average of 100 queries. The duration of each query is 60 seconds.

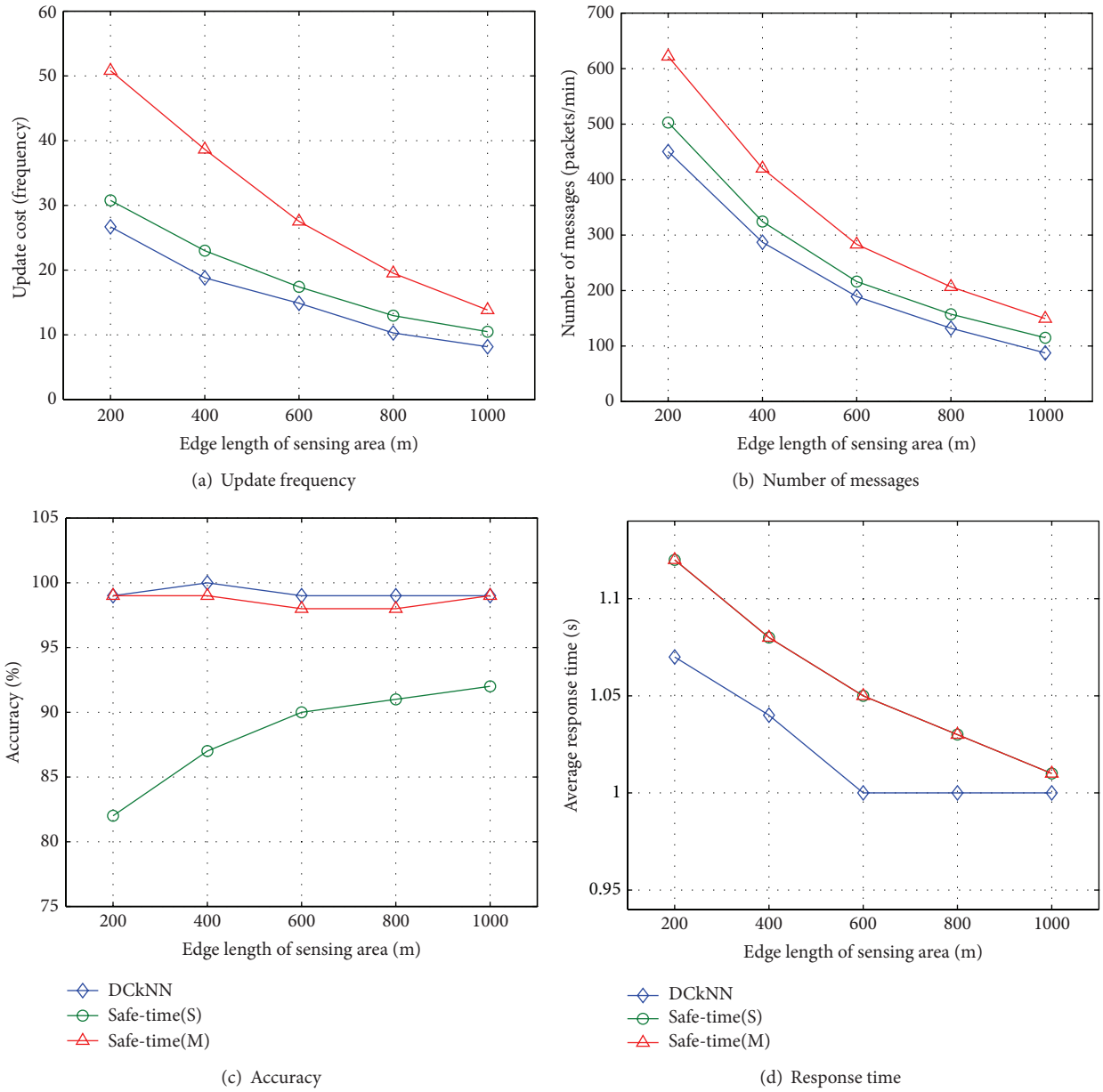



Figure 13: The effects of different sensing areas on (a) update frequency, (b) number of messages, (c) accuracy, and (d) response time.

Table 1: Simulation parameters.

| Parameter | Range | Default value |
|---|---|---|
| Edge length of sensing area (m) | 200, 400, 600, 800, 1000 | 600 |
| Edge length of grid (m) | 50, 100, 150 | 50 |
| Number of objects | 20, 30, 40, 50, 60 | 30 |
| Transmission range (m) | 125 | 125 |
| $k$ | 1, 3, 5, 10, 20 | 3 |
| Moving objects (%) | 20, 40, 60, 80, 100 | 20 |
| Speed (m/s) | 2, 4, 6, 8, 10 | 4 |

*6.1. Effects of Sensing Area.* Sensing area affects the number of update and query messages since if the area becomes greater with the fixed number of objects, relatively, the density of the objects becomes sparse. For example, when the sensing areas are 200∗200, 400∗400, 600∗600, 800∗800, and 1000∗1000, the densities are 1.88, 0.47, 0.21, 0.12, and 0.08, respectively. Figure 13(a) shows that DC$k$NN has the lowest number of updates with all the different areas and the number of updates for Safe-time(S) is the highest (worst) because DC$k$NN considers the moving direction and speed of the queries and objects and selects the proper safe-distance to derive the safe-time. Safe-time(S) the Safe-time(M) estimate the safe time according to the maximum speed of the objects, so more number of updates can be expected. Additionally, Safe-time(M) is updated more frequently than Safe-time(S). Recall that Safe-time(M) considers the query movement.

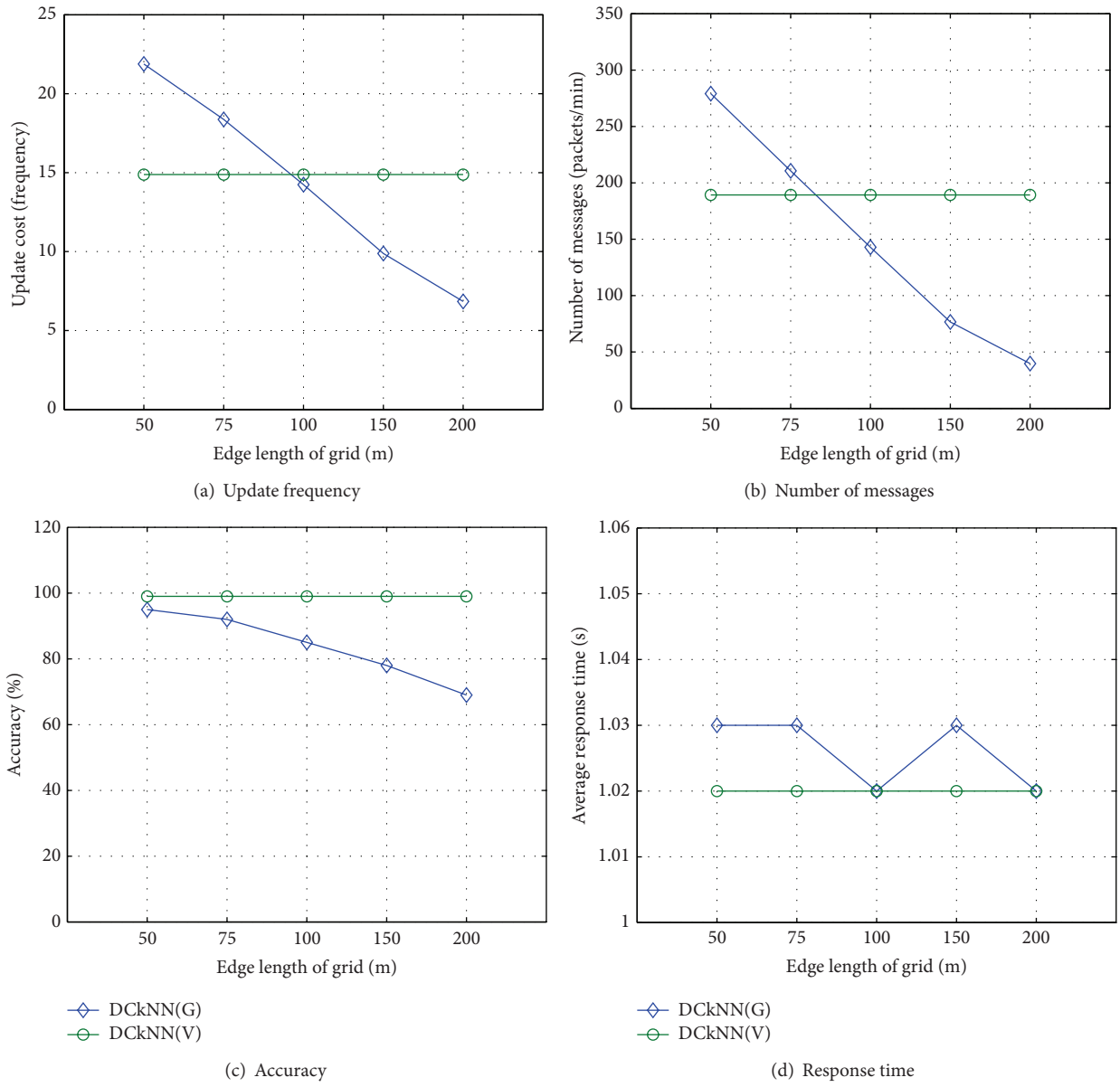



FIGURE 14: The comparisons of using grid-based and VD-base safe distances to derive the safe-time: (a) update frequency, (b) number of messages, (c) accuracy, and (d) response time.

The average safe-distance becomes shorter, thus leading to more updates. Besides, when the sensing area increases from 200∗200 to 600∗600, the update frequency drops dramatically. The curve becomes flatter when the area is from 600∗600 to 1000∗1000 since the difference on the densities of the areas from 600∗600 to 1000∗1000 is not much. The number of updates is less as the safe-time becomes larger. The total number of transmitted messages of naïve approach is fixed at about 1800 packets. All the other distributed methods use fewer number of messages than naïve.

To monitor the results, DC*k*NN only needs to check the related objects in the local Voronoi cell at the proper safe-time, thus reducing a lot of redundant update messages. Safe-time(S) and Safe-time(M) use the maximum speed of the objects to calculate the safe time, so the total number of transmitted messages of both approaches is more than that of DC*k*NN. Figure 13(b) shows that more than 30% of the total number of transmitted messages can be saved by DC*k*NN in comparison with Safe-time(M).

For the accuracy, since Safe-time(S) does not consider the query movement, the safe-time may be inaccurate. Furthermore, due to the mobility, when the density is high, the estimated results become more inaccurate. So, Safe-time(S) performs worst as shown in Figure 13(c). Safe-time(M) is revised from Safe-time(S) by considering the query movement and the safe-distance is less in average for guaranteeing better results. Hence Safe-time(M) has a better accuracy and almost the same as DC*k*NN. The accuracy of Safe-time(M) and DC*k*NN is almost about 100% correct and the gap may be caused by arithmetical errors in the experiments.

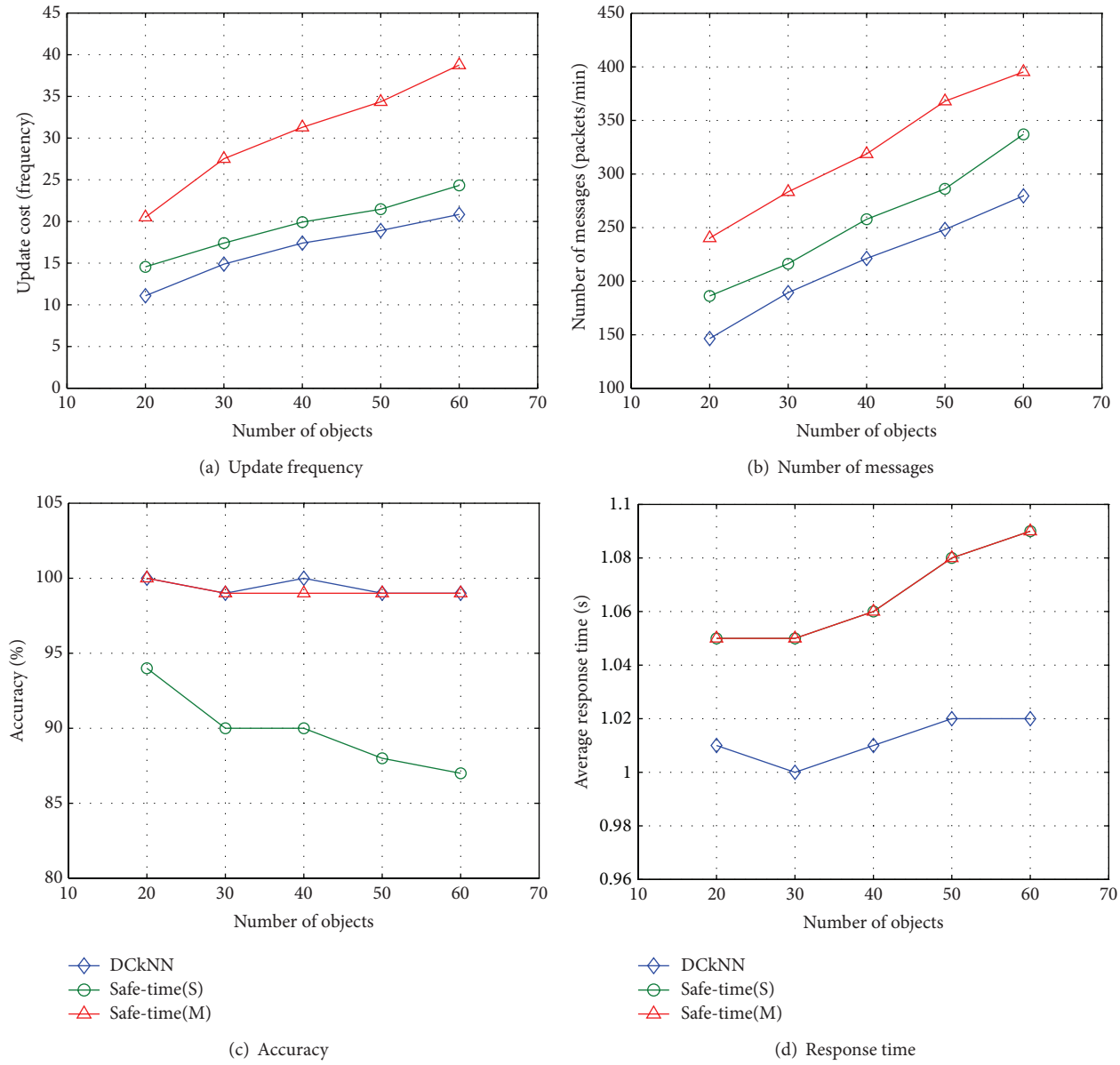



FIGURE 15: The effects of the number of objects on (a) update frequency, (b) number of messages, (c) accuracy, and (d) response time.

When the area becomes larger, the density becomes lower. So, the number of objects in each object's transmission range is fewer. Under such a circumstance, DC*k*NN uses less computation and the response time is shorter as shown in Figure 13(d). In contrast, the Safe-time(S) and the Safe-time(M) need to collect information before performing each new query and cause longer response time. Since Safe-time(M) and Safe-time(S) use the same method and procedure to handle the first query, the response time of these two methods will be almost the same.

*6.2. Effects of Grid Size.* In this simulation experiment set, we want to observe the update efficiency of each object in the third phase of DC*k*NN. Recall that we consider the grid-based and VD-base safe distances to derive a proper safe-time. In this experiment, DC*k*NN(G) uses grid-based safe distance to determine the safe time for the next update and DC*k*NN(V) uses VD-base safe distance to derive the safe time. The sensing area is fixed to $600*600\,m^2$ and there are 30 data objects. We compare DC*k*NN(V) with DC*k*NN(G) by observing the impacts of different grid sizes. All the other settings have the similar trends in our experimental results. In Figure 14(a), the update frequency of the objects in DC*k*NN(G) is high when the grid size is $50*50\,m^2$ and becomes less when the grid size decreases. Since the sensing area is $600*600\,m^2$ and 30 objects are given, the average distance between an object and the boundary of a Voronoi cell is about 25 to 50 meters. With the setting of grid size $50*50\,m^2$, the average distance obtained in DC*k*NN(G) is about 25 meters. Thus, the safe time of DC*k*NN(V) is longer. In general, as Figures 14(a) and 14(b) show, the update

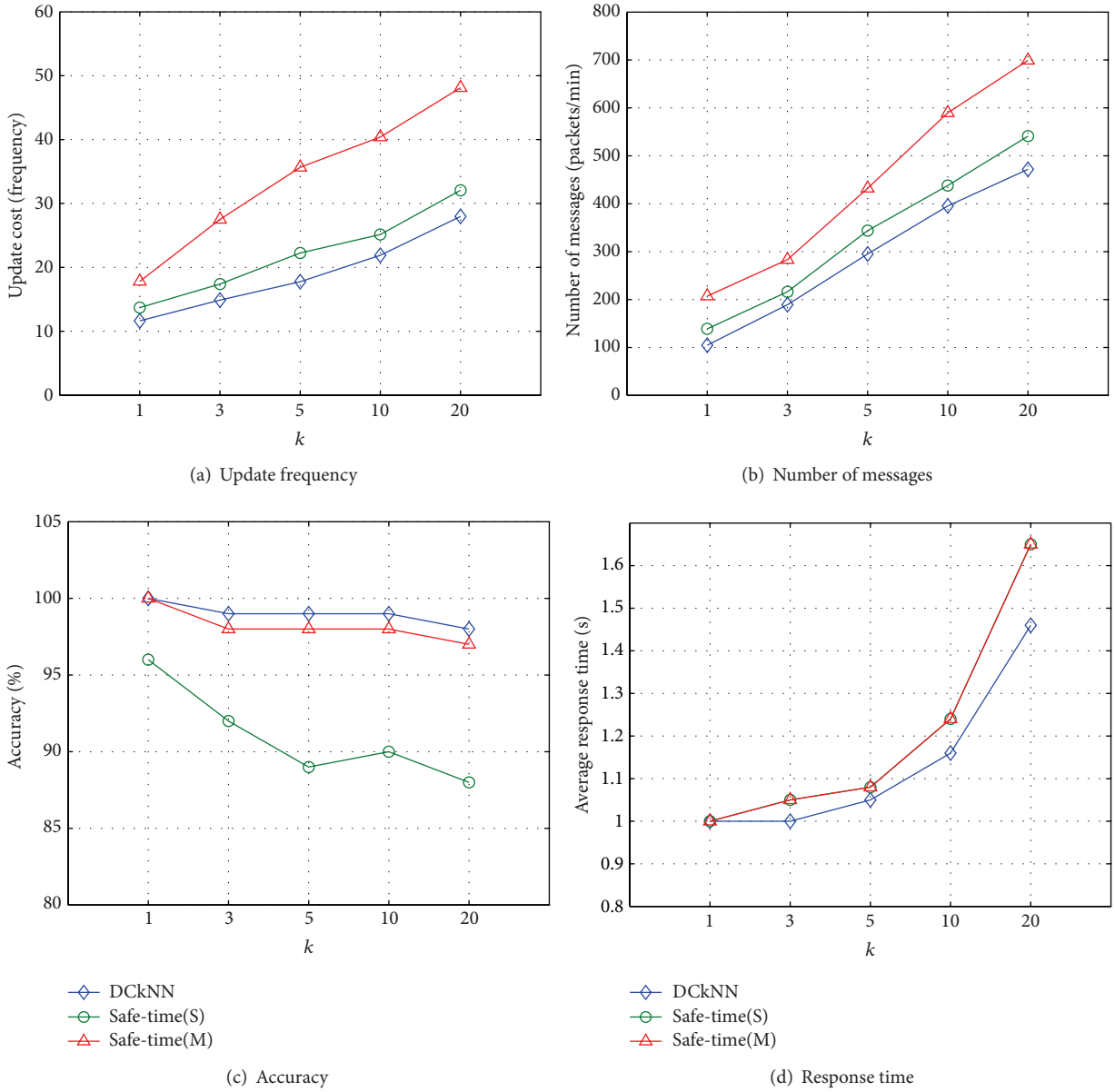



FIGURE 16: The impacts of different values of $k$ on (a) update frequency, (b) number of messages, (c) accuracy, and (d) response time.

frequency of the objects and the number of messages, sent by the objects, decrease as the grid size increases.

When the grid size is large, the safe time becomes long and less number of updates are issued. Hence, the results become more inaccurate. As shown in Figure 14(c), when the grid sizes are $50*50\,m^2$, $75*75\,m^2$, $100*100\,m^2$, and $150*150\,m^2$, the corresponding accuracies are 95%, 92%, 85%, and 78%, respectively. While the grid size increases to $200*200\,m^2$, the accuracy is significantly reduced to 69%, and the right answer thus can not be guaranteed. On the other hand, Figure 14(d) shows that the response times of DC$k$NN(G) and DC$k$NN(V) are not much different, because the size of grid is not the major factor to influence the complexity of obtaining information for the first query.

*6.3. Number of Objects.* In this subsection, we discuss the results as the number of data objects varies. The area in the presented experiment set is $600*600\,(m^2)$ with grid size of $50*50\,(m^2)$. As shown in Figure 15(a), when the number of objects increases, the number of updates in DC$k$NN is still relatively less than the ones in Safe-time(S) and Safe-time(M). If the number of objects increases, the number of replied messages per second to the central server in naïve will grow dramatically. With 20 to 60 objects, the number of replied messages to the server is from 1200 to 3600 (packets/min). By observing the results in Figures 15(b) and 15(c), as more objects are in the area, the number of messages increases for the other three methods and the accuracy reduces in Safe-time(S). These match the trends for different areas we

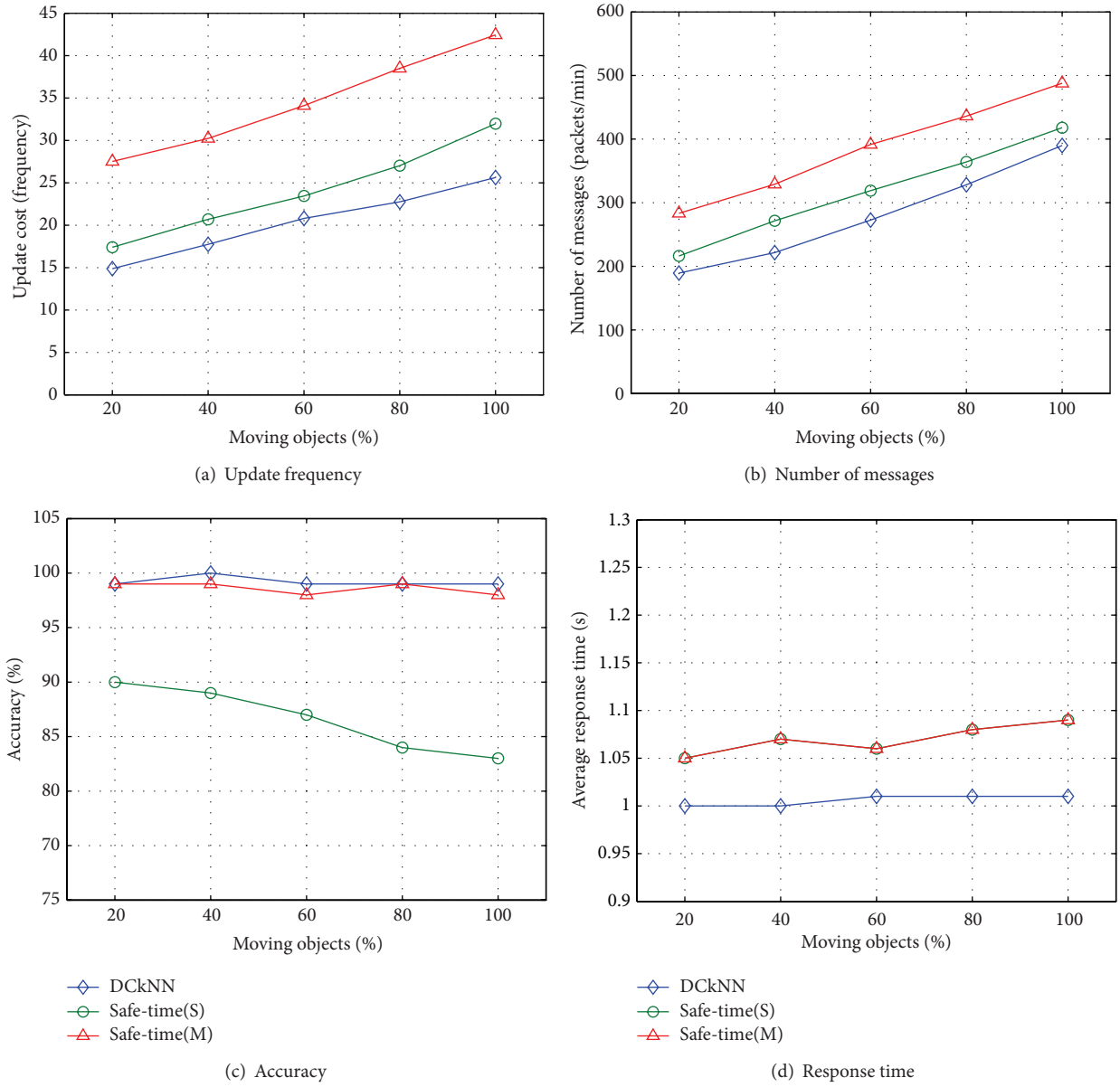



FIGURE 17: The effects of the moving objects on (a) update frequency, (b) number of messages, (c) accuracy, and (d) response time.

discussed earlier due to the density. Similarly, the response time can be expected as shown in Figure 15(d) where the response time of DC$k$NN is still the shortest.

*6.4. Different Values of k.* It is obvious that the value of $k$ affects the performance of query processing significantly. When the value of $k$ becomes larger, the number of related objects required to be checked also increases. In other words, we need to spend more cost to find out which $k$ objects are currently in the result for the C$k$NN query. Therefore, the changes in the value of $k$ mainly affect the amount of data required for the first query and the update cost for maintaining the $k$th and the $(k + 1)$th objects. So, in Figure 16(a), when the value of $k$ is large, more updates are necessary for all the three methods. As Figure 16(b) shows, the influence on the number of transmitted messages in Safe-time(M) is more serious than the one in Safe-time(S). Due to the mobility, the number of influenced objects in Safe-time(M) will increase as the value of $k$ increases. In particular, the update frequency in Safe-time(M) becomes nearly twofold the update frequency in Safe-time(S).

Figure 16(c) presents the impact on the accuracy for different values of $k$. With a larger value of $k$, an object in Safe-time(S) will have a higher chance to derive the incorrect safe time, thus causing lower accuracy. When the value of $k$ increases, the response time of all three methods will increase significantly, as shown in Figure 16(d). The reason is that all DC$k$NN, Safe-time(S), and Safe-time(M) need to search more objects and collect more information for answering the query when the value of $k$ increases to a certain number. For

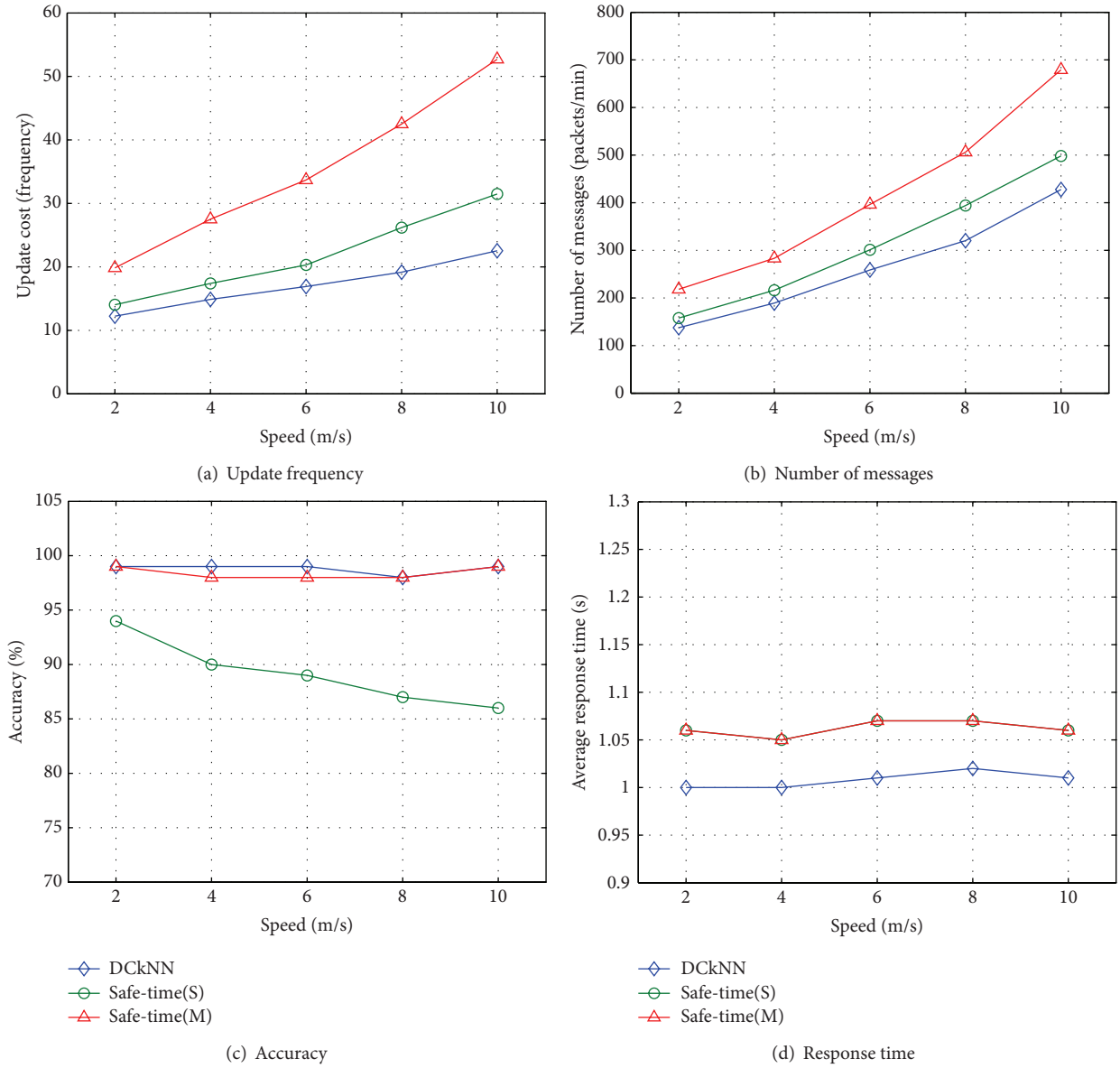



Figure 18: The impacts of speed on (a) update frequency, (b) number of messages, (c) accuracy, and (d) response time.

DC*k*NN, DC*k*NN only needs to store local Voronoi diagram, where the information required is less than required in Safe-time(S) and Safe-time(M), so the response time of DC*k*NN is the shortest.

*6.5. The Effect of Moving Objects.* In this experiment set, we consider different percentages of moving objects in all the data objects. The default value is 20%, which means that if there is a total of 30 objects, then there are six moving objects. In Figures 17(a) and 17(b), as the percentage of moving objects increases, the overall number of updates required between the objects increases relatively and the change rate of answer to the DC*k*NN accelerates, thus increasing the update frequency of the query and the number of messages. However, DC*k*NN still keeps the best performance. Figure 17(c) shows the accuracy of the result for different percentage of moving objects. DC*k*NN and Safe-time(M) both have higher accuracy and Safe-time(S) is still the worst. The safe time of each object in Safe-time(S) is calculated using the maximum speed of the objects. Hence whether the object is moving or not does not directly result in the incorrectness of the answer collection. In fact, in Safe-time(S), one of the main reasons to affect the accuracy is the directions of moving objects. Last, as Figure 17(d) shows, the percentage of moving objects does not impact the response time too much.

*6.6. Effects of Speed.* If each object moves faster, the relative velocities between objects may also become faster. This will make the safe time shorter and the update frequency of each object and query increases. The safe time, used in Safe-time(S) and Safe-time(M), is directly affected by the maximum speed of objects and does not consider the direction of each object's movement and the speed of individual movement. No matter the objects move or not and move

fast or slow, both of the above methods directly use the maximum speed of objects to calculate the safe time for updating. On the other hand, DC*k*NN considers the moving direction of objects and the proper safe-time for updates. In this experiment, the number of moving objects is 20% of all the objects. As Figure 18(a) shows, as the moving objects speed up, the update cost increases and DC*k*NN still performs best. As the speed increases, the safe-time in Safe-time(S) and the Safe-time(M) becomes much shorter due to a larger maximum speed among all the objects. The trend becomes even worse as the speed increases.

Figure 18(b) shows the trend about the number of transmitted messages when the speed is up. DC*k*NN still has the best performance on the number of messages. There are 427.81 messages sent in average, when the speed of object is 10 m/s. However, the Safe-time(M) needs 679.13 messages. The accuracy will not be effected as the speed changes for DC*k*NN and Safe-time(M) because both of them are designed or adapted for the moving objects and queries. Safe-time(S) will have more inaccurate results as the speed is high as shown in Figure 18(c). However, the response time does not have a significant impact on the response time for all the three methods. Figure 18(d) presents this trend.

## 7. Conclusions

In this paper, we propose a new approach, DC*k*NN, which uses the local Voronoi diagram and proper safe-time to effectively process the C*k*NN query in distributed and mobile environments, such as wireless sensor networks. The proper safe-time can be quickly derived by a simple formula. By comparing our proposed DC*k*NN with other existing approaches, Safe-time(S), DC*k*NN, can effectively improve search efficiency, reduce response time and the number of messages, and keep the accuracy of the result high. Additionally, DC*k*NN reduces more 30% of transmission messages than Safe-time(M) because the characteristics of the Voronoi diagram are used and each object's moving direction is considered instead of directly using the maximum speed of the objects as Safe-time method. The derived safe-time is more precise. As a result, DC*k*NN needs less update frequency, reduces a great number of messages transmitted, and yields more accurate results. In this work, simulation experiments are also performed. The experimental results validate the proposed DC*k*NN approach for the continuous *k* nearest neighbor query in distributed and mobile environments where all the data objects and query can move.

## Acknowledgment

This work is partially supported by NSC under the Grant 98-2220-E-027-007, 100-2221-E-027-087-MY2, and 102-2221-E-027-088-.

## References

[1] O. Bohl, S. Manouchehri, and U. Winand, "Mobile information systems for the private everyday life," *Mobile Information Systems*, vol. 3, no. 3-4, pp. 135–152, 2007.

[2] Y. Tao and D. Papadias, "Time-parameterized queries in spatio-temporal databases," in *Proceedings of the ACM SIGMOD International Conference on Managment of Data*, pp. 334–345, June 2002.

[3] Y. Tao, D. Papadias, and Q. Shen, "Continuous nearest neighbor search," in *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB '02)*, pp. 287–298, 2002.

[4] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.

[5] K. Kim, Y. Cai, and W. Tavanapong, "Safe-time: distributed real-time monitoring of cKNN in mobile peer-to-peer networks," in *Proceedings of the 9th IEEE International Conference on Mobile Data Management (MDM '08)*, pp. 124–131, April 2008.

[6] M. Kolahdouzan and C. Shahabi, "Voronoi-based k nearest neighbor search for spatial network databases," in *Proceedings of the 30th International Conference on Very Large Data Bases*, vol. 30, pp. 840–851, 2004.

[7] M. R. Kolahdouzan and C. Shahabi, "Alternative solutions for continuous K nearest neighbor queries in spatial network databases," *GeoInformatica*, vol. 9, no. 4, pp. 321–341, 2005.

[8] D. Papadias, Y. Tao, K. Mouratidis, and C. K. Hui, "Aggregate nearest neighbor queries in spatial databases," *ACM Transactions on Database Systems*, vol. 30, no. 2, pp. 529–576, 2005.

[9] T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Data on air: organization and access," *IEEE Transactions on Knowledge and Data Engineering*, vol. 9, no. 3, pp. 353–372, 1997.

[10] S. Hambrusch, C.-M. Liu, and S. Prabhakar, "Broadcasting and querying multi-dimensional index trees in a multi-channel environment," *Information Systems*, vol. 31, no. 8, pp. 870–886, 2006.

[11] C.-M. Liu and K.-F. Lin, "Disseminating dependent data in wireless broadcast environments," *Distributed and Parallel Databases*, vol. 22, no. 1, pp. 1–25, 2007.

[12] T.-C. Su and C.-M. Liu, "On-demand data broadcasting for data items with time constraints on multiple broadcast channels," in *Database Systems for Advanced Applications*, vol. 6193 of *Lecture Notes in Computer Science*, pp. 458–469, 2010.

[13] B. Zheng, J. Xu, W.-C. Lee, and D. L. Lee, "Grid-partition index: a hybrid method for nearest-neighbor queries in wireless location-based services," *The VLDB Journal*, vol. 15, no. 1, pp. 21–39, 2006.

[14] B. Gedik, A. Singh, and L. Liu, "Energy efficient exact kNN search in wireless broadcast environments," in *Proceedings of the 12th ACM International Symposium on Advances in Geographic Information Systems (GIS '04)*, pp. 137–146, November 2004.

[15] K. Mouratidis, S. Bakiras, and D. Papadias, "Continuous monitoring of spatial queries in wireless broadcast environments," *IEEE Transactions on Mobile Computing*, vol. 8, no. 10, pp. 1297–1311, 2009.

[16] G. Chatzimilioudis, D. Zeinalipour-Yazti, W.-C. Lee, and M. D. Dikaiakos, "Continuous all k-nearest-neighbor querying in smartphone networks," in *Proceedings of the 13th IEEE International Conference on Mobile Data Management (MDM '12)*, pp. 79–88, 2012.

[17] T.-Y. Fu, W.-C. Peng, and W.-C. Lee, "Parallelizing itinerary-based KNN query processing in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 711–729, 2010.

[18] T. P. Nghiem, A. B. Waluyo, and D. Taniar, "A pure peer-to-peer approach for kNN query processing in mobile ad hoc networks,"

*Personal and Ubiquitous Computing*, vol. 17, no. 5, pp. 973–985, 2013.

[19] H. D. Chon, D. Agrawal, and A. El Abbadi, "Range and kNN query processing for moving objects in Grid model," *Mobile Networks and Applications*, vol. 8, no. 4, pp. 401–412, 2003.

[20] X. Xiong, M. F. Mokbel, and W. G. Aref, "SEA-CNN: scalable processing of continuous K-nearest neighbor queries in spatio-temporal databases," in *Proceedings of the 21st International Conference on Data Engineering (ICDE '05)*, pp. 643–654, April 2005.

[21] G. Zhao, K. Xuan, W. Rahayu et al., "Voronoi-based continuous $\kappa$ nearest neighbor search in mobile navigation," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 6, pp. 2247–2257, 2011.

[22] K. Mouratidis, D. Papadias, S. Bakiras, and Y. Tao, "A threshold-based algorithm for continuous monitoring of k nearest neighbors," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1451–1464, 2005.

[23] S.-H. Wu, K.-T. Chuang, C.-M. Chen, and M.-S. Chen, "DIK-NN: an itinerary-based KNN query processing algorithm for mobile sensor networks," in *Proceedings of the 23rd International Conference on Data Engineering (ICDE '07)*, pp. 456–465, April 2007.

[24] D. T. Lee, "On $k$-nearest neighbor voronoi diagrams in the plane," *IEEE Transactions on Computers*, vol. 31, no. 6, pp. 478–487, 1982.

[25] K. Patroumpas, T. Minogiannis, and T. Sellis, "Approximate order-k Voronoi cells over positional streams," in *Proceedings of the 15th ACM International Symposium on Advances in Geographic Information Systems (GIS '07)*, pp. 276–283, November 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 370953, 12 pages
http://dx.doi.org/10.1155/2013/370953

*Research Article*

# Robust Indoor Sensor Localization Using Signatures of Received Signal Strength

**Yungho Leu,[1] Chi-Chung Lee,[2] and Jyun-Yu Chen[1]**

[1] *Department of Information Management, National Taiwan University of Science and Technology, Taipei 10607, Taiwan*
[2] *Department of Information Management, Chung Hua University, Hsinchu 30012, Taiwan*

Correspondence should be addressed to Yungho Leu; yhl@cs.ntust.edu.tw

Received 1 July 2013; Revised 20 October 2013; Accepted 22 October 2013

Academic Editor: Chang Wu Yu



Indoor localization based on the received signal strength (RSS) values of the wireless sensors has recently received a lot of attention. However, due to the interference of other wireless devices and human activities, the RSS value varies significantly over different times. This hinders exact location prediction using RSS values. In this paper, we propose three methods to counter the adverse effect of the RSS value variation on location prediction. First, we propose to use an index location to select the best radio map, among several preconstructed radio maps, for online location prediction. Second, for an observed value of the signal strength of a sensor, we record, respectively, the distances from the sensor to the nearest location and the farthest location where the signal strength value has been observed. The minimal and maximal (min-max) distances for each signal strength value of a sensor are then used to reduce the search space in online location prediction. Third, a location-dependent received signal strength vector, called the RSS signature, is used to predict the location of a user. We have built a system, called the region-point system, based on the proposed three methods. The experimental results show that the region-point system offers less mean position error compared to the existing methods, namely, RADAR, TREE, and CaDet. Furthermore, the index location method correctly selects the best radio map for online location prediction, and the min-max distance method promotes the prediction accuracy of RADAR by restricting the search space of RADAR in location prediction.

## 1. Introduction

Indoor localization is important for many real-life applications. For example, it gives the location context of a context-aware system that provides proper settings of the system based on the location, activity, and physiology of the user and the environmental context information [1]. Recently, indoor navigation applications, which require an exact indoor location, are becoming a very popular research area [2]. Due to the increasing need for indoor localization, many indoor localization techniques have been proposed. An indoor localization method can be categorized as a range-based or a range-free method [3]. While point-to-point distance information is required for a range-based method, it is not required for a range-free method. The techniques for estimating the distance between two communication nodes include the time of arrival (TOA) [4], time difference of arrival (TDOA) [5], and the angle of arrival (AOA) [6]. The TOA technique uses the radio signal propagation time to estimate the distance. The TDOA technique utilizes two radio signals with different propagation speeds and estimates the distance between the two communication nodes by measuring the difference between the arrival times of the two signals. Unlike TOA and TDOA, AOA technique measures the angle at which a signal arrives. It can be used to complement TDOA or TOA in location calculation [3]. Indoor localization methods that use range information usually achieve high accuracy in location estimation. For example, the *Cricket* [7] indoor localization system of MIT reported the error of 1 to 3 centimeters in position estimation. Despite being accurate in location prediction, the range-based localization techniques require large scale deployment and costly devices.

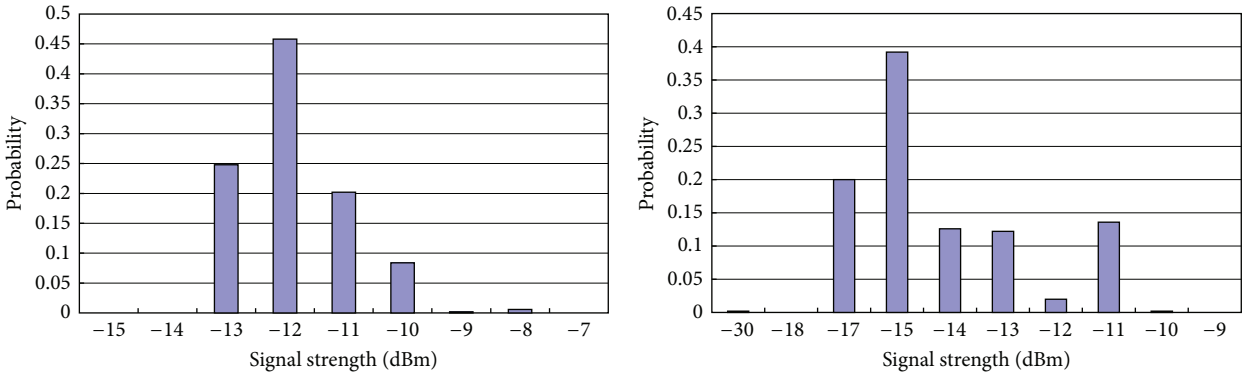
The range-free location prediction techniques have received a lot of attention recently. The well-known range-free location prediction methods include RADAR [8] and the probability-based methods [9–12]. RADAR is developed by Microsoft. In RADAR, for a predefined set of training locations, the received signal strength (RSS) values from several IEEE 802.11 access points are recorded in a database, called the radio map. To estimate the position of a user, the RSS values from the access points are collected at the location of the user. Afterwards, RADAR performs pattern matching of the collected RSS values against the RSS values in the radio map to find a fixed number of locations with the most similar RSS values against those of the user. Finally, the positions with the most similar RSS values are averaged to give the estimated position of the user. The probability-based methods also use the RSS values for location prediction. However, instead of a fixed number of locations for prediction, the probability-based methods use the Bayes theorem to predict the location of the user by finding the location where the collected RSS values of the user can be observed with the highest probability. In [13], the authors proposed to learn, at time $t_0$, a set of equations to fit the RSS values of a location using the RSS values of a set of reference points. With this method, the RSS value pattern of a specific location at a later time $t_j$ can be calculated by using the RSS value patterns of the reference locations at time $t_j$. Therefore, the effort to collect the RSS values at the offline training phase can be significantly reduced. However, in an environment where the RSS values observed at a location vary over times, the regression equations learned at time $t_0$ may not properly reflect the relationship between the RSS values of the location and those of the reference points. This may result in poor prediction accuracy. In [14], the authors proposed a method, called CaDet, which uses multiple decision trees for location prediction. They first divide the training dataset into several clusters and build a decision tree for each cluster. To predict the user's location, the RSS values of the user are compared against the means of the RSS values of each cluster center to find the cluster with the least distance from the RSS values of the user for prediction. Finally, the decision tree of the selected cluster is used to predict the location of the user. Besides using the values of the received signal strength, in [15], the authors proposed to use the link quality indicator (LQI) values for location prediction. They modeled the location prediction problem as a classification problem and used a neural network model to solve the problem. However, their method is more suitable for finding a coarse position for a user, such as in the kitchen or in the living room.

The most difficult problem for the range-free methods in location prediction is that the offline constructed radio map may not be suitable for online location prediction. The variation of the received signal strength values may outdate the radio map when an online location prediction is required. In this paper, we propose three methods to counter the adverse effect of the variation of the received signal strength values on location prediction. First, we propose to construct several radio maps over different nonoverlapping time intervals and use an index location to select the best radio map for online location prediction. Second, for an RSS value of a sensor observed in the location prediction area, we propose to record the minimal and the maximal (min-max) distance from the sensor to the locations where the same RSS value has been observed. The min-max distance information is used to reduce the number of locations required to be searched for in online location prediction. Thirdly, we propose to use a location-dependent received signal strength vector, called the RSS location signature, for pattern matching in online location prediction. A system, called the region-point system, which implemented the three proposed methods, has been implemented. The experimental results show that the region-point system offers less position prediction error compared to the existing methods, including RADAR, TREE, and CADet. Furthermore, the experiment also shows that the index location method correctly selects the best radio map for location prediction, and the min-max distance method significantly reduces the position prediction error of RADAR. The rest of this paper is organized as follows. In Section 2, we describe the phenomenon of the variation of the received signal strength values. In Section 3, we present the details of the region-point localization system. In Section 4, we present the experimental results. In Section 5, we give a discussion of the experimental result. Finally, in Section 6, we give the conclusion of this paper.

## 2. Variation of the Received Signal Strength

The most challenging problem for location prediction using RSS values is that the RSS values of a sensor observed at a fixed location change over different times [12–14, 16]. In this paper, we use the MPR2400CA sensor, a ZigBee-based sensor called Mote, to show the phenomenon of RSS value variation over different times. The Mote uses the RF frequency band of 2.4–2.4835 GHz for communication. The 2.4 GHz band frequency is a very noisy band since the wireless local area network (802.11b and 802.11g), the Bluetooth personal area network (802.15.1), and the industrial, scientific, and medical (ISM) devices are all using this unlicensed frequency band. The interference from other networks or devices forces the received signal strength value of a sensor at a fixed location to vary significantly over different times. Furthermore, the unpredictable people moving and door opening or closing cause the changes in the reflection, absorption, diffraction and scattering of the RSS values amplify the variation of the RSS values in an indoor environment [13].

To show the variation of RSS values over different times, we collected 500 RSS values from a fixed location which is 84.85 centimeters away from a ZigBee sensor for a time interval of 4 consecutive hours. Figure 1(a) shows the distribution of the RSS values from 10 a.m. to 2 p.m., while Figure 1(b) shows the distribution of the RSS values from 3 p.m. to 7 p.m. These figures show that not only the shapes of the distributions but also the averages of the signal strength values in different time intervals are different. The variation of the RSS values over different times implies that the RSS values collected at the offline training phase may not be good for online location prediction [13].




(a) From 10:00 a.m. to 2:00 p.m.

(b) From 3:00 p.m. to 7:00 p.m.

Figure 1: Distributions of the signal strength values.

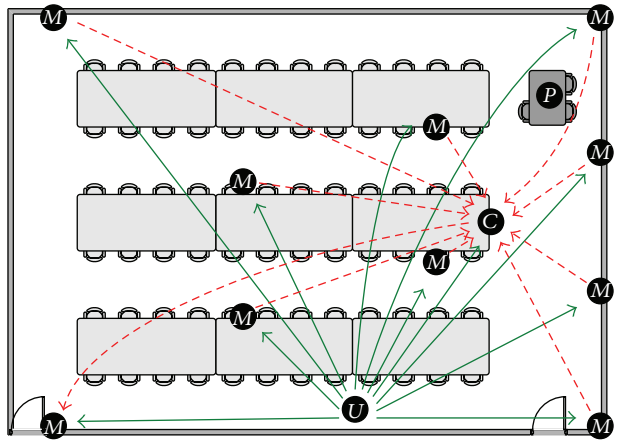



Figure 2: The components and layout of the system.

## 3. The Region-Point Location Prediction System

In this section, we present the implementation of a robust sensor prediction system which considers the variation of the RSS values.

*3.1. The Components and Layout of the System.* The components and the layout of the system are shown in Figure 2. The system is implemented in a classroom measuring 9.3 m × 13 m. There are three rows of tables with a desktop on each table. There are two doors and one electronic podium in the room. We placed ten Mote sensors, denoted by $M$ in Figure 2, as the reference sensors.

A sensor, denoted by $U$, is mounted on a moving cart for testing the location prediction algorithm. To predict the location of a user, the sensor $U$ (stands for the user) broadcasts a packet to the reference sensors. Upon receiving the packet from $U$, a reference sensor records the RSS value of its received packet, stores the RSS value in a new packet, and then sends the new packet to the location prediction computer, denoted by $C$ in Figure 2, to predict the location of $U$.

*3.2. Architecture of the System.* Figure 3 shows the architecture of the region-point location prediction system. It contains the offline training phase and the online location prediction phase. The offline training phase contains the following steps.

(1) For different time periods, collect the RSS values of the reference sensors for each training location and store the RSS values in the radio maps.

(2) Create a min-max distance table for each radio map.

(3) Find the index location for radio map selection.

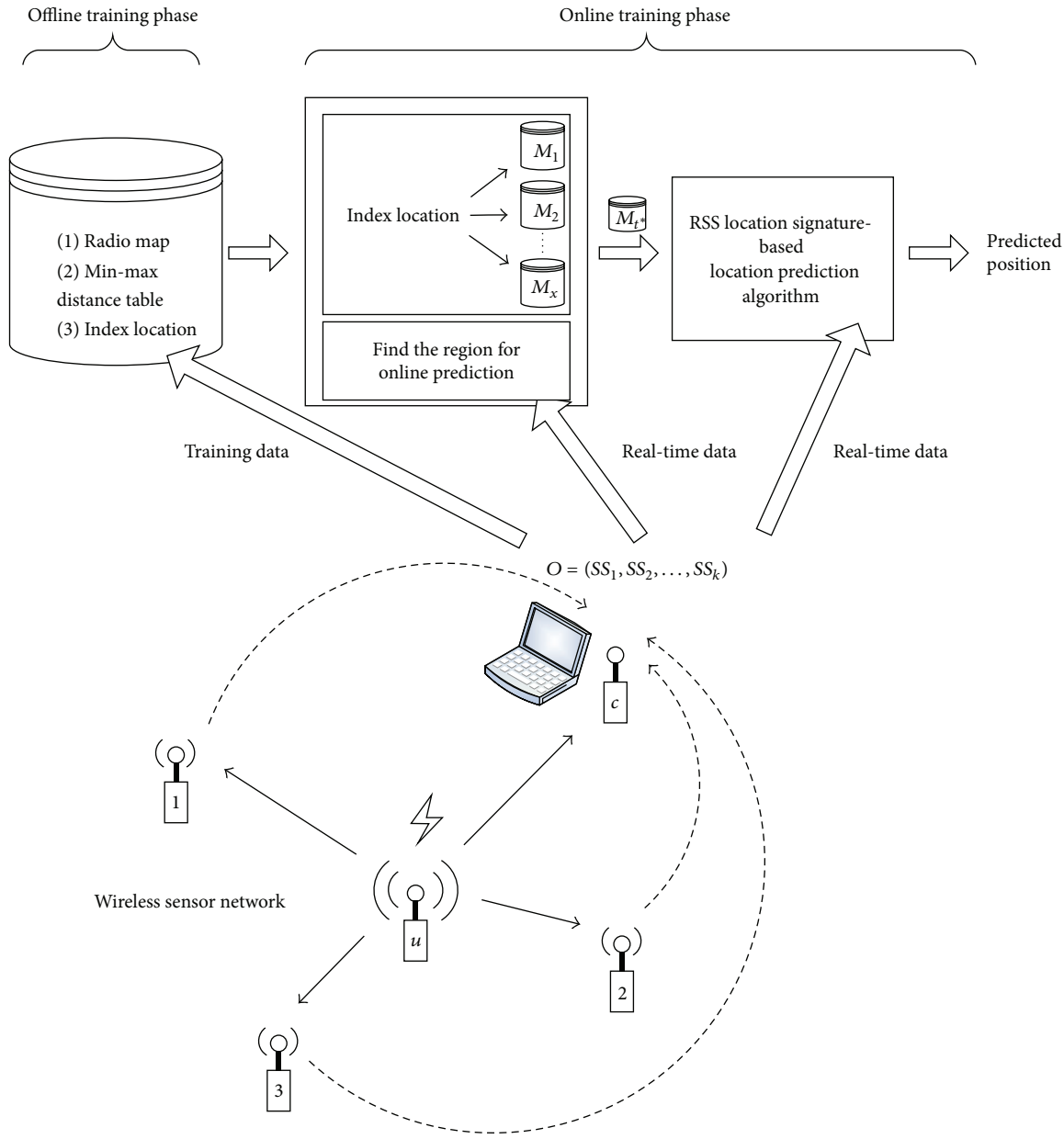



Figure 3: Architecture of the region-point location prediction system.

The online location prediction phase contains the following steps.

(1) Collect a number of RSS values at the index location.

(2) Select the best radio map for online location prediction.

(3) At the location that needs to be localized, collect the RSS values from the reference sensors; find the region for location prediction using the RSS values and the min-max distance table.

(4) Find the position of the predicted location in the selected region using the RSS signature of the collected RSS values.

The details of each step are discussed in the following.

*3.3. Radio Map Construction.* During the offline phase, we choose a number of different time intervals and construct a radio map for each time interval. A set of training locations denoted by $L = \{l_1, l_2, \ldots, l_n\}$ is chosen for collecting the RSS values. Each location $l_i$ is associated with a coordinate $(x_i, y_i)$. Assume that there are $k$ reference sensors, denoted by $R = \{m_1, m_2, \ldots, m_k\}$, where $m_i$ denotes sensor $i$. Then, each RSS value is stored in a vector $o_j$ of $k$ elements, denoted by $o_j = (ss_1, ss_2, \ldots, ss_k)$, where $ss_i$ is the RSS value of the packet received from reference sensor $i$. Table 1 shows an example of the radio map.

*3.4. The Min-Max Distance Table.* Due to the variation of the RSS values, a reference sensor may observe different RSS

Table 1: An example of the radio map of the system.

| locx | locy | $ss_1$ | $ss_2$ | $ss_3$ | $ss_4$ | $ss_5$ | $ss_6$ | $ss_7$ | $ss_8$ | $ss_9$ | $ss_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | −9 | −33 | −30 | −43 | −40 | −32 | −32 | −30 | −33 | −31 |
| 1 | 1 | −9 | −33 | −30 | −43 | −40 | −32 | −32 | −30 | −33 | −31 |
| 1 | 1 | −9 | −33 | −30 | −43 | −40 | −32 | −31 | −30 | −32 | −31 |
| 1 | 1 | −9 | −33 | −29 | −42 | −40 | −32 | −31 | −29 | −33 | −31 |
| 1 | 1 | −9 | −33 | −29 | −43 | −40 | −32 | −31 | −31 | −33 | −31 |
| 1 | 1 | −9 | −33 | −30 | −43 | −40 | −32 | −32 | −29 | −32 | −32 |
| 1 | 1 | −9 | −33 | −30 | −43 | −41 | −33 | −31 | −30 | −33 | −31 |
| 1 | 1 | −9 | −33 | −29 | −43 | −44 | −32 | −32 | −30 | −33 | −31 |
| 1 | 1 | −9 | −33 | −30 | −43 | −43 | −32 | −31 | −30 | −33 | −31 |
| 1 | 1 | −9 | −33 | −30 | −43 | −42 | −32 | −32 | −31 | −33 | −31 |
| 1 | 1 | −9 | −33 | −29 | −43 | −43 | −32 | −32 | −30 | −33 | −31 |
| 1 | 1 | −9 | −33 | −29 | −43 | −43 | −32 | −32 | −30 | −32 | −31 |
| 1 | 1 | −9 | −32 | −29 | −43 | −43 | −33 | −31 | −30 | −32 | −31 |
| 1 | 1 | −9 | −32 | −30 | −44 | −42 | −32 | −31 | −31 | −32 | −31 |
| 1 | 1 | −9 | −33 | −30 | −43 | −42 | −33 | −31 | −30 | −32 | −31 |
| 1 | 1 | −9 | −33 | −29 | −43 | −42 | −33 | −32 | −31 | −33 | −31 |
| 1 | 7 | −25 | −23 | −29 | −35 | −31 | −31 | −31 | −29 | −36 | −30 |
| 1 | 7 | −25 | −23 | −26 | −33 | −31 | −31 | −33 | −28 | −36 | −30 |
| 1 | 7 | −27 | −22 | −29 | −33 | −31 | −31 | −34 | −28 | −36 | −29 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

values from the localization sensor $U$ when $U$ is fixed at a specific location. Similarly, the same RSS value observed by a reference sensor may be from different packets transmitted by $U$ at different locations. For example, the RSS value −29 dbm of sensor $ss_8$ in Table 1 is observed when $U$ is located at location $(1, 1)$ and location $(1, 7)$. During the offline training phase, for each observed received signal strength value $ss_i$ of reference sensor $m_j$, we keep track of the minimum and the maximum distances from sensor $m_j$ to sensor $U$. Table 2 shows an example of the min-max distance table.

The min-max distance table is used to reduce the search region of locations during the online location prediction phase.

*3.5. The Index Location.* As noted in [13], the radio map constructed in the training phase may not be suitable for online location prediction. We propose to use several radio maps for location prediction. Assume that the set of time intervals is $T = \{t_1, t_2, \ldots, t_x\}$. Let $M_t$ denote the radio map constructed at time interval $t$, $t \in T$.

Let $\overline{o}_{t,i} = (\overline{ss}_{t,i,1}, \overline{ss}_{t,i,2}, \ldots, \overline{ss}_{t,i,k})$ denote the average RSS vector at location $l_i$ in $M_t$, where $\overline{ss}_{t,i,j}$, $1 \le j \le k$, is the average of the received signal strength values of sensor $j$ at location $l_i$. Then, for each location $l_i$, we calculate $D_i$, the summation of the Manhattan distances between every pair of average RSS vectors at location $l_i$, where each vector belongs to a different radio map. That is,

$$D_i = \sum_{t,t' \in T, t \neq t'} \sum_{m=1}^{k} \left| \overline{ss}_{t,i,m} - \overline{ss}_{t',i,m} \right|. \quad (1)$$

Table 2: The min-max distance table.

| Mote | $ss$ | Min dist. | Max dist. |
|---|---|---|---|
| 1 | −35 | 5.099019514 | 23.60084744 |
| 1 | −39 | 5.099019514 | 23.60084744 |
| 1 | −38 | 5.099019514 | 23.60084744 |
| 1 | −40 | 5.099019514 | 23.60084744 |
| 1 | −41 | 5.099019514 | 23.60084744 |
| 1 | −44 | 5.099019514 | 23.60084744 |
| 1 | −43 | 5.099019514 | 23.60084744 |
| 1 | −42 | 5.099019514 | 23.60084744 |
| 1 | −37 | 5.099019514 | 23.60084744 |
| 1 | −36 | 5.099019514 | 23.60084744 |
| 1 | −20 | 5.099019514 | 7.071067812 |
| 1 | −19 | 5.099019514 | 7.071067812 |
| ... | ... | ... | ... |

The index location $l_i$ is the location which maximizes $D_i$. That is, $D_i \ge D_j$, $j = 1, \ldots, n$.

During the online localization phase, we collect five received signal strength vectors at the index location. Take the average of the signal strength vectors, and then use the average RSS vector to select the best radio map for online location prediction. Assume that the average RSS vector is $\overline{o}_i = (\overline{ss}_{i,1}, \overline{ss}_{i,2}, \ldots, \overline{ss}_{i,k})$. Then, the radio map $M_{t^*}$ is found by using the following equation:

$$t^* = \arg\min_{t=1,\ldots,x} \sum_{m=1}^{k} \left| \overline{ss}_{t,i,m} - \overline{ss}_{i,m} \right|. \quad (2)$$

That is, we choose the radio map which minimizes the Manhattan distance against the online average RSS vector at location $l_i$ for online location prediction.

*3.6. The RSS Location Signature.* While the probability-based methods use the original radio map, as shown in Table 1, for location prediction, we propose to use a refined variant of the RSS vectors, called RSS signatures, for location prediction. An RSS signature of a location is a distinctive RSS representative for the location. Let $P(ss_{j,r} = k)$ denote the probability that the RSS value $k$ of sensor $r$ is observed at location $l_j$. Probability $P(ss_{j,r} = k)$ is defined in the following equation:

$$P\left(ss_{j,r} = k\right) = \frac{\text{fr}\left(ss_{j,r} = k\right)}{\sum_{z=1}^{n} \text{fr}\left(ss_{z,r} = k\right)}, \tag{3}$$

where $\text{fr}(ss_{j,r} = k)$ denotes the number of observations (frequency) of RSS value $k$ of sensor $r$ at location $l_j$. Note that, since the RSS value $k$ of sensor $r$ may be observed at different locations, $P(ss_{j,r} = k)$ is the location distribution of the RSS value $k$ of sensor $r$ at location $l_j$. We then define the *discernability factor* $f(ss_r = k)$ of an RSS value $k$ of sensor $r$ by the following equation:

$$f\left(ss_r = k\right) = 1 - \frac{(-1)}{\log n} * \sum_{j=1}^{n} p\left(ss_{j,r} = k\right) * \log\left(p\left(ss_{j,r} = k\right)\right). \tag{4}$$

The third and fourth terms of (4) together represent the entropy of location distribution of RSS value $k$ of sensor $r$ over different locations. The second term is used to normalize the entropy value to the interval $(0, 1)$. The maximal value of the entropy function occurs when value $k$ of sensor $r$ is evenly distributed over $n$ locations. In this case, value $k$ of sensor $r$ does not have any discernability to distinguish between different locations. The higher the skewness of the location distribution is, the smaller the normalized entropy is. The normalized entropy value equals zero if the RSS value $k$ of sensor $r$ can only be observed at a single location. Therefore, the discernability factor of an RSS value $k$ of a sensor $r$ is a measure of the ability to distinguish between different locations in the system. Note that $n$ in (4) is the number of locations in the system.

Having defined the discernability factor, we define the weight of an RSS value $k$ of sensor $r$ at location $l_i$ by the following equation:

$$w\left(ss_{i,r} = k\right) = \frac{\text{fr}\left(ss_{i,r} = k\right)}{m_i} * f\left(ss_r = k\right), \tag{5}$$

where $m_i$ is the total number of RSS samples, that is, the number of RSS vectors, collected at location $l_i$. Equation (5) shows that the weight of RSS value $k$ of sensor $r$ at location $l_i$ is the product of the discernability factor of RSS value $k$ and the probability of observing $k$ at location $l_i$.

For location $l_i$, we define its location signature at sensor $r$ to be the RSS value received from sensor $r$ whose weight is greater than that of any other RSS value received by $U$ at location $l_i$ from sensor $r$. To obtain the RSS location signature vector for location $l_i$, we find the RSS location signature value of each sensor $r$, $1 \le r \le k$. Table 3 shows an example of the table of RSS location signatures for the radio map in Table 1. Table 4 shows the weights of the corresponding RSS location signatures in Table 3.

*3.7. The Online Location Prediction Phase.* During the online localization phase, we first collect several RSS samples at the index location. Then, we compute the average RSS value vector of the collected samples and use it to select the best radio map for online location prediction.

To find the position of the user, we collect an RSS value vector, denoted by $O^* = (ss_1, ss_2, \ldots, ss_k)$, at the designated location of the user. Then, for each component $ss_i$ of vector $O^*$, we refer to the min-max distance table to find the minimum and the maximum distances from sensor $i$ for this signal strength value. Figure 4 shows the minimum and maximum distances from three sensors for an example.

From the circles with radii of minimum and maximum distances from their corresponding sensors, we can find the intersection points, that is, $P1, P2, P3, P4, P5, P6$, and $P7$, as shown in Figure 4. Then, we find the bounding box of the intersection points as the region within which the position (coordinates) of the user is to be found.

Finally, we find the training locations within the bounding box and use these locations to predict the position of the user. The pattern matching on RSS location signatures is used to find the position of the user. For each location $l_i$ in the bounding box, we find the *top-p* weighted RSS value components of its RSS location signature. Then, we compute the Euclidean distance between the vector of the *top-p* RSS value components of location $l_i$ and the vector of the corresponding components of $O^*$. Let us denote the *top-p* weighted RSS value components of the RSS location signature of $l_i$ by $V_i' = (f_1', f_2', \ldots, f_p')$ and the corresponding components of $O^*$ by $O' = (ss_1', ss_2', \ldots, ss_p')$. Then, the Euclidean distance between $V'$ and $O'$ is calculated according to the following equation:

$$\text{distance}\left(V', O'\right) = \sum_{z=1}^{p} \left(f_z' - ss_z'\right)^2. \tag{6}$$

After computing the distances between $O^*$ and the RSS location signatures of the training locations in the bounding box, the position of the user is predicted to be the position of the location with the smallest Euclidean distance of its top-p weighted RSS value components against $O'$.

## 4. Experiments

To show the performance of the region-point system, we perform several experiments on location prediction in the classroom. In this section, we present the experiments and the results.

Table 3: The RSS location signature table.

| locx | locy | $ss_1$ | $ss_2$ | $ss_3$ | $ss_4$ | $ss_5$ | $ss_6$ | $ss_7$ | $ss_8$ | $ss_9$ | $ss_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | −9 | −33 | −30 | −43 | −37 | −29 | −29 | −25 | −33 | −27 |
| 1 | 7 | −25 | −21 | −34 | −35 | −33 | −27 | −33 | −37 | −43 | −29 |
| 1 | 13 | −29 | −25 | −19 | −29 | −33 | −27 | −35 | −37 | −47 | −36 |
| 1 | 19 | −35 | −25 | −19 | −13 | −34 | −30 | −24 | −29 | −33 | −36 |
| 5 | 19 | −35 | −19 | −19 | −17 | −23 | −25 | −29 | −25 | −39 | −27 |
| 5 | 13 | −35 | −20 | −9 | −33 | −39 | −32 | −29 | −26 | −44 | −25 |
| 5 | 7 | −47 | −14 | −17 | −35 | −34 | −25 | −24 | −26 | −25 | −29 |
| 5 | 1 | −21 | −17 | −23 | −42 | −39 | −23 | −35 | −23 | −26 | −13 |
| 10 | 1 | −41 | −31 | −30 | −35 | −46 | −24 | −19 | −31 | −13 | −29 |
| 10 | 7 | −31 | −35 | −35 | −46 | −29 | −14 | −15 | −21 | −30 | −32 |
| 10 | 13 | −49 | −27 | −22 | −33 | −25 | −13 | −19 | −21 | −36 | −30 |
| 10 | 19 | −47 | −35 | −22 | −23 | −21 | −17 | −22 | −33 | −35 | −37 |
| 14 | 19 | −48 | −34 | −29 | −29 | −11 | −24 | −38 | −25 | −39 | −38 |
| 14 | 13 | −43 | −27 | −33 | −50 | −22 | −21 | −25 | −24 | −29 | −31 |
| 14 | 7 | −31 | −26 | −43 | −37 | −29 | −29 | −22 | −17 | −28 | −32 |
| 14 | 1 | −25 | −27 | −46 | −35 | −27 | −39 | −19 | −8 | −19 | −38 |

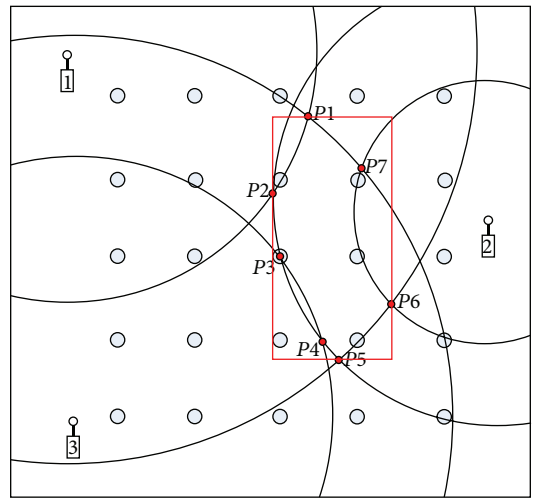



Figure 4: Min-max distance and bounding box.

*4.1. The Experimental Environment.* As shown in Figure 2, we implement the localization system in a classroom. Figure 5 shows the layout of the reference sensors and the locations where the training samples are taken. The ground of the classroom is decorated with tiles. The tile's dimension is 60 centimeters on each side. We set the origin of the coordinate system at the top left corner of Figure 5. Ten reference sensors, denoted by large circles in Figure 5, are evenly located in the classroom. The training locations are denoted by small circles. Totally, we have 16 training locations. The coordinates of two examples of training locations are $(1, 1)$ and $(1, 7)$. Note that, since each grid in Figure 5 represents one tile on the floor, the Euclidean distance between any two locations in Figure 5 can be calculated by multiplying their Euclidean distance by 0.6 meters. To build the radio maps, we collect 500 RSS value samples from each of the 16 training locations over a consecutive 4-hour time interval of the day. Three radio maps, denoted by $M_1$, $M_2$, and $M_3$, are constructed for the experiment.

For comparison purpose, we implement the RADAR method and a decision tree method called TREE and the CaDet method. For RADAR, the RSS vectors of different training samples from the same location are averaged. As a result, each location is associated with only one average RSS vector. To predict the coordinates of a test sample, three neighbors whose RSS vectors are among the top 3 shortest distances from the test sample are retrieved from the radio map and their corresponding coordinates are averaged to give the predicted coordinates of the test sample. To examine the effect of the search space reduction on RADAR, we revised

Table 4: The weight table.

| locx | locy | $ss_1$ | $ss_2$ | $ss_3$ | $ss_4$ | $ss_5$ | $ss_6$ | $ss_7$ | $ss_8$ | $ss_9$ | $ss_{10}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0.3640 | 0.3016 | 0.0736 | 0.1075 | 0.0913 | 0.1034 | 0.1029 | 0.0906 | 0.1324 | 0.0766 |
| 1 | 7 | 0.1376 | 0.1832 | 0.0733 | 0.1479 | 0.0987 | 0.0535 | 0.0709 | 0.0947 | 0.0822 | 0.1346 |
| 1 | 13 | 0.0862 | 0.0697 | 0.1518 | 0.1566 | 0.0587 | 0.0946 | 0.0864 | 0.0782 | 0.0708 | 0.1242 |
| 1 | 19 | 0.1026 | 0.0782 | 0.1773 | 0.2940 | 0.0703 | 0.1022 | 0.0953 | 0.1014 | 0.1317 | 0.1576 |
| 5 | 19 | 0.1823 | 0.2862 | 0.2723 | 0.4140 | 0.1325 | 0.1147 | 0.0686 | 0.3700 | 0.0813 | 0.2053 |
| 5 | 13 | 0.1141 | 0.1504 | 0.2700 | 0.1944 | 0.0851 | 0.1423 | 0.0668 | 0.0895 | 0.0941 | 0.2057 |
| 5 | 7 | 0.0630 | 0.5700 | 0.0965 | 0.0986 | 0.1421 | 0.1159 | 0.0823 | 0.1278 | 0.4695 | 0.0990 |
| 5 | 1 | 0.7471 | 0.1425 | 0.3014 | 0.0724 | 0.0779 | 0.1030 | 0.0803 | 0.2471 | 0.3600 | 0.3980 |
| 10 | 1 | 0.0646 | 0.0864 | 0.0895 | 0.1881 | 0.0939 | 0.2180 | 0.1038 | 0.0979 | 0.3600 | 0.1663 |
| 10 | 7 | 0.2262 | 0.0787 | 0.0903 | 0.0630 | 0.1668 | 0.1865 | 0.3520 | 0.3476 | 0.1817 | 0.1327 |
| 10 | 13 | 0.0521 | 0.1717 | 0.0967 | 0.0934 | 0.0992 | 0.1764 | 0.2365 | 0.5402 | 0.1206 | 0.1699 |
| 10 | 19 | 0.1132 | 0.1129 | 0.1063 | 0.2117 | 0.3700 | 0.6320 | 0.1667 | 0.1122 | 0.1170 | 0.1287 |
| 14 | 19 | 0.1255 | 0.1264 | 0.0743 | 0.4479 | 0.3220 | 0.2139 | 0.0947 | 0.2848 | 0.0813 | 0.1493 |
| 14 | 13 | 0.0597 | 0.1727 | 0.0806 | 0.1265 | 0.1485 | 0.2720 | 0.1916 | 0.1437 | 0.0728 | 0.1105 |
| 14 | 7 | 0.0929 | 0.0722 | 0.0825 | 0.1094 | 0.3351 | 0.1344 | 0.0863 | 0.2180 | 0.2163 | 0.1156 |
| 14 | 1 | 0.1480 | 0.2418 | 0.0610 | 0.1380 | 0.0970 | 0.0755 | 0.2427 | 0.2360 | 0.2991 | 0.1104 |

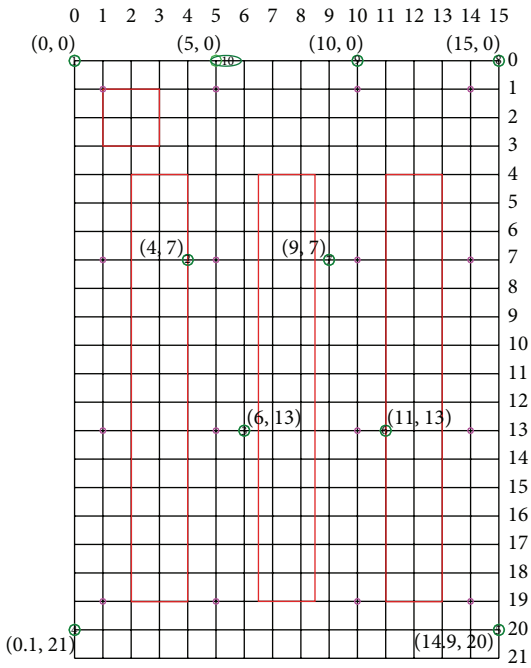



Figure 5: The positions of the sensors and the training locations.

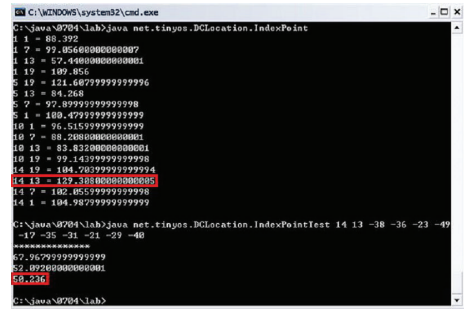



Figure 6: An execution of the radio map selection algorithm.

the RADAR method by using the min-max distance table to confine the search region of RADAR. We call the revised RADAR method ReRADAR in the experiment.

For the TREE method, a decision tree is constructed for every radio map. The decision tree is then used to predict the coordinates of a test sample. Note that we use the CART decision tree model in R [17] to construct the decision trees. For CaDet method [14], we first use the K-means method in R to divide the training samples into three clusters based on their RSS vectors. A CART decision tree is then built for each cluster. To predict the coordinates a test sample, we compare the RSS vector of the test sample against the cluster mean of each cluster and select the decision tree whose corresponding cluster center has the shortest distance against the test sample to predict the location of the test sample.

*4.2. The Experimental Results.* Figure 6 shows an execution of the radio map selection algorithm. It shows that the location $(14, 13)$ is chosen as the index location since it has the largest variance on the RSS values of different radio maps. Furthermore, based on the index location, radio map $M_3$ is selected as the best radio map for the ongoing experiment.

To conduct the experiment, we consecutively collect 20 RSS value samples at each of the 16 testing locations. Totally, there are 320 test samples. Figure 7 shows the four executions of the region point with different lengths of the RSS location signatures.

Figure 7 shows that the longer the RSS location signature, the higher the prediction accuracy. However, the length effect decreases as the length becomes longer. This is evidenced in Figure 7, where the accumulated errors for region point with 9 and 10 components, respectively, are almost the same.

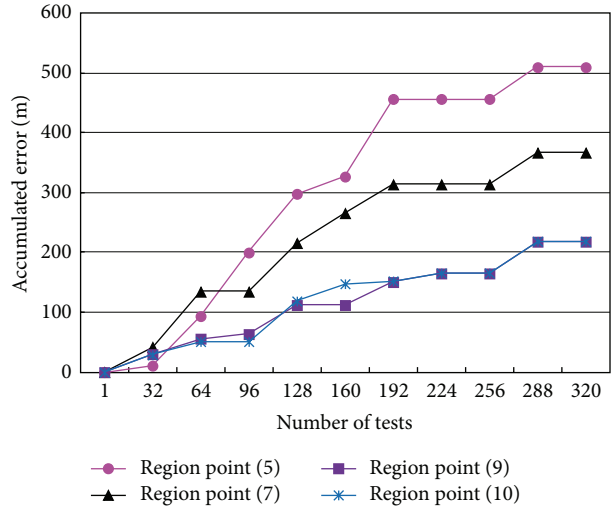



Figure 7: Performance of region point with different lengths of the RSS location signature.

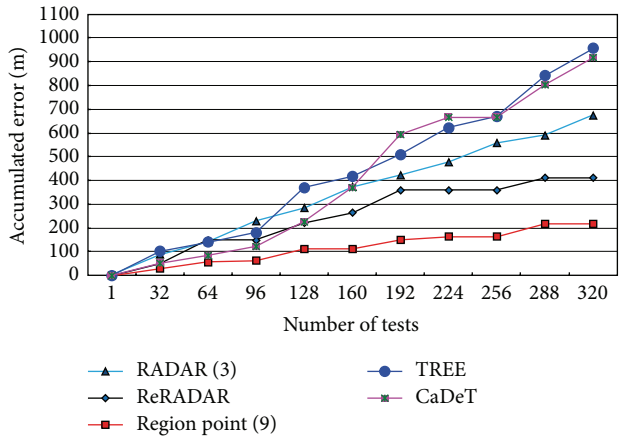



Figure 8: Accumulated errors for different methods based on radio map $M_1$.

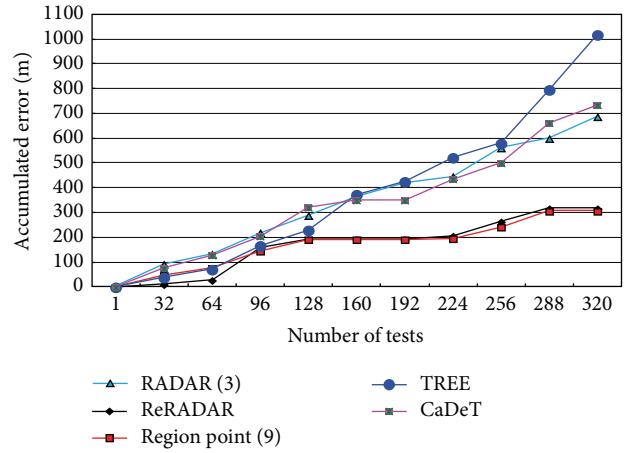



Figure 9: Accumulated errors for different methods based on radio map $M_2$.

Figure 8 shows the accumulated errors for RADAR, ReRADAR, TREE, CaDet, and region point. It shows that the region point method has the smallest accumulated error compared with RADAR, ReRADAR, TREE, and CaDet. It also shows that the accumulated error for ReRADAR is much less than that of RADAR. The mean errors for the 320 test samples are 0.681, 1.29, 2.11, 2.87, and 2.99 for region point, ReRADAR, RADAR, CaDet, and TREE, respectively. Figure 8 shows the fact that the search region restriction using the min-max distance table effectively reduces the prediction error of RADAR.

Figure 9 shows the accumulated errors for different methods based on radio map $M_2$. It again shows that the accumulated error for ReRADAR is much less than that of RADAR. The mean errors are 0.958, 0.991, 2.15, 2.29, and 3.18 meters for region point, ReRADAR, RADAR, CaDet, and TREE, respectively.

Figure 10 shows the accumulated errors based on radio map $M_3$ which is chosen by the index location. The mean errors are 0.556, 0.822, 1.98, 1.18, and 1.30 meters for region point, ReRADAR, RADAR, CaDet, and TREE, respectively. Note that the errors for different methods based on $M_3$ are all less than their corresponding errors in radio map $M_1$ and $M_2$, respectively. This shows that the index location method correctly selects the best radio map for online prediction. It is also noted, from Figures 8, 9, and 10, that, although clustering before constructing decision trees helps to promote the prediction accuracy of CaDet, the improvement is not significant.

## 5. Discussion

The fact that the TREE and CaDet methods do not perform well in our experimental environment needs to be carefully studied. To do so we show the decision tree built by CART based on radio map $M_3$ in Figure 11. Note that $M_3$ contains 8000 samples with 500 samples for each of the 16 locations.

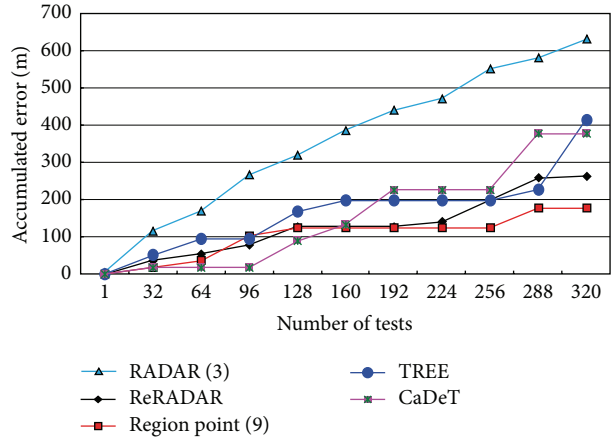



Figure 10: Accumulated errors for different methods based on radio map $M_3$.

Table 5: Confusion table for testing samples.

| | 1_1 | 1_13 | 1_19 | 1_7 | 10_1 | 10_13 | 10_19 | 10_7 | 14_1 | 14_13 | 14_19 | 14_7 | 5_1 | 5_13 | 5_19 | 5_7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1_1 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1_13 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1_19 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1_7 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 14 | 0 | 0 |
| 10_1 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10_13 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10_19 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10_7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14_1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14_13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14_19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 |
| 14_7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5_1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 | 0 | 0 |
| 5_13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 0 | 0 | 12 | 0 | 0 |
| 5_19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 0 |
| 5_7 | 0 | 0 | 0 | 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 |

The label at the terminal node denotes the class and the number of samples in the training dataset that are classified as this label. For example, the terminal node 1 has label 1_13, which denotes location (1, 13), and there are 173 samples being classified as location (1, 13). The classification accuracy for the training dataset of this decision tree is 93.3 percent. Table 5 shows the confusion table for predicting the 320 testing samples based on the decision tree of Figure 11. It shows that 234 out of 320 samples are correctly classified, that is, correctly predicted. For comparison, we show the histogram of prediction errors for both TREE and region point in Figure 12. It shows that the region point has more samples correctly classified than the TREE does, that is, 283 versus 234. Furthermore, for the misclassified samples, the region point tends to classify them to their nearby locations. These two observations account for a less mean prediction error in region point than those of the TREE and CaDet.

## 6. Conclusions

In this paper, we present the implementation of a robust indoor localization system using a wireless sensor network. In this system, we propose three methods to counter the adverse effect of variation on the received signal strength values on location prediction. First, we propose to use an index location to select the best radio map for online location prediction. Second, we propose to use the min-max distance table to confine the search region for online location prediction. Finally, we propose to use the RSS location signature for pattern matching in online location prediction. The experimental results showed that the index location method correctly selects the best radio map for online location prediction. It also showed that the min-max distance table method effectively reduces the prediction error of RADAR, and the region point system offers a higher

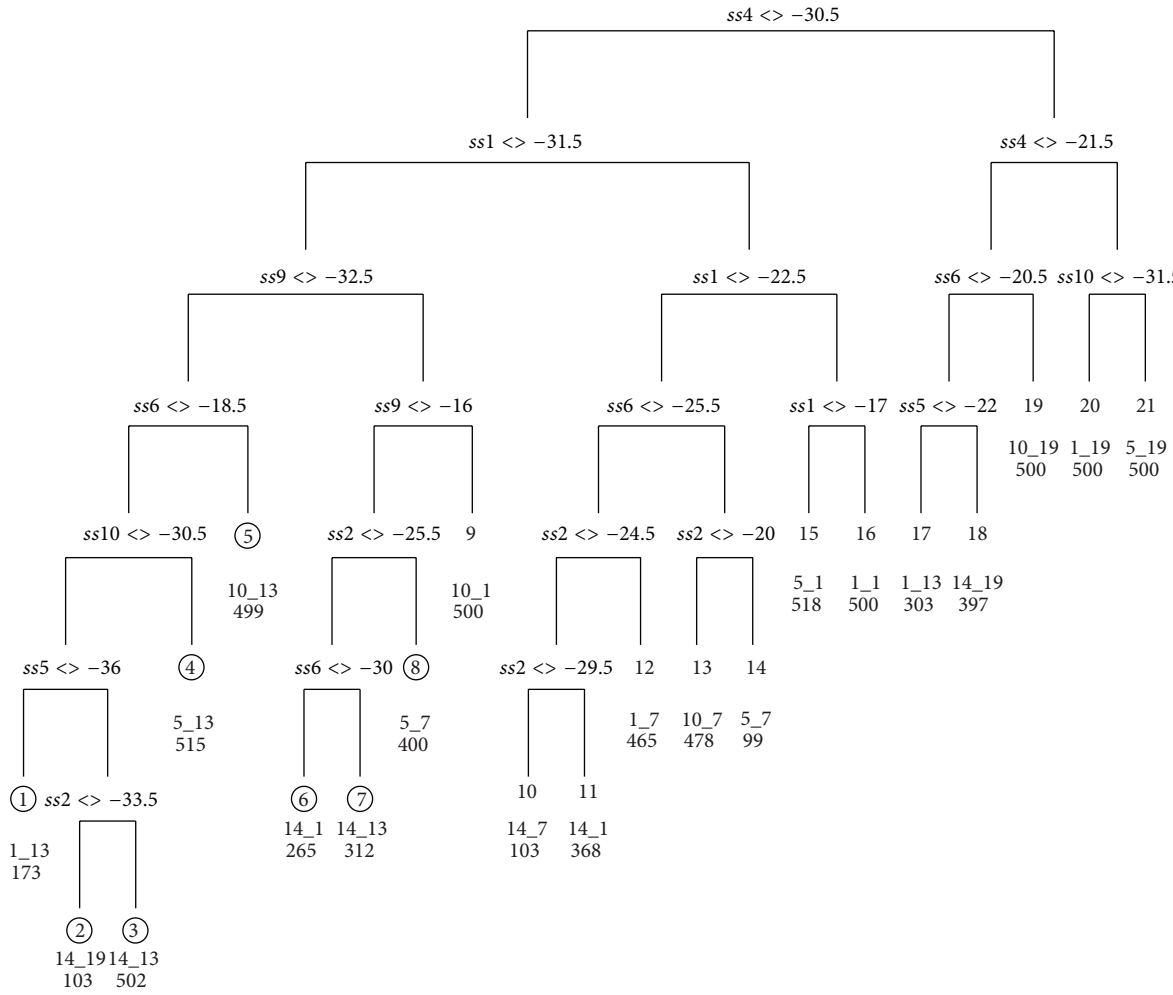



Figure 11: The CART decision tree for radio map $M_3$.

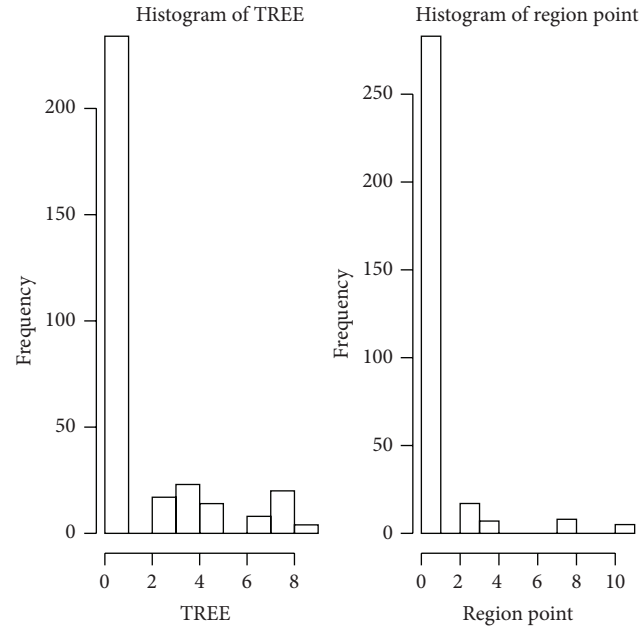



Figure 12: Histograms of prediction errors for TREE and region point.

prediction accuracy than those of the RADAR, TREE, and CaDet.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References


[1] A. Krause, A. Smailagic, and D. P. Siewiorek, "Context-aware mobile computing: learning context-dependent personal preferences from a wearable sensor array," *IEEE Transactions on Mobile Computing*, vol. 5, no. 2, pp. 113–127, 2006.

[2] A. W. S. Au, C. Feng, S. Valaee et al., "Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 2050–2062, 2012.

[3] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 81–95, San Diego, Calif, USA, September 2003.

[4] S. Capkun, M. Hamdi, and J. Hubaux, "GPS-free positioning in mobile ad-hoc networks," in *Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS '01)*, pp. 3481–3490, January 2001.

[5] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM '01)*, pp. 166–179, Rome, Italy, July 2001.

[6] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) using AoA," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, pp. 1734–1743, San Francisco, Calif, USA, March-April 2003.

[7] 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, Boston, Mass, USA, August 2000.

[8] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, pp. 775–784, March 2000.

[9] X. Chai and Q. Yang, "Reducing the calibration effort for probabilistic indoor location estimation," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 649–662, 2007.

[10] M. A. Youssef, A. Agrawala, and A. U. Shankar, "WLAN location determination via clustering and probability distributions," in *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom '03)*, pp. 143–150, March 2003.

[11] Q. Yao, F.-Y. Wang, H. Gao, K. Wang, and H. Zhao, "Location estimation in Zigbee network based on fingerprinting," in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES '07)*, December 2007.

[12] M. Youssef and A. Agrawala, "The horus WLAN location determination system," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pp. 205–218, June 2005.

[13] J. Yin, Q. Yang, and L. M. Ni, "Learning adaptive temporal radio maps for signal-strength-based location estimation," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 869–883, 2008.

[14] Y. Chen, Q. Yan, J. Yin, and X. Chai, "Power-efficient access-point selection for indoor location estimation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 7, pp. 877–888, 2006.

[15] Y.-G. Ha, A.-C. Eun, and Y.-C. Byun, "Efficient sensor localization for indoor environments using classification of link quality patterns," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 701259, 6 pages, 2013.

[16] S.-P. Kuo, B.-J. Wu, W.-C. Peng, and Y.-C. Tseng, "Cluster-enhanced techniques for pattern-matching localization systems," in *Proceedings of the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS '07)*, October 2007.

[17] "The R Project for Statistical Computing," http://www.r-project.org/.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 481570, 15 pages
http://dx.doi.org/10.1155/2013/481570

*Research Article*

# Towards Robust Routing in Three-Dimensional Underwater Wireless Sensor Networks

**Ming Xu,[1,2] Guangzhong Liu,[1] Huafeng Wu,[3] and Wei Sun[1]**

[1] *College of Information Engineering, Shanghai Maritime University, 1550 Haigang Avenue, Shanghai 201306, China*
[2] *Shanghai Key Lab of Intelligent Information Processing, Fudan University, 220 Handan Road, Shanghai 200433, China*
[3] *Merchant Marine College, Shanghai Maritime University, 1550 Haigang Avenue, Shanghai 201306, China*

Correspondence should be addressed to Ming Xu; mingxu@shmtu.edu.cn

Received 20 June 2013; Revised 26 September 2013; Accepted 1 October 2013

Academic Editor: Chang Wu Yu



The areas of three-dimensional (3D) underwater wireless sensor networks (UWSNs) have attracted significant attention recently due to their applications in detecting and observing phenomena that cannot be adequately observed by means of two-dimensional UWSNs. However, designing routing protocols for UWSNs is a challenging task because path breakage occurs frequently due to uncertain node and link failures. In this paper, we present a Robust Routing Protocol (RRP) that aims to achieve robustness under harsh underwater conditions of 3D UWSNs. In RRP, each sensor has a data structure called backup bin, which allows to construct main backup links and auxiliary backup links for repairing failed links and ensuring normal data delivery. Moreover, sensor nodes can also enlarge their transmission range to build new links when the main routing path is not reachable through backup links. Simulation results show that the proposed protocol can reduce node and link failures' impact on metrics of packet delivery ratio, network throughput, energy consumption, and average end-to-end delay.

## 1. Introduction

Underwater wireless sensor networks (UWSNs) have a lot of potential application areas such as oceanographic data collection, disaster prevention, pollution monitoring, offshore exploration, and military surveillance [1–3]. Radio frequency (RF) signals suffer from severe attenuation in water and have been successfully deployed only at very low frequencies, involving large antenna and high transmission power. Hence, acoustic signals have been used for wireless communication in current underwater physical layer, which has challenges to be overcome such as long propagation delay resulting from low speed of sound propagation, severely limited bandwidth, and time-varying multipath propagation [4, 5]. All the above distinct features of UWSNs give birth to new challenges areas for every level of the network protocol suite. UWSNs consist of a variable number of sensors and vehicles that are deployed to perform collaborative monitoring tasks over a given area. To achieve this objective, sensors and vehicles self-organize in an autonomous network that can adapt to the characteristics of the underwater environment.

Three-dimensional (3D) UWSNs are used to detect and observe phenomena that cannot be adequately observed by means of ocean bottom underwater sensor nodes, that is, to perform cooperative sampling of the 3D ocean environment [6–8]. In 3D UWSNs, sensors float at different depths to observe a given phenomenon. Many problems arise with 3D UWSNs that need to be solved in order to enable underwater monitoring in the new environment. Among them, providing efficient routing is a very challenging task due to the unique characteristics of 3D UWSNs. 3D UWSNs must rely on underwater acoustic communications because high-frequency radio signals used in traditional ground-based sensor networks can be rapidly absorbed by water. Therefore, many research results in land-based sensor networks as well as traditional ad hoc networks cannot be applied to 3D UWSNs directly, which requires new routing protocol to be designed for the new features of the 3D UWSNs in order to ensure that the performance can meet the actual underwater environmental needs.

According to their architectures, the routing protocols of 3D UWSNs can be divided into three categories:

location-based routing, flat routing, and hierarchical routing [9, 10]. Location-based routing has good scalability, but it requires a positioning system or positioning algorithm to help the nodes to calculate the location information. Flat routing protocols have better robustness, but the excessive overhead for maintaining routing information restricts their application to small-scale underwater acoustic circumstances. Hierarchical routing also has good scalability, but the cluster maintenance overhead and the failure of key nodes will affect the routing efficiency.

In 3D UWSNs, communication links face high bit error rate, temporary losses, or even permanent failure. Moreover, the impact of node failure is more severe than that of link failure because even a single node failure is logically equivalent to several concurrent link failures. In this paper, we propose a Robust Routing Protocol (RRP) that addresses some of the major requirements imposed by 3D UWSNs including high robustness combined with energy-efficient communication. Since 3D UWSNs are made up of expensive sensor nodes and they operate in harsh underwater environments, the likely possibility of node or link failures must be considered. In RRP, each sensor has a data structure called backup bin, which has been utilized to construct main backup links and auxiliary backup links together with routing table for repairing routing path and ensuring normal data delivery. As geographic routing protocol, RRP does not require state information on the sensor nodes and only a small fraction of the nodes are involved in routing to ensure robust operation for surveillance and monitoring applications.

The remainder of the paper is organized as follows. Section 2 presents a brief overview of related work, while Section 3 introduces the technical details of our routing protocol. Performance evaluation is described in Section 4. Finally, we conclude the paper in Section 5.

## 2. Related Work

The underwater environment introduces difficulties in designing efficient routing protocols not experienced terrestrially, such as transmission loss due to geometric spreading and absorption by the ocean [11, 12]. Tan et al. [13] proposed a new protocol based on hop-by-hop hybrid implicit/explicit acknowledgment scheme which is proposed for a multihop UWSN. In this protocol, data packets forwarded by downstream nodes can work as implicit ACKs for previous transmitted data packets. Vahdat and Becker [14] proposed Epidemic Routing (ER) protocol where each node replicates a packet to every encountered node. ER can utilize every opportunity to deliver a packet to the destination and maximize successful delivery ratio and minimize average end-to-end delay in unconstrained networks. However, this routing protocol consumes too many resources that make it not desirable in resource-constrained networks such as UWSNs. Pompili et al. [15] introduced two distributed routing algorithms for delay-insensitive and delay-sensitive applications, respectively, with the objective of minimizing the energy consumption taking the varying condition of the underwater channel and the different application requirements into account.

Vector-based forwarding (VBF) [16] is a geographic approach, which allows the nodes to weigh the benefit to forward packets and reduce energy consumption by discarding low benefit packets. Therefore, over a multihop path, only the nodes that are located within a pipe of given width between the source and the destination are considered for relaying. However, in the areas of low density of nodes, VBF may not find the path close to the routing vector. Similarly, Jornet et al. proposed Focused-Beam Routing (FBR) [17] protocol that is suitable for networks containing both static and mobile nodes. The objective of FBR is to determine which nodes are candidates for relaying. Candidate nodes are those that lie within a cone of angle $\pm\theta/2$ emanating from the transmitter towards the final destination. An RTS/CTS handshake is set up to isolate closer nodes within this cone. If a node determines that it is within the transmitter's cone, it will respond to the RTS. Those nodes that are outside the cone will not respond. A theoretical argument supporting geographic routing has been discussed in [18] based on simple propagation and energy consumption models for underwater networks. The study shows that an optimal number of hops along a path exist and that increasing the number of hops by choosing closer relays is preferable with respect to keeping the route shorter. In view of this, several position-based routing algorithms are proposed and compared; results show that selecting relays closer than a given maximum distance before seeking farther ones achieves in fact optimal energy consumption.

Depth-Based Routing (DBR) [19] can handle network dynamics efficiently without the assistance of a localization service. DBR forwards data packets greedily towards the water surface (i.e., the plane of data sinks). In DBR, a data packet has a field that records the depth information of its recent forwarder and is updated at every hop. But DBR has only greedy forwarding mode, which alone is not able to achieve high delivery ratios in sparse areas. A similar idea can be found in [20], a hydraulic pressure-based anycast routing protocol that exploits the measured pressure levels to route data to surface buoys where relays are chosen based on a weighted average of advancement towards the sources and probability of packet delivery, and an efficient underwater dead end recovery method is added to handle the absence of a relay node at lower depth than the current packet holder. In order to remove the constraints imposed by special hardware (like every node should be equipped with depth or pressure sensor, which not only increase the cost of the network but also become a burden for extra energy consumption), Ayaz and Abdullah [21] proposed a Hop-by-Hop Dynamic Addressing-Based (H2-DAB) routing protocol to provide scalable and time-efficient routing for UWSN. The H2-DAB routing protocol does not require any dimensional location information or any extra specialized hardware compared with many other routing protocols in the same area. However, the problem of multihop routing still exists as it is based on multihop architecture, where nodes near the sinks drain more energy because they are used more frequently. Noh et al. [22] proposed a Void-Aware Pressure Routing (VAPR) protocol that uses sequence number, hop count, and depth information embedded in periodic beacons

to set up next-hop direction and to build a directional trail to the closest sink. VAPR employs periodical beacon messages to identify the direction of each node in a heuristic manner. The direction is set as *up* when a beacon is received from a shallower depth node; otherwise, it is set as *down*. When multiple direction cues are received from different sinks, the direction cue with the minimal hop count is chosen. Also, a node's next-hop data forwarding direction is set based on the beacon sender's data forwarding direction. VAPR employs a multisink structure to prevent a rapid battery drain in the nodes closer to the sinks. Moreover, it can handle the void problem with a heuristic method. However, each node in VAPR requires the knowledge of 2-hop connectivity and neighboring nodes' pairwise distances. In addition, the enhanced beaconing component should be repeated in a short interval (usually 50 seconds at a time), which causes a significant increase in the network overhead.

Packet redundancy and multiple paths can be exploited in order to increase the reliability of UWSNs. Ayaz et al. [23] provided a two-hop acknowledgment reliability model in order to insure the reliable data deliveries to the surface sinks, where two copies of the same data packet are maintained in the network without extra burden on the available resources. A relay node that has data packets to forward will not reply the acknowledgment until it cannot find the next hop towards the destination. But if a node is unable to find the next hop due to any failure, or even if it is lost, then packets in the buffer are not considered lost. All the nodes that send the data packets towards this node will wait for a certain amount of time before trying again for the next hop. Xu et al. [24] proposed a Multiple-Path Forward Error Correction (M-FEC) approach that integrated multiple-path communications and Hamming Coding to eliminate retransmission and enhance reliability in underwater sensor networks. Moreover, a Markov model and a dynamical decision and feedback scheme were developed to decrease the number of the paths in order to save energy and ensure the desirable packet error rate. However, M-FEC may cause much long delay because of additional process of encoding and decoding the data packets, and it does not tackle the issues of node and link failures.

## 3. Proposed Scheme

*3.1. Network Model and Definitions.* We consider that 3D UWSNs are composed of a certain number of sensor nodes uniformly scattered in monitoring fields. We present a generic model for a 3D UWSN that is represented by $G = (V, E)$ with $n$ sensor nodes. Every sensor node has the same transmission range and is assigned with a triplet of coordinates $(x, y, z)$, where each coordinate represents the hop distance of the node from one anchor. We also assume that all sensor nodes know their own locations through a certain localization service [25]. Such assumption is justified in underwater systems where fixed bottom-mounted nodes have location information upon deployment. In fact, the underwater localization is a nontrivial task for which relatively very few options are available. For example, employing global positioning system (GPS) does not work well because radio frequency (RF) waves are heavily attenuated in underwater environment. Many researchers have proposed a variety of localization schemes and techniques to address this issue specially [26, 27]. It is not always feasible to deploy anchor nodes at the sea floor for deep water environment. In this case, mobile beacon nodes such as autonomous underwater vehicles (AUVs), which are equipped with internal navigation systems, are exploited as reference nodes to assist in corresponding distributed localization algorithms. This paper takes advantage of these research results as existing preconditions. Moreover, we also consider symmetric links; that is, for any two nodes $s_u$ and $s_v$, $s_u$ reaches $s_v$ if and only if $s_v$ reaches $s_u$. Each node can either transmit or receive data packet.

*Definition 1.* The function $\delta(u, v)$ defines the distance between two nodes $s_u$ and $s_v$ in a 3D Euclidean space as

$$\delta : N \times N \longrightarrow \Gamma : \delta(u, v),$$
$$\delta(u, v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2 + (u_z - v_z)^2}. \quad (1)$$

All sensor nodes have two transmission ranges $r_{\text{norm}}$ (normal transmission range) and $r_{\max}$ (maximal transmission range). The normal transmission range is used for regular routing process with less energy usage, while the maximal transmission range is used to counteract the negative influence of node and link failures with more energy consumption when necessary. Two nodes $s_u$ and $s_v$ are neighbors and connected by a link if $\delta(u, v) < r_{\text{norm}}$. When the distance of two sensor nodes is equal or shorter than $r_{\text{norm}}$, the receiving rate is 1. When the distance is larger than $r_{\text{norm}}$ but still within the range of $r_{\max}$, then the receiving rate is between 0 and 1. Generally, the receiving rate of nodes $s_u$ and $s_v$ is calculated as

$$\eta(u, v) = \begin{cases} 1, & \delta(u, v) \le r_{\text{norm}}, \\ \dfrac{r_{\max} - \delta(u, v)}{r_{\max} - r_{\text{norm}}}, & r_{\text{norm}} < \delta(u, v) < r_{\max}, \\ 0, & \delta(u, v) \ge r_{\max}. \end{cases} \quad (2)$$

Note that even at a short distance from the transmitter to the receiver, the receiving rate is still lower than 1. The reason is that underwater acoustic channels are affected by many factors such as path loss, noise, multipath fading, and Doppler spread. All these factors cause a high bit error rate in underwater environment. For simplicity, the receiving rate in the normal transmission range is assumed to be 1, which will not affect the performance comparison of different routing protocols. In other cases, the receiving rate is calculated according to formula (2).

Consider two sensor nodes at minimum hop distance $h$; there exist two values $u(h)$ and $v(h)$ such that the Euclidean distance $\delta(u, v)$ between the two nodes is bounded; that is, $u(h) \le \delta(u, v) \le v(h)$. The quality of the bounds depends on the network density $\rho$. In particular for each $h > 0$ holds

$$\lim_{\rho \to \infty} v(h) - u(h) = r_{\text{norm}}, \quad (3)$$

where $r_{\text{norm}}$ is the normal transmission range of the sensor nodes.

Sensing devices generally have widely different theoretical and physical characteristics. Thus, numerous models of varying complexity can be constructed based on application needs and device features. However, for most kinds of sensors, the sensing ability diminishes as distance increases.

*Definition 2.* Given a source sensor node $s_u$ and a destination sensor node $s_v$, then the sensor model is defined as

$$S(u,v) = \frac{\lambda}{\delta(u,v)^k + \xi}, \tag{4}$$

with $\lambda > 0$ and $\xi > 0$, where $\lambda$ is the signal amplitude, $k$ is the sensor technology-dependent parameter, and $\xi$ is a predefined parameter for handling the situation when the source node and the destination node locate at the same position.

Underwater wireless sensor nodes have been equipped with sensing devices. They collect data from the external environment and transmit these data by one or multihop to the sink node. Sink node is the node that generates data aggregation results and also the target location of the data transmission. A routing path $p$ that consists of $m$ sensor nodes can be expressed as $p = \{s_1, s_2, \ldots, s_m\}$, where $(s_i, s_{i+1}) \in E$, $i \in [1, \ldots, m-1]$, and the length of path $p$ is $L(p)$. If there exist $n$ paths from the source node to the sink node, then these paths can be expressed as $\chi = \{p_1, p_2, \ldots, p_n\}$ with length $L(\chi) = \sum L(p_i)$.

In order to build nodes' state model, we use a random variable $X(u)$ to denote the node $s_u$'s state and $X(u)$ obeys the following Bernoulli distribution:

$$X(u) = \begin{cases} 1, & s_u \text{ transmits the packet successfully,} \\ 0, & s_u \text{ fails to transmit the packet.} \end{cases} \tag{5}$$

*Definition 3.* The probability $\Pr(X(u) = 1)$ for a sensor node $s_u$ to transmit a packet successfully is called $s_u$'s packet transmit rate and is denoted by $\gamma(s_u)$.

The state model for a routing path $p = \{s_1, s_2, \ldots, s_m\}$ is represented by $Y(p)$ and $Y(p)$ obeys the following distribution:

$$Y(p) = \begin{cases} 1, & \text{path } p \text{ transmits the packet successfully,} \\ 0, & \text{path } p \text{ fails to transmit the packet} \end{cases} \tag{6}$$

and $Y(p) = \prod_{i=1}^{m} X(i)$. If the failure probabilities for the nodes in path $p$ are independent, $Y(p)$ also obeys the Bernoulli distribution and the packet transmit rate of path $p$ is denoted by $\gamma(p)$, which is equal to $\prod_{i=1}^{m} \gamma(s_i)$. If there exist $n$ paths $\chi = \{p_1, p_2, \ldots, p_n\}$ from the source node to the sink node, the packet transmit rate of path $\chi$ is calculated as

$$\gamma(\chi) = 1 - \prod_{i=1}^{n} (1 - \gamma(p_i)). \tag{7}$$

Acoustic signal has different transmission modes in shallow water (where the depth of the water is less than 100 meters) and deep water (where the depth of the water is more than 100 meters). In shallow water, the transmission of the acoustic signal is limited to a cylindrical area from bottom to the surface. The energy consumption for transmission in the shallow water is calculated as

$$\varepsilon_t = 10 \log_2 \delta(u,v) + \alpha\delta(u,v) \cdot 10^3, \tag{8}$$

where $\delta(u,v)$ denotes the Euler distance between the sender and the receiver and $\alpha$ is the absorption coefficient.

In deep water, the transmission of the acoustic signal is mainly with spherical diffusion. The energy consumption is caused by spherical diffusion and water absorption, which can be calculated as

$$\varepsilon_t = 20 \log_2 \delta(u,v) + \alpha\delta(u,v) \cdot 10^3. \tag{9}$$

*3.2. Packets Reception Strategies.* We denote the size of the transmit node's buffer pool by *buf* and $z_{ij}$ is the stochastic variable that represents the number of packets transmitted from the source node $s_i$ to the destination node $s_j$ per slot; then we get $z_{i\cdot} = \sum_{j=1}^{n} z_{ij}$. The number of packets from all sources to the destination node $s_j$ per slot is $z_{\cdot j} = \sum_{i=1}^{n} z_{ij}$. We denote by $x_{ij}$ the number of packets that $s_i$ retransmits to the destination node $s_j$; then the total number of retransmissions for $s_i$ is $x_{i\cdot} = \sum_{j=1}^{n} x_{ij}$ and the total number of retransmitted packets to the destination node $s_j$ is $x_{\cdot j} = \sum_{i=1}^{n} x_{ij}$. In the symmetric model, we get

$$\begin{aligned} E[z_{i\cdot}] &= E[z_{\cdot j}] = z, \quad \forall i, j = 0, 1, \ldots, n, \\ E[x_{i\cdot}] &= E[x_{\cdot j}] = x, \quad \forall i, j = 0, 1, \ldots, n. \end{aligned} \tag{10}$$

The First In First Out (FIFO) reception strategy selects the packet in the rightmost position of the current slot for reception. If there are several packets in the rightmost nonempty position, one is chosen deterministically. Intuitively, this strategy can be expected to have the least packet loss, as it selects the packet that has the shortest time still to spend in the receiver. Suppose the receiver is in state $S_t$, $t = 0, 1, \ldots, d$, we build a discrete time Markov chain with $d+1$ states $S_0, S_1, S_2, \ldots, S_d$. The chain changes state every $d+1$ received slots according to the arriving distribution $z_{\cdot j}$. Then the matrix of transition probabilities $p_{ik}$ is calculated as

$$p_{ik} = \sum_{z_{\cdot j} | S_i \to S_k} P(z_{\cdot j}), \quad \forall i, k = 0, 1, 2, \ldots, d. \tag{11}$$

Let $\{\pi_i \mid i = 0, 1, \ldots, d\}$ be the steady state of the model; then the packet retransmission distribution $P(x_{\cdot j} = k)$ is calculated as

$$P(x_{\cdot j} = k) = \sum_{i=0}^{d} \pi_i \cdot x_{ik}, \quad \forall k = 0, 1, \ldots, n. \tag{12}$$

Suppose the probability of packet rejection is $\alpha$, then the reception position distribution for a received packet $\rho_k$ is calculated as

$$\rho_k = \frac{1}{1-\alpha} \sum_{i=0}^{d} \pi_i \cdot p_{ik}, \quad \forall k = 0, 1, \ldots, n. \tag{13}$$

The Last In First Out (LIFO) packets reception strategy selects the packet in the leftmost position of the current slot for reception. If several packets are in the leftmost nonempty position, one is chosen deterministically. Intuitively, this strategy is the easiest to implement, although it may exhibit the worst behavior in terms of packet loss since it selects the packet that has the longest time still to spend in the receiver. In LIFO, the packet retransmission distribution $P(x_{\cdot j} = k)$ is calculated as

$$P\left(x_{\cdot j} = k\right) = \sum_{z_{\cdot j} | S_i \to S_k}^{d} P\left(z_{\cdot j}\right), \quad \forall k = 0, 1, \ldots, n. \tag{14}$$

Suppose the probability of packet rejection is $\beta$, then the reception position distribution for a received packet $\rho_k$ is calculated as

$$\rho_k = \frac{1}{1-\beta} \sum_{z_{\cdot j}} P\left(z_{\cdot j}\right), \quad \forall k = 0, 1, \ldots, n. \tag{15}$$

*3.3. Routing Protocol.* RRP consists of two phases: route discovery process and route maintenance process. During the process of route discovery, each packet carries the positions of the source node, the target node, and the relay node (i.e., the node that transmits this packet). Upon receiving a packet, a node computes the gravities of its neighbor nodes and chooses the neighbor with maximal gravity value to be the relay node that is in charge of forwarding the data packet.

*Definition 4.* Given a sensor node $s_i$ and its neighbor node $s_j$, the gravity function from $s_i$ to $s_j$ is defined as $G(\vec{ij})$ and its value is calculated as

$$\left|G(\vec{ij})\right| = \frac{\varepsilon_i \cdot \varepsilon_j \cdot \cos\left(A/\upsilon\right)}{\delta(i,j)^2} \tag{16}$$

with $\varepsilon_i$, $\varepsilon_i > 0$ and $\upsilon > 2$, where $\varepsilon_i$ and $\varepsilon_j$ denote the residual energy of sensor nodes $s_i$ and $s_j$, $A$ is the intersection angle between the direction of $s_i$ to $s_j$ and the direction of $s_i$ to $s_t$ (the target node), and a predefined parameter $\upsilon$ is used to adjust the impact of neighbor's direction angle on the gravity value.

The process of selecting a neighbor node as the relay node to forward a packet to a sink node is illustrated in Figure 1.

After sensor node $s_i$ receives a packet, it calculates gravity values of all neighbors within the normal transmission range $r_{\text{norm}}$ to the sink node $s_t$ and the neighbor node with the maximum gravity value is selected as the relay node to forward the packet. The transmission space of $s_i$ is a sphere centered at this node as we address 3D UWSN in this paper. In order to save energy, every sensor node only adopts

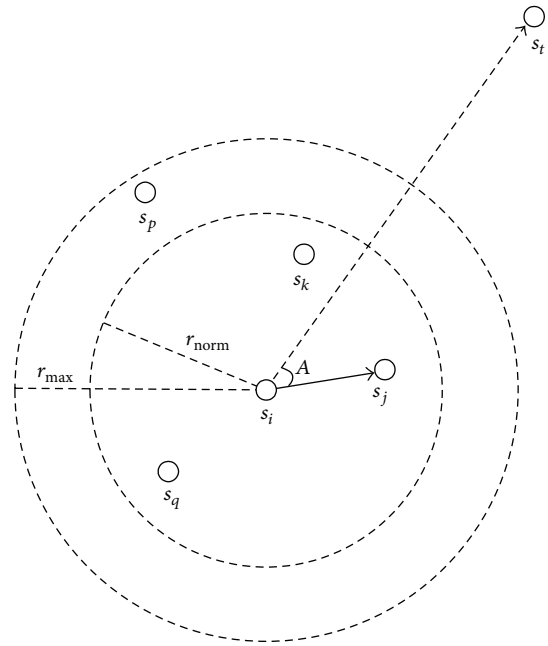



Figure 1: Relay node selection.

$r_{\text{norm}}$ during the process of normal routing. The maximal transmission range $r_{\max}$ is used to counteract the negative influence of node and link failures when necessary.

In order to describe our routing protocol clearly, the following data structures and definitions are given.

Given an arbitrary node $s_i$, then all nodes within node $s_i$'s transmission range are called 1-hop neighbors of $s_i$, which consist of $s_i$'s routing table $R\_table(s_i)$. All 1-hop neighbors of the nodes in $R\_table(s_i)$, that is, 2-hop neighbors of $s_i$, consist of $s_i$'s backup bin $B\_bin(s_i)$. Given an arbitrary routing path $p_i$, then all links in the path $p_i$ from the source node to the sink node are called main links of this path. Given an arbitrary main link $(s_j, s_k)$ of an arbitrary routing path $p_i$, where $s_j$ and $s_k$ are not the source node of $p_i$, then all main backup links of $(s_j, s_k)$ consist of the main backup set of $(s_j, s_k)$, which is denoted by $M_{jk}$, and if $s_j \in R\_table(s_i) \wedge s_k \in B\_bin(s_i)$, for all $s_b$ for all $s_m$ $((s_b \in B\_bin(s_i) \wedge s_b \in R\_table(s_j) \wedge s_b \neq s_k) \vee (s_m \in R\_table(s_i) \wedge s_k \in R\_table(s_m) \wedge s_m \neq s_j)) \to ((s_j, s_b) \in M_{jk} \wedge (s_m, s_k) \in M_{jk})$. The main backup links have the following characteristics: (1) the two endpoints of each main backup link belong to the two separate data structures, that is, routing table and backup bin; (2) each main backup link has and only has one endpoint overlaping with the corresponding main link. Likewise, all auxiliary backup links $(s_j, s_k)$ consist of the auxiliary backup set of $(s_j, s_k)$, which is denoted by $A_{jk}$ and if $s_j \in R\_table(s_i) \wedge s_k \in B\_bin(s_i)$, for all $s_a$ for all $s_u$ $(s_a \in R\_table(s_i) \wedge s_u \in B\_bin(s_i) \wedge s_a \neq s_j \wedge s_u \neq s_k) \to ((s_a, s_u) \in A_{jk})$. The auxiliary backup links have the following characteristics: (1) the two endpoints of each auxiliary backup link belong to the two separate data structures, that is, routing table and backup bin; (2) the two endpoints of each auxiliary backup link do not overlap with any endpoint of the corresponding main link.

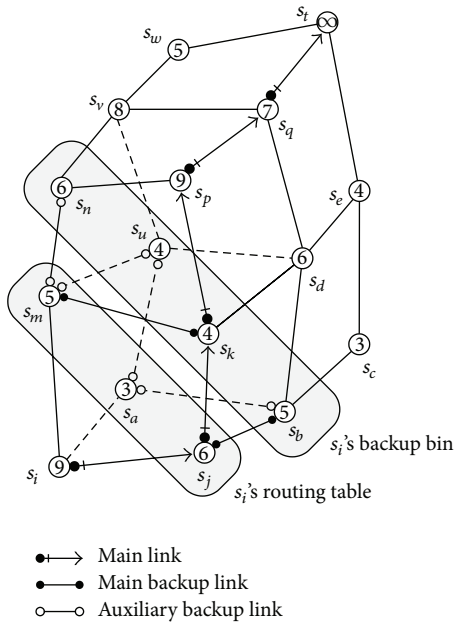



Figure 2: Building main and auxiliary backup links.

Given a special main link $(s_i, s_j)$ of an arbitrary routing path $p_i$, where $s_i$ is the source node of $p_i$, then $(s_i, s_j)$ has only main backup links, which is denoted by $M_{ij}$ and for all $s_a$ $(s_a \in R\_table(s_i) \wedge s_a \neq s_j) \rightarrow ((s_i, s_a) \in M_{ij})$. Given a special main link $(s_q, s_t)$ of an arbitrary routing path $p_i$, where $s_t$ is the sink node of $p_i$, then $(s_q, s_t)$ has only main backup links, which is denoted by $M_{qt}$ and for all $s_d$ $(s_d \in R\_table(s_q) \wedge s_d \neq s_t) \rightarrow ((s_q, s_d) \in M_{qt})$.

Figure 2 shows the process of building main and auxiliary backup links. Each circle denotes a sensor node and each number in the circle denotes the residual energy of local sensor node in a 3D UWSN. Sink nodes do not have any energy constraints because they are equipped with both radio frequency (RF) and acoustic modems and are deployed at the water surface. During the process of routing, all sensor nodes along the routing path will build backup links for the forthcoming link failure and node failure. For example, the source node $s_i$'s routing table $R\_table(s_i)$ includes three neighbor nodes $s_m$, $s_a$, and $s_j$. $s_i$'s backup bin $B\_bin(s_i)$ includes four 2-hop neighbor nodes $s_n$, $s_u$, $s_k$, and $s_b$. As we discussed above, the main link $(s_i, s_j)$ has only main backup links $(s_i, s_a)$ and $(s_i, s_m)$ since $s_i$ is the source node. The main link $(s_j, s_k)$ has two main backup links $(s_j, s_b)$ and $(s_m, s_k)$. Both of them have and only have one endpoint overlaping with the main link $(s_j, s_k)$. Moreover, the main link $(s_j, s_k)$ has four auxiliary backup links $(s_a, s_b)$, $(s_a, s_u)$, $(s_m, s_u)$, and $(s_m, s_n)$. The two endpoints of each auxiliary backup link belong to $R\_table(s_i)$, and $B\_bin(s_i)$ respectively. Moreover, the two endpoints of each auxiliary backup link do not overlap with any endpoint of the main link $(s_j, s_k)$.

In the same way, we obtain main and auxiliary backup links for other main links. Table 1 lists all main and auxiliary backup links for each main link along the routing path from the source node $s_i$ to the sink node $s_t$.

Table 1: Main and auxiliary backup links for each main link.

| Main link | Main backup set | Auxiliary backup set |
|---|---|---|
| $(s_i, s_j)$ | $(s_i, s_a)$, $(s_i, s_m)$ | None |
| $(s_j, s_k)$ | $(s_j, s_b)$, $(s_m, s_k)$ | $(s_a, s_b)$, $(s_a, s_u)$, $(s_m, s_u)$, $(s_m, s_n)$ |
| $(s_k, s_p)$ | $(s_k, s_d)$, $(s_k, s_m)$ | $(s_b, s_c)$, $(s_b, s_d)$ |
| $(s_p, s_q)$ | $(s_p, s_n)$, $(s_d, s_q)$ | $(s_m, s_n)$, $(s_d, s_e)$ |
| $(s_q, s_t)$ | $(s_q, s_v)$, $(s_q, s_d)$ | None |

**Input**: source node $s_i$, sink node $s_t$, TTL;
**Output**: routing path $p$;
(1) Queue $p \leftarrow \Phi$; //routing path initialization
(2) $p.enqueue(s_i)$; //add the node $s_i$ into the routing path $p$;
(3) **while** (TTL > 0) and $(s_t \notin R\_table(s_i))$ **do**
(4) **if** $|G(\vec{ij})| = \max\{|G(\vec{ij})|\}(s_j \in R\_table(s_i))$ **then**
(5) $p.enqueue(s_j)$;
(6) Builds $M_{ij}$, $A_{ij}$;
(7) $s_i.update(\varepsilon_i)$;
(8) **endif**
(9) $s_i \leftarrow s_j$;
(10) TTL--;
(11) **endwhile**
(12) **if** $(s_t \in R\_table(s_j))$ **then**
(13) $p.enqueue(s_t)$;
(14) **else** $p.clear()$; //remove all elements from $p$;
(15) **endif**
(16) **return** $p$;

Algorithm 1: Building the routing path and backup links.

Algorithm 1 describes the process of building the routing path and the corresponding backup links in detail.

All packets at relay nodes should have limited lifetime, which are controlled by TTL (Time-To-Live) information carried in the packet header. At first, the routing path $p$ is set to an empty queue structure and the source node $s_i$ is added into the routing path $p$ after initialization as described from line 1 to line 2. After that, $s_i$ will select a neighbor node as the relay node to forward a packet by means of calculating gravity values of the nodes in its routing table. Suppose $s_j$ is the candidate neighbor in that the gravity value from $s_i$ to $s_j$ is the biggest one among all neighbors. As a result, $s_j$ is added into the routing path $p$. Furthermore, main backup set and auxiliary backup set of $(s_i, s_j)$ are also constructed. The residual energy of node $s_i$ is updated in order to reflect the energy consumption of transmitting the packet to its neighbor. Of course, the TTL value is decreased by 1 so as to control the lifetime of the packet as described from line 3 to line 11. If the sink node $s_t$ is found within the given TTL value, it will be added into the routing path $p$; otherwise, all elements will be removed from $p$, which means no sink node is found as described from line 12 to line 16. After the packet arrives at the sink node, a robust routing path that can deal with the forthcoming link failure and node failure is also established.

**Input**: source node $s_i$, sink node $s_t$, TTL;
**Output**: repaired routing path $p'$;
(1) Queue $p' \leftarrow \Phi$;
(2) $s_j \leftarrow p.gethead()$; //returns the node at the head of path $p$;
(3) $p'.enqueue(s_j)$;
(4) $s_k \leftarrow p.gethead()$;
(5) **while** (TTL > 0) and ($s_k \in R\_table(s_j)$) **do**
(6) Sends the *Hello* packet from $s_j$ to $s_k$;
(7) **if** *ACK* packet is NOT received **then**
(8) **if** $|G(\vec{jb})| = \max\{|G(\vec{jb})|\}(s_b \in (M_{jk} \cup A_{jk}))$ **then**
(9) $p'.enqueue(s_b)$;
(10) Updates $M_{jb}$, $A_{jb}$;
(11) $s_j.update(\varepsilon_j)$;
(12) **else** switch $s_j$'s transmission range to $r_{\max}$;
(13) Updates $R\_table(s_j)$;
(14) **if** $|G(\vec{jb})| = \max\{|G(\vec{jb})|\}(s_b \in R\_table(s_j))$ **then**
(15) $p'.enqueue(s_b)$;
(16) Updates $M_{jb}$, $A_{jb}$;
(17) $s_j.update(\varepsilon_j)$;
(18) **else** $p'.clear()$;
(19) **break**;
(20) **endif**
(21) **endif**
(22) **endif**
(23) $s_j \leftarrow s_b$;
(24) $s_k \leftarrow p.gethead()$;
(25) TTL--;
(26) **endwhile**
(27) **return** $p'$;

Algorithm 2: Building the repaired routing path.

In many proactive routing protocols, the active nodes must send periodic update to other nodes even when the routing information is similar to the previous one. In RRP, route maintenance process is evoked only when a link or a node in a routing path is failed. Here, each node forwarding a *Hello* packet is responsible for confirming that the packet has been successfully received by a relay node. If it does not receive any *ACK* packet, the transmitting node treating the link to next hop is broken. It will mark all the nodes in the routing path that use that link as "invalid." Then it will return a route error to each node that has sent a packet over that broken link so that all those nodes can update their own routing tables and backup bins as well. Algorithm 2 describes the process of repairing failure link and failure node as well as building a repaired routing path.

From line 1 to line 4, the repaired routing path $p'$ is set to an empty queue structure and it will get the relay node from existent routing path $p$ one by one. From line 5 to line 7, the *Hello* packet is sent along the routing path until the *ACK* packet is not received from the downstream node, which means the route maintenance process is evoked. From line 8 to line 11, the upstream node chooses the relay node from the main and auxiliary backup sets and the node with maximal gravity value is selected as the candidate to forward the packet. Moreover, main backup set and auxiliary backup set of the new main link are updated and the residual energy of the upstream node is also decreased to a certain extent. Otherwise, the upstream node has to switch its transmission range to $r_{\max}$ and find the relay node from its new neighbors as described from line 12 to line 17. If all of these measures do not work, it means network partition may happen, which is not considered in this paper as it belongs to another complicated issue that needs to be thoroughly addressed. As a result, the repaired routing path $p'$ is cleared and the iteration process is broken as described from line 18 to line 26. At last, the repaired routing path $p'$ is returned as described in line 27.

Figure 3 illustrates the process of repairing failure link. Suppose a routing path $p = \{s_i, s_j, s_k, s_p, s_q, s_t\}$ is found after the execution of Algorithm 1. In order to build a repaired routing path $p'$, the source node $s_i$ retrieves a node (i.e., $s_j$) at the head of the path $p$ and forwards the packet to $s_j$. Likewise, $s_j$ retrieves $s_k$ at the head of the path $p$ and sends the packet to $s_j$. Since the link $(s_j, s_k)$ is broken, no *ACK* packet will be received at the node $s_j$ and $s_j$ will replace $(s_j, s_k)$ with a main backup link $(s_j, s_b)$. After $s_b$ receives the packet, it finds that $s_k$ is not in its routing table. So $s_b$ switches transmission range to $r_{\max}$ and updates its routing table. Suppose the distance between $s_b$ and $s_k$ is less than $r_{\max}$, then $s_b$ sends the packet to $s_k$. After that, $s_k$ forwards the packet along the rest of path $p$ until it arrives at the sink node $s_t$.

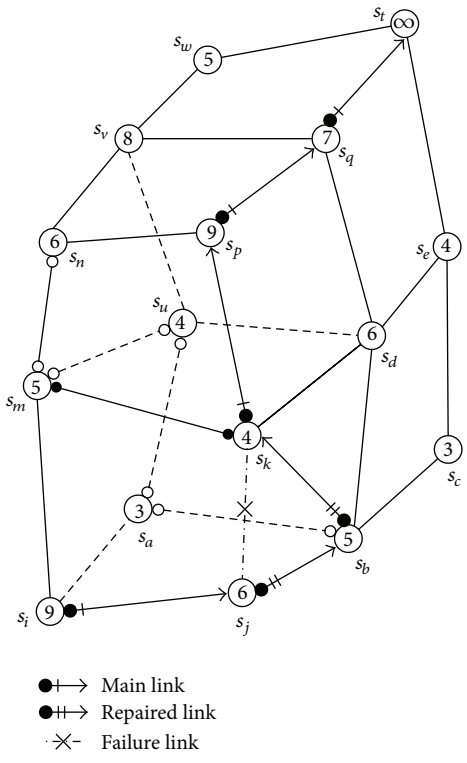



Figure 3: The process of repairing failure link.

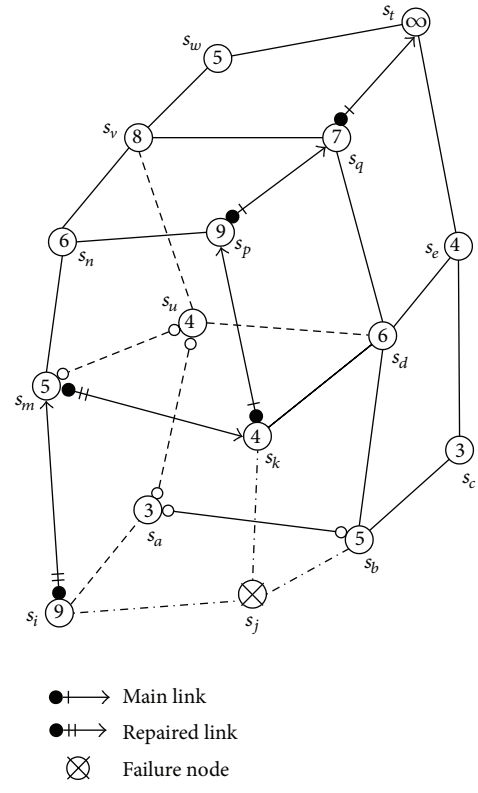



Figure 4: The process of repairing failure node.

Figure 4 illustrates the process of repairing failure node. Suppose a routing path $p = \{s_i, s_j, s_k, s_p, s_q, s_t\}$ is found after the execution of Algorithm 1. There is a failure node $s_j$ in the path $p$. The source node $s_i$ retrieves the node $s_j$ at the head of the path $p$ and forwards the packet to $s_j$. Since the node $s_j$ is broken, $s_i$ cannot receive any *ACK* packet and it has to replace $(s_i, s_j)$ with a main backup link $(s_i, s_m)$. After $s_m$ receives the packet, it finds that $s_j$ is not in its routing table. So $s_m$ switches transmission range to $r_{\max}$ and updates its routing table. Of course, $s_m$ still cannot build a link between $s_m$ and $s_j$. It retrieves the node $s_k$ at the head of the path $p$ and tries to forward the packet to $s_k$. When $s_k$ receives the packet, it means that the influence of node $s_j$'s failure is avoided and $s_k$ forwards the packet along the rest of path $p$ until it arrives at the sink node $s_t$.

In multiple sinks scenario, each source node calculates its distances to all sinks and chooses the nearest sink as the target node for data transmission. In this way, computational cost could be reduced without considering each sink's location during the process of choosing relay nodes.

RRP implicitly assigns the same quality to every link as well as most routing protocols that use hop count as their route metric. The reason is that link quality is not easy to be measured or calculated in complex underwater environments. A simple way is to apply the expected transmission count (ETX) as a measure of link quality [28]. ETX estimates the number of times a node will have to transmit a message before it successfully receives an acknowledgement. However, those protocols that use ETX to measure link quality must periodically send a large number of probe messages across a link in order to calculate its ETX value. In [29], the link quality is computed based on the success of past transmissions to its neighbors. In order to calculate the value of link quality, a coefficient called smoothing factor must be estimated in advance, which is used to control how quickly the influence of older transmissions decreases. A higher value of smoothing factor could be used for very variable underwater channels as it discounts older transmissions faster. However, the smoothing factor used for computing link quality is simply set to 0.7 without considering different acoustic channel conditions. In [30], the average link quality is estimated in terms of signal to interference and noise ratio (SINR) at the output of the equalizer at four receiving stations that deploy at different distances and orientations. The obtained results for shallow water scenarios (with the water column of 100 m) show that channel quality of the longer links (1000 m) is comparable to that of the shorter links (200 m) above the physical layer. However, these conditions are not enough to understand how the combination of time-varying sound speed profile and surface conditions affects the channel quality in the horizontal space.

The effectiveness of RRP is built on the basis of its routing tables and other related data structures. Suppose there are $n$ sensor nodes evenly deployed in a 3D monitoring region with average node degree of $d$. The forwarding process of VBF is to build a routing pipe between the source node and the destination node. Therefore, each sensor node needs $O(d)$ memory space for building its routing table. In DBR, each sensor node only records its 1-hop neighboring nodes

and selects the neighboring node with the minimal depth to be the first one to forward a packet. Accordingly, each sensor node in DBR also consumes $O(d)$ memory space for building its routing table. VAPR adopts a greedy clustering approach for improving routing performance over depth-based approaches. To this end, each sensor node requires the knowledge of 2-hop neighboring nodes. As a result, VAPR requires $O(d^2)$ memory space on each sensor node. In RRP, each sensor node needs to know its 1-hop neighboring nodes for building its routing table and 2-hop neighboring nodes for building its backup bin. Consequently, each sensor node of RRP demands $O(d^2)$ memory space for its routing activity. But $O(d^2)$ memory consumption is not a burden since the average node degree in UWSN is usually not high.

RRP is robust against link failure and node failure in that it uses backup links or enlarges the transmission range to build new links when main routing path is not available. Some of these links are interleaved and some are parallel, which are determined during the process of finding the sink node. Collision occurs when two or more nodes send data at the same time over the same transmission channel. Medium Access Control (MAC) protocols have been developed to assist each node to decide when and how to access the channel, which allow the sensor nodes to transmit data packets on the basis of a predefined schedule that will not cause the packet collision. If underlying MAC protocols are not available or out of service, RRP can adopt a round-robin scheduling method in order to avoid collision. The whole process is divided into two phases: round determination phase and timeslot allocation phase. During the routing process, each node calculates its round number based on the hops in current routing path and each round is scheduled sequentially. As a result, a downstream node usually gets bigger round number than an upstream node does. After that, each node will assign timeslots in pairs with its neighbors for packet transmission. The length of each round is determined by the number of timeslots that are necessary to avoid conflicts. Due to nonuniform deployment, some nodes may experience high traffic loads and cause more collisions than other nodes. Therefore, the number of timeslots varies from one round to another and is proportional to the residual energy of the neighbors. In this way, the initial setting of timeslot values may not be suitable for every node in the network. It achieves better global energy balance with the iterative execution of RRP.

RRP requires location awareness of neighbors, but no topology information needs to be exchanged among neighboring nodes. It is a localized and distributed routing algorithm.

## 4. Performance Evaluation

We use Aqua-Sim [31] as simulation framework to evaluate our approach. Aqua-Sim is an *ns*-2-based underwater sensor network simulator developed by underwater sensor network lab at University of Connecticut. Aqua-Sim can simulate the attenuation and propagation of acoustic signals. It can also simulate packet collision in underwater sensor networks.

We use a 3D region with size 1000 m × 1000 m × 1000 m and different number of sensor nodes varied from 100 to 600. Six sink nodes are randomly deployed at the water surface. We assume that the sink nodes are stationary and the sensor nodes follow the random-walk mobility pattern. Each sensor node randomly selects a direction and moves to the new position with a random speed between the minimal speed and maximal speed, which are 1 m/s and 5 m/s, respectively. The data generating rate varies from one packet per second to 6 packets per second with a packet size of 50 bytes (i.e., from 400 bps to 2.4 kbps). The communication parameters are similar to those on a commercial acoustic modem and the bit rate is 10 kbps. The normal and maximal transmission range is set to 50 m and 100 m, respectively, in all directions. The failure node/link ratio is set from 0% to 30% during the runtime of simulations.

We use the following metrics to evaluate the performance of routing protocols.

(1) *Packet delivery ratio* is defined as the ratio of the number of distinct packets received successfully at the sinks to the total number of packets generated at the source node. Although a packet may reach the sinks several times, these redundant packets are considered as only one distinct packet.

(2) *Network throughput* equals the total data bits received at the sinks divided by the simulation time.

(3) *Energy consumption* takes into account the total energy consumed in packet delivery, including transmitting, receiving, and idling energy consumption of all nodes in the network.

(4) *Average end-to-end delay* represents the average time taken by a packet to travel from the source node to any of the sinks.

We compared the performance of Robust Routing Protocol (RRP) with that of Vector-Based Forwarding (VBF) protocol, Depth-Based Routing (DBR) protocol, and Void-Aware Pressure Routing (VAPR) protocol.

In the first set of experiments, we compared the packet delivery ratio with the node/link failure ratio in different routing protocols. The number of nodes is set to 100 for each protocol. As shown in Figure 5, the packet delivery ratio of four routing protocols is inversely proportional to the node failure ratio. When node failure ratio increases from 0% to 30%, packet delivery ratio of VBF decreases from 58.1% to 37.2% and packet delivery ratio of DBR decreases from 73.8% to 49.2%. The curves of RRP and VAPR lie above those of VBF and DBR but intersect with each other. When node failure ratio is not more than 10%, the packet delivery ratio of VAPR is higher than that of RRP. When node failure ratio increases to 15% and above, the packet delivery ratio of RRP is higher than that of VAPR. The reason is that even when node density is low, VAPR still works well in the presence of voids, which in turn enhances the packet delivery ratio. When node failure ratio reaches 15% and above, the packet delivery ratio of VAPR diminishes rapidly while the packet delivery ratio of RRP shows a slow descent. The reason is that

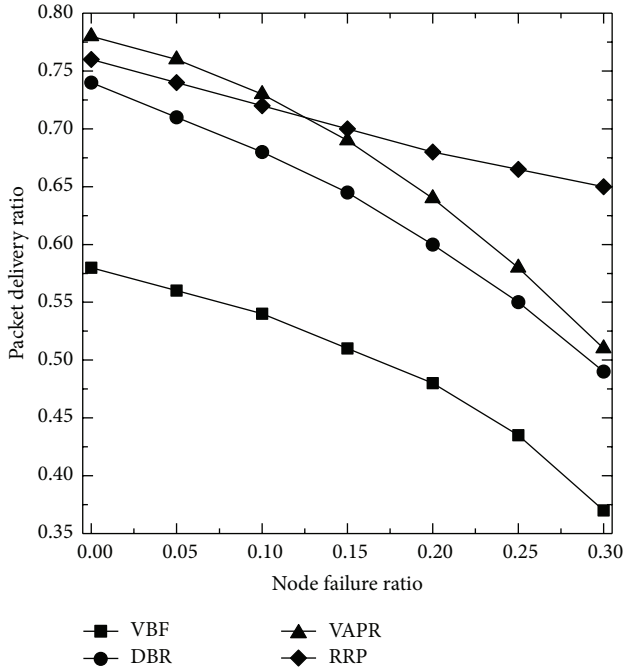



Figure 5: Packet delivery ratio versus node failure ratio.

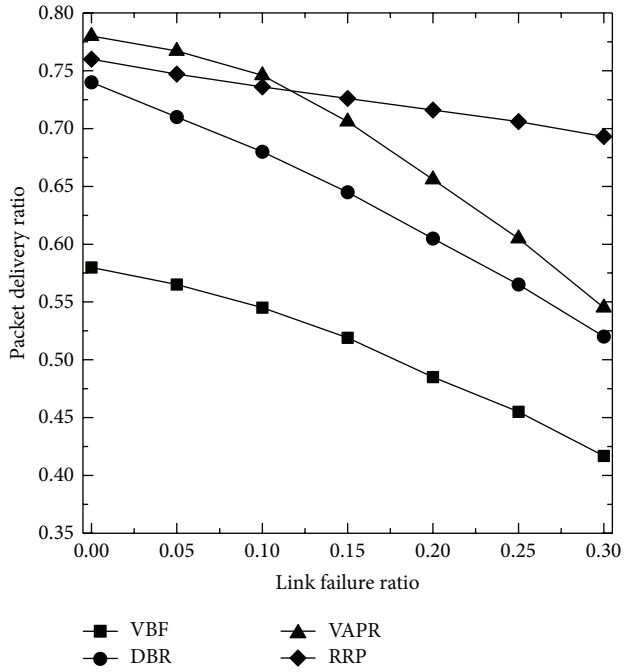



Figure 6: Packet delivery ratio versus link failure ratio.

the router recovery process of RRP counteracts the effects of node failure.

And then, we compared the packet delivery ratio with the link failure ratio in different routing protocols. As shown in Figure 6, the packet delivery ratio of four routing protocols is inversely proportional to the link failure ratio. Both RRP and VAPR still perform better than other routing protocols in the same circumstances. When link failure ratio is not more than 10%, the packet delivery ratio of VAPR is higher than that of RRP. When link failure ratio increases to 15% and above, the packet delivery ratio of RRP is higher than that of VAPR. Overall, RRP improves 21.8% of packet delivery ratio than that of VBF on average, 8.8% of packet delivery ratio than that of DBR on average, and 4.1% of packet delivery ratio than that of VAPR on average. Moreover, compared with Figure 5, the average packet delivery ratio of RRP in link failure is 72.6%, while the average packet delivery ratio of RRP in node failure is 70.2%. It means that RRP works better with link failure restoration than that with node failure restoration.

In the second set of experiments, we compared the network throughput with the node/link failure ratio in different routing protocols. As shown in Figure 7, the network throughput of four routing protocols is inversely proportional to the node failure ratio, but RRP and VAPR perform better than other routing protocols in the same circumstances. The curves of RRP and VAPR intersect with each other and the junction lies between 0.10 and 0.15 on horizontal axis. Overall, RRP improves 16.6% of network throughput than that of VBF on average, 5.6% of network throughput than that of DBR on average, and 1.1% of network throughput than that of VAPR on average.

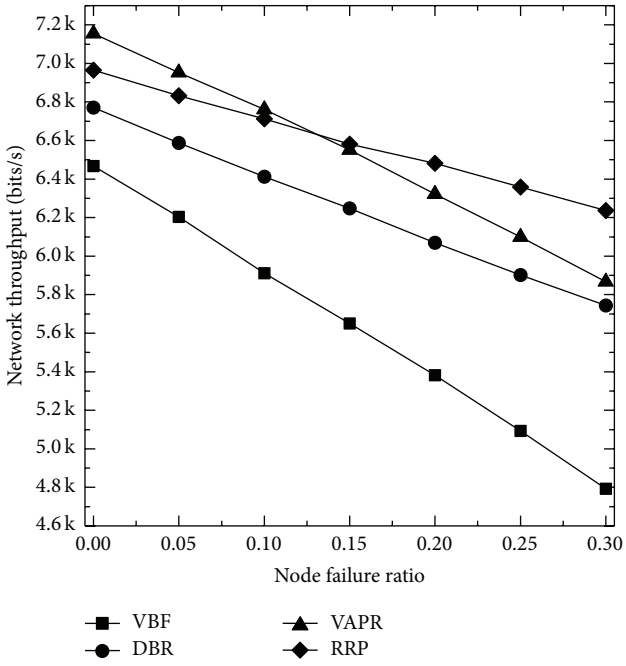



Figure 7: Network throughput versus node failure ratio.

Figure 8 illustrates the comparison of the network throughput with the link failure. The network throughput of four routing protocols is inversely proportional to the link failure ratio. Both RRP and VAPR still perform better than other routing protocols in the same circumstances. When link failure ratio is not more than 10%, the network throughput of VAPR is higher than that of RRP. When

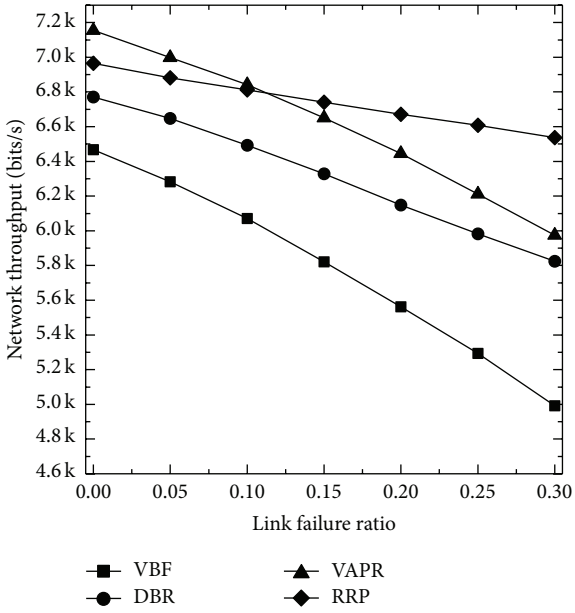



Figure 8: Network throughput versus link failure ratio.

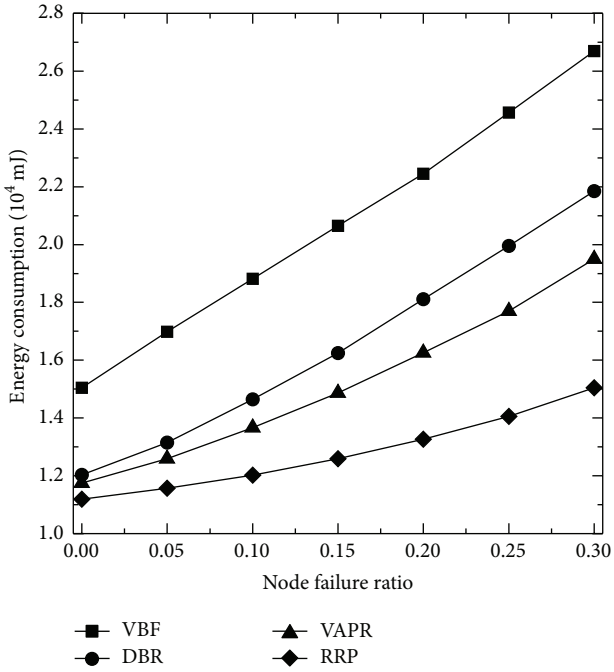



Figure 9: Energy consumption versus node failure ratio.

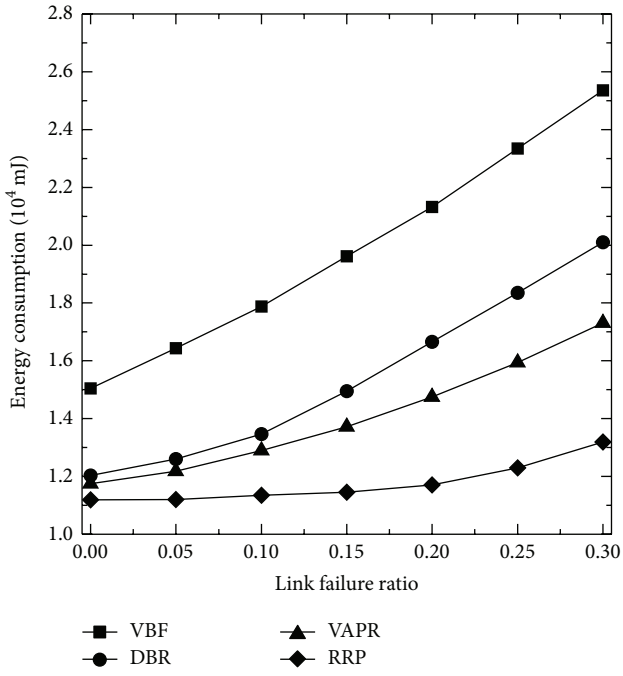



Figure 10: Energy consumption versus link failure ratio.

link failure ratio increases to 15% and above, the network throughput of RRP is higher than that of VAPR. Overall, RRP improves 16.8% of network throughput than that of VBF on average, 6.8% of network throughput than that of DBR on average, and 2.0% of network throughput than that of VAPR on average. Moreover, compared with Figure 7, RRP improves 9.2% of network throughput on the situation of link failure than that of node failure on average. When node failure ratio increases from 0% to 30%, RRP decreases 10.1% of network throughput, while when link failure ratio increases from 0% to 30%, RRP decreases 5.7% of network throughput.

In the third set of experiments, we compared the energy consumption with the node failure ratio in different routing protocols. As shown in Figure 9, the energy consumption of four routing protocols is proportional to the node failure ratio. RRP performs better than other routing protocols in the same circumstances. When node failure ratio improves from 0% to 30%, RRP decreases 39.3% of energy consumption than that of VBF on average, 24.1% of energy consumption than that of DBR on average, and 15.6% of energy consumption than that of VAPR on average.

And then, we compared the energy consumption with the link failure ratio in different routing protocols. As shown in Figure 10, the energy consumption of four routing protocols is proportional to the link failure ratio. RRP performs better than other routing protocols in the same circumstances. Moreover, compared with Figure 9, RRP consumes 86.5% of energy under the circumstances of 30% of link failure than that of RRP under the circumstances of 30% of node failure, which means that RRP saves more energy with link failure restoration than that of node failure restoration.

In the fourth set of experiments, we compared the average end-to-end delay with the node failure ratio in different routing protocols. As shown in Figure 11, the average end-to-end delay of four routing protocols is proportional to the node failure ratio. The curve of VBF is obviously above the other three curves. It indicates that VBF obtains the longest end-to-end delay among the four routing protocols. The curve of RRP intersects with that of DBR. When no node failure happens, the average end-to-end delay of RRP is lower than that of DBR. When node failure increases from 5% to 10%, the average end-to-end delay of RRP is higher than that of

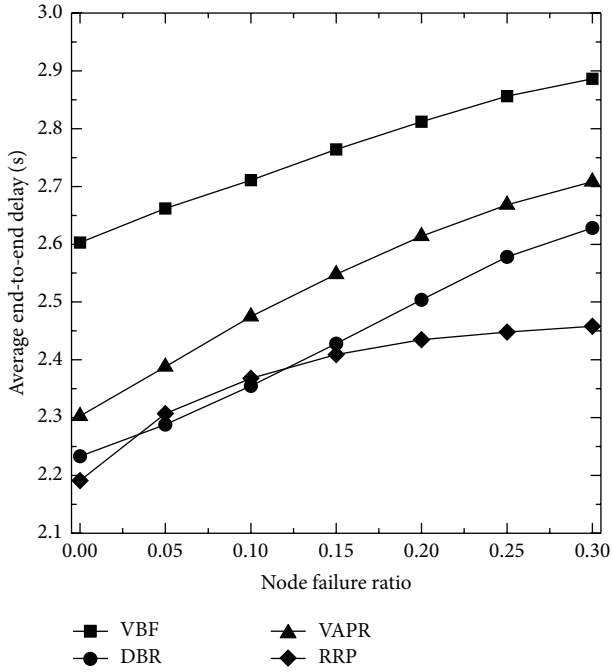



Figure 11: Average end-to-end delay versus node failure ratio.

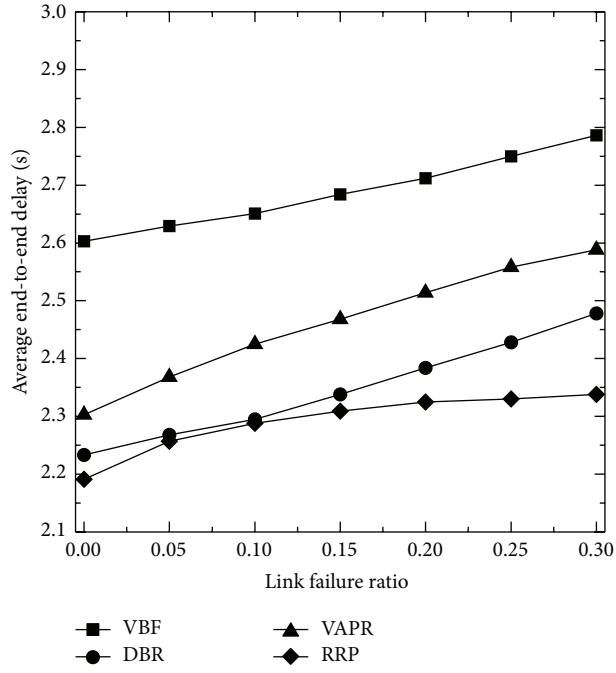



Figure 12: Average end-to-end delay versus link failure ratio.

DBR because RRP needs some time to replace the links of failure node with backup links. When node failure increases from 15% to 30%, the average end-to-end delay of RRP is lower than that of DBR again as more node failure happens, RRP may enlarge the transmission range, which cuts down the length of the routing path. The average end-to-end delay of VAPR is even higher than that of DBR because VAPR needs periodic beaconing using timers when nodes broadcast beacon messages and every neighbor would refresh its entry whenever beacon messages are received from other nodes.

And then, we compared the average end-to-end delay with the link failure ratio in different routing protocols. As shown in Figure 12, the average end-to-end delay of four routing protocols is proportional to the link failure ratio. RRP performs better than other routing protocols in the same circumstances as the curve of RRP is obviously below other protocols. Overall, RRP reduces 14.8% of the end-to-end delay than that of VBF on average, 2.4% of the end-to-end delay than that of DBR on average, and 6.9% of the end-to-end delay than that of VAPR on average. Compared with Figure 11, RRP decreases 3.7% of the average end-to-end delay under the circumstances of link failure than that of RRP under the circumstances of node failure.

In the last set of experiments, we compare the performance of different protocols under various node densities in metrics of packet delivery ratio, network throughput, energy consumption, and average end-to-end delay. The number of nodes is set from 100 to 600. As shown in Figure 13, the packet delivery ratio of four routing protocols is proportional to the number of nodes. The curves of RRP and VAPR lie above those of VBF and DBR but intersect with each other. When the number of nodes is not more than 300, the packet delivery ratio of VAPR is higher than that of RRP. When the number of nodes increases to 400 and above, the packet delivery ratio of RRP is higher than that of VAPR. The reason is that when node density is low, voids are more likely to appear in the network while VAPR can undercut their influence on the packet delivery ratio. When the number of nodes reaches 400 and above, the impact of voids on the packet delivery ratio is diminished. Meanwhile, RRP can find more appropriate

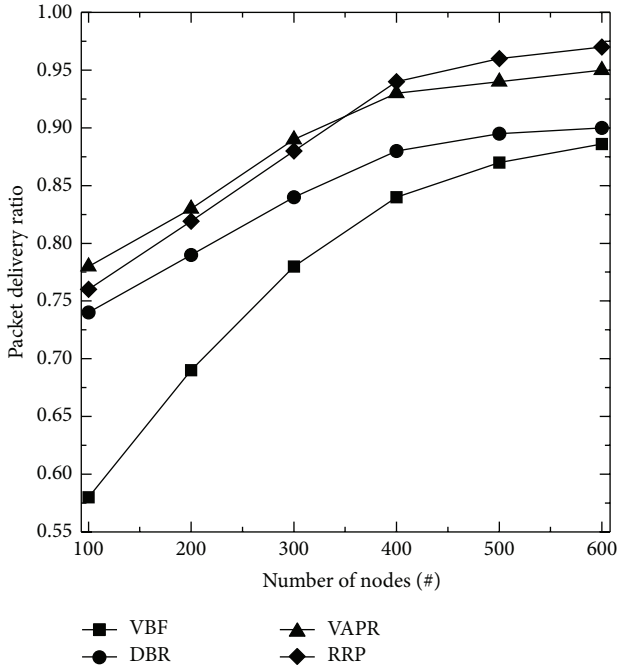



Figure 13: Packet delivery ratio versus number of nodes.

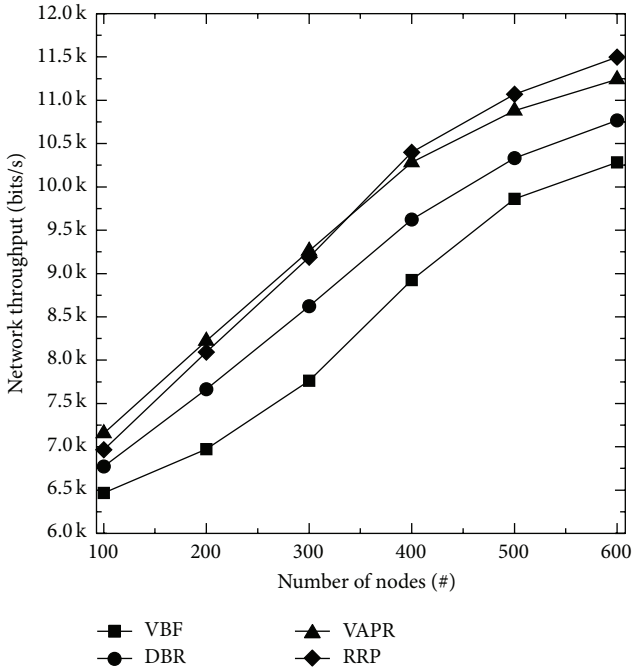



Figure 14: Network throughput versus number of nodes.

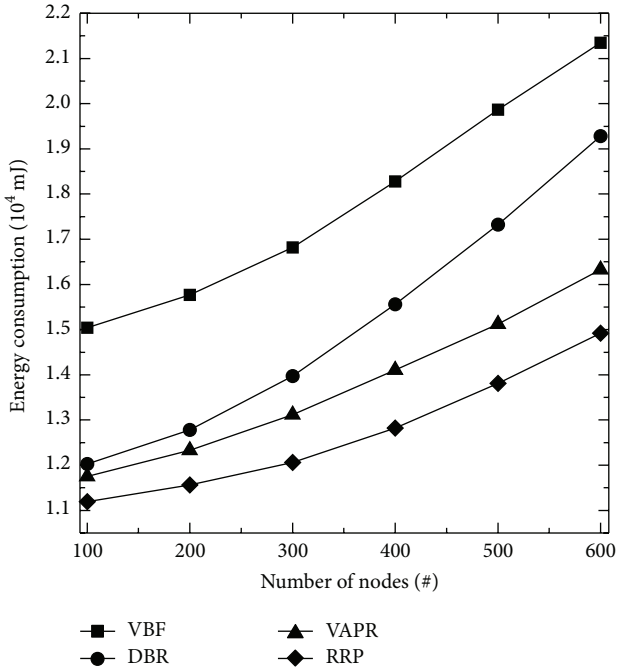



Figure 15: Energy consumption versus number of nodes.

nodes as candidates to forward the packets and the path from the source node to the target node becomes closer to the optimal path, which in turn improves its packet delivery ratio.

And then, we compared the network throughput with the number of nodes in different routing protocols. As shown in Figure 14, the network throughput of four routing protocols is proportional to the number of nodes. When the number of nodes is not more than 300, the network throughput of VAPR is higher than that of RRP. When the number of nodes increases to 400 and above, the network throughput of RRP is higher than that of VAPR. Overall, RRP improves 13.8% of network throughput than that of VBF on average, 6.4% of network throughput than that of DBR on average, and 0.3% of network throughput than that of VAPR on average.

Next, we compared the energy consumption with the number of nodes in different routing protocols. As shown in Figure 15, the energy consumption of four routing protocols is proportional to the number of nodes. RRP performs better than other routing protocols in the same circumstances. When the number of nodes improves from 100 to 600, RRP decreases 28.7% of energy consumption than that of VBF on average, 16.1% of energy consumption than that of DBR on average, and 7.7% of energy consumption than that of VAPR on average.

At last, we compared the average end-to-end delay with the number of nodes in different routing protocols. As shown in Figure 16, the average end-to-end delay of four routing protocols is inversely proportional to the number of nodes. The reason is that when the number of nodes increases, the path from the source node to the target node is closer to the optimal path; therefore, the end-to-end delay decreases. Moreover, when the node density is low, the average end-to-end delay decreases rapidly with density. However, when there are more than 400 nodes deployed in the 3D region, the average end-to-end delay decreases slightly under different node densities. Overall, RRP reduces 22.1% of the end-to-end delay than that of VBF on average, 3.6% of the end-to-end delay than that of DBR on average, and 9.9% of the end-to-end delay than that of VAPR on average.

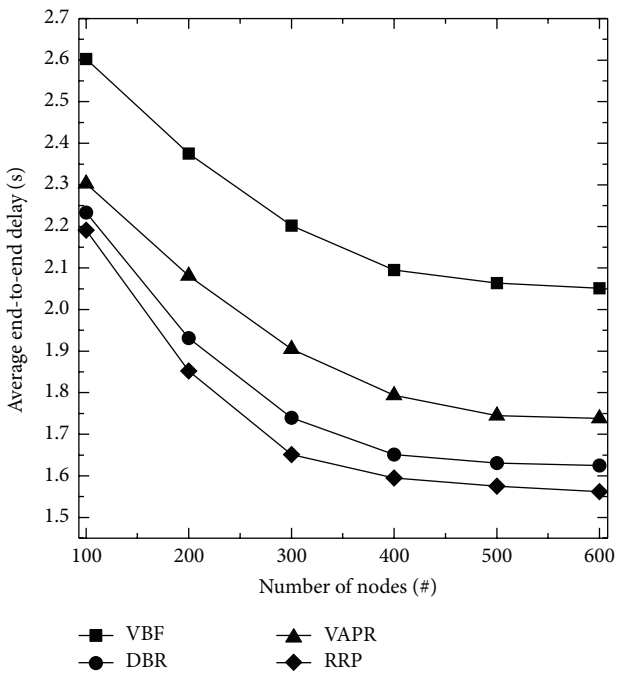



Figure 16: Average end-to-end delay versus number of nodes.

## 5. Conclusion

In 3D UWSNs, acoustic communication links face high bit error rate, temporary losses, or even permanent failure, which in turn undermine robustness and stability of the routing protocols. In order to deal with the unreliable and unstable nature of the acoustic medium, we have proposed a Robust Routing Protocol for the new features of the 3D UWSNs in order to meet the actual underwater environmental performance needs. RRP is robust against link failure and node failure in that it uses backup links to forward the data packets or enlarges the transmission range to build new links when main routing path is not available. As a result, the routing robustness issue has been well addressed in the proposed route discovery and maintenance processes. Simulation results show that RRP can reduce node failure and link failures' impact in metrics of packet delivery ratio, end-to-end delay, network throughput, and energy consumption.

## Acknowledgments

This work was sponsored by the National Nature Science Foundation of China (61202370, 51279099), the Innovation Program of Shanghai Municipal Education Commission (14YZ110), the Shanghai Pujiang Program from Science and Technology Commission of Shanghai Municipality (11PJ1404300), and the Open Program of Shanghai Key Laboratory of Intelligent Information Processing (IIPL-2011-008).

## References

[1] R. B. Manjula and S. M. Sunilkumar, "Issues in underwater acoustic sensor networks," *International Journal of Computer and Electrical Engineering*, vol. 3, no. 1, pp. 101–110, 2011.

[2] M. T. Kheirabadi and M. M. Mohamad, "Greedy routing in underwater acoustic sensor networks: a survey," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 701834, 21 pages, 2013.

[3] C. Detweiler, M. Doniec, I. Vasilescu, and D. Rus, "Autonomous depth adjustment for underwater sensor networks: design and applications," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 1, pp. 16–24, 2012.

[4] S. Basagni, C. Petrioli, R. Petroccia, and M. Stojanovic, "Optimized packet size selection in underwater wireless sensor network communications," *IEEE Journal of Oceanic Engineering*, vol. 37, no. 3, pp. 321–337, 2012.

[5] J. M. Jornet, M. Stojanovic, and M. Zorzi, "On joint frequency and power allocation in a cross-layer protocol for underwater acoustic networks," *IEEE Journal of Oceanic Engineering*, vol. 35, no. 4, pp. 936–947, 2010.

[6] G. Isbitiren and O. B. Akan, "Three-dimensional underwater target tracking with acoustic sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 8, pp. 3897–3906, 2011.

[7] D. Pompili, T. Melodia, and I. F. Akyildiz, "Three-dimensional and two-dimensional deployment analysis for underwater acoustic sensor networks," *Ad Hoc Networks*, vol. 7, no. 4, pp. 778–790, 2009.

[8] A. Y. Teymorian, W. Cheng, L. Ma, X. Cheng, X. Lu, and Z. Lu, "3D underwater sensor network localization," *IEEE Transactions on Mobile Computing*, vol. 8, no. 12, pp. 1610–1621, 2009.

[9] M. Ayaz, I. Baig, A. Abdullah, and I. Faye, "A survey on routing techniques in underwater wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1908–1927, 2011.

[10] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, "The challenges of building scalable mobile underwater wireless sensor networks for aquatic applications," *IEEE Network*, vol. 20, no. 3, pp. 12–18, 2006.

[11] Z. H. Jiang, "Underwater acoustic networks-issues and solutions," *International Journal of Intelligent Control and Systems*, vol. 13, no. 3, pp. 152–161, 2008.

[12] I. F. Akyildiz, D. Pompili, and T. Melodia, "State-of-the-art in protocol research for underwater acoustic sensor networks," in *Proceedings of the 1st ACM International Workshop on Underwater Networks (WUWNet '06)*, pp. 7–16, Los Angeles, Calif, USA, September 2006.

[13] H.-P. Tan, W. K. G. Seah, and L. Doyle, "A multi-hop ARQ protocol for underwater acoustic networks," in *Proceeding of the OCEANS '07*, Aberdeen, Scotland, June 2007.

[14] A. Vahdat and D. Becker, "Epidemic routing for partially connected Ad Hoc networks," Tech. Rep. TR CS-200006, 2000.

[15] D. Pompili, T. Melodia, and I. F. Akyildiz, "Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 298–309, Los Angeles, Calif, USA, September 2006.

[16] P. Xie, J. H. Cui, and L. Lao, "VBF: vector-based forwarding protocol for underwater sensor networks," in *Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, vol. 3976 of *Lecture Notes in Computer Science*, pp. 1216–1221, Springer, 2006.

[17] J. M. Jornet, M. Stojanovic, and M. Zorzi, "Focused beam routing protocol for underwater acoustic networks," in *Proceedings of the 3rd International Workshop on Underwater Networks (WUWNet '08)*, pp. 75–81, San Francisco, Calif, USA, September 2008.

[18] M. Zorzi, P. Casari, N. Baldo, and A. F. Harris III, "Energy-efficient routing schemes for underwater acoustic networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1754–1766, 2008.

[19] H. Yan, Z. Shi, and J. H. Cui, "DBR: depth-based routing for underwater sensor networks," in *NETWORKING 2008 Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, vol. 4982 of *Lecture Notes in Computer Science*, pp. 72–86, Springer, 2008.

[20] U. Lee, P. Wang, Y. Noh, F. M. L. Vieira, M. Gerla, and J. H. Cui, "Pressure routing for underwater sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications*, pp. 1–9, San Diego, Calif, USA, 2010.

[21] M. Ayaz and A. Abdullah, "Hop-by-hop dynamic addressing based (H2-DAB) routing protocol for underwater wireless sensor networks," in *Proceedings of the International Conference on Information and Multimedia Technology (ICIMT '09)*, pp. 436–441, Jeju Island, Korea, December 2009.

[22] Y. Noh, U. Lee, P. Wang, B. Choi, and M. Gerla, "VAPR: void-aware pressure routing for underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 5, pp. 895–908, 2013.

[23] M. Ayaz, A. Abdullah, and I. Faye, "Hop-by-hop reliable data deliveries for underwater wireless sensor networks," in *Proceedings of the 5th International Conference on Broadband Wireless Computing, Communication and Applications (BWCCA '10)*, pp. 363–368, November 2010.

[24] J. Xu, K. Li, and G. Min, "Reliable and energy-efficient multipath communications in underwater sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1326–1335, 2012.

[25] Z. Zhou, Z. Peng, J.-H. Cui, Z. Shi, and A. Bagtzoglou, "Scalable localization with mobility prediction for underwater sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 3, pp. 335–348, 2011.

[26] W. Cheng, A. Y. Teymorian, L. Ma, X. Cheng, X. Lu, and Z. Lu, "Underwater localization in sparse 3D acoustic sensor networks," in *27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 798–806, Phoenix, Ariz, USA, April 2008.

[27] H.-P. Tan, Z. A. Eu, and W. K. G. Seah, "An enhanced underwater positioning system to support deepwater installations," in *Proceedings of the MTS/IEEE Biloxi—Marine Technology for Our Future: Global and Local Challenges (OCEANS '09)*, pp. 1–8, Biloxi, Miss, USA, October 2009.

[28] E. A. Carlson, P. P. Beaujean, and E. An, "Location-aware routing protocol for underwater acoustic networks," in *Proceedings of MTS/IEEE OCEANS '06*, pp. 1–6, Boston, Mass, USA, 2006.

[29] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini, "Channel-aware routing for underwater wireless networks," in *Proceedings of MTS/IEEE OCEANS '12*, pp. 1–9, Yeosu, Korea, 2012.

[30] B. Tomasi, J. Preisig, and M. Zorzi, "On the spatial correlation in shallow water and its impact on networking protocols," in *Proceedings of MTS/IEEE OCEANS '12*, pp. 1–7, Yeosu, Korea, 2012.

[31] P. Xie, Z. Zhou, Z. Peng et al., "Aqua-sim: an NS-2 based simulator for underwater sensor networks," in *Proceedings of the MTS/IEEE Biloxi—Marine Technology for Our Future: Global and Local Challenges, OCEANS '09*, pp. 1–7, Biloxi, Miss, USA, October 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 147839, 9 pages
http://dx.doi.org/10.1155/2013/147839

*Research Article*

# PDA: A Novel Privacy-Preserving Robust Data Aggregation Scheme in People-Centric Sensing System

**Ziling Wei, Baokang Zhao, and Jinshu Su**

*School of Computer Science, National University of Defense Technology, Changsha, Hunan, China*

Correspondence should be addressed to Baokang Zhao; bkzhao@nudt.edu.cn

Received 27 June 2013; Accepted 4 October 2013

Academic Editor: Qin Xin



With the advancement of wireless communication technologies, mobile phones, PDAs, and car embedded devices are equipped with sensors, such as sound and image. People can apply these devices to form a new sensing network called people-centric sensing network. And this network offers new opportunities for cooperative sensing applications. However, it introduces some challenges, including security challenge and robust challenge. As sensor nodes need to send their individual sensed data to an aggregator node and these data are related to users' real life, privacy-preserving data aggregation is a challenge issue. As a node could become offline or a message could be lost before reaching the aggregator, retaining the correctness of the aggregate computed is important. In this paper, we present the design of PDA, a novel privacy-preserving robust data aggregation scheme in people-centric sensing system. Based on $K$-anonymity, homomorphic encryption, and secret sharing, PDA can support a wide range of statistical additive and non-additive aggregation functions such as Sum, Subtraction, Average, Count, Max/Min, and Median without leaking individual sensed data. Moreover, PDA is robust to node failure and data loss. We also evaluate the efficacy and efficiency of PDA. The result shows that our scheme can achieve the security and robust goal under a reasonable cost.

## 1. Introduction

Nowadays, technological advances in sensing, computation, storage, and wireless communications turn the mobile devices carried by people into a global mobile sensing device [1]. For example, currently smartphone is equipped with accelerometers, audio, location, and image sensors. So the mobile device's owner becomes sensor custodian that can use the device to collect meaningful context data [2]. Therefore, people as individuals or special interest groups can apply the new sensing devices to form sensing networks which are called people-centric sensing networks [3]. This transformation provides a chance to create intelligence systems that collect data from widespread public participations [4]. There are many systems in present, such as BikeNet [5], CitySense [6], Mobiscopes [7], Urban Sensing [8], SenseWeb [9], and CarTel [10]. They could be used to monitor environmental pollution, temperature, or noise intensity of urban areas. In order to declare the impact of people-centric sensing networks, we will introduce the CitySense. CitySense passively "senses" the most popular places based on actual real-time activity and displays a live heat map. The application intelligently leverages the inherent wisdom of crowds without any change in existing user behavior, in order to navigate people to the hottest spots in a city [6]. So we could find the most popular places to visit easily through CitySense. That is, people-centric sensing networks have a great potential to provide large-scale, flexible, and global sensing network services.

Even though people-centric sensing networks can offer many benefit's to users, there are still some challenges. One of them is that the mobile device collects context information closely related to user's real life. So the concern of user privacy is a challenge for people-centric sensing networks. As users usually do not gain a direct benefit from reporting data, users are unwilling to share their data if their privacy is at risk [11]. The procedure of data aggregation is the most likely to by attacked by adversary. The other challenge is that a node could become offline or a message could be lost before reaching the aggregator. So we should consider a solution that ensures that these problems could not affect the correctness of

the aggregate computed. These challenges motivate our design of PDA, a novel privacy-preserving robust data aggregation scheme in people-centric sensing system.

Although the people-centric sensing network is becoming popular, there is relatively little work focusing on its privacy aspects and robust to node or communication failure. The shortage of prior work can be concluded in three points. The first point is that prior work has generally focused on preserving user's privacy to a certain extent [11–14]. However, these researches are not robust to node or communication failure. The second point is that these researches rarely address the privacy issues of data aggregation which is the most likely to be attacked by adversary [15–17]. The third point is that most of prior work did not suit for the peculiarity of this network. For example, the devices are no longer owned by a single authority but belong to individuals; therefore, the devices could be mobile and malicious.

Though the above analysis, we could know that none of the existing schemes could overcome the challenges that one mentioned above in people-centric sensing networks. The contributions of this paper are summarized as follows.

(i) We formulate the threat model in people-centric sensing networks. These models enable researchers to study the privacy-preserving in people-centric sensing networks conveniently.

(ii) We present a novel scheme, PDA, to protect users' privacy against threats and be robust to node or communication failure in people-centric sensing networks. This scheme is based on $K$-anonymity, homomorphic encryption, and secret sharing. And it can support a wide range of statistical additive and nonadditive aggregation functions such as Subtraction, Sum, Average, Count, Max/Min, and Median.

The paper is organized as follows. We present the related work in Section 2. In Section 3, we introduce the basic of $K$-anonymity, homomorphic encryption, and secret sharing. We present the system architecture, threat model, and design goal in Section 4. We present our scheme in detail in Section 5. Then, we give the evaluation of PDA in Section 6. Conclusion could be found in Section 7.

## 2. Related Work

Although the people-centric sensing network is becoming popular, there is relatively little work focusing on its privacy aspects and robust to node or communication failure.

In [11], AnonySense architecture for anonymous tasking and reporting has been proposed. Shi et al. [12] present the PriSense scheme to protect privacy during the aggregation process of data. And this scheme support additive and nonadditive aggregation functions. Feng et al. [13] present a scheme to realize the data aggregation. However, it only supports additive aggregation functions such as max/min and median at the sacrifice in data accuracy. Zhang et al. [14] also present a scheme which supports both additive aggregation functions and nonadditive ones. These researches can preserve users' privacy to a certain extent. However, these researches are not robust to node or communication failure.

Wagner [15] present a scheme that protect the data aggregation in the presence of false data injection attack by a few malicious nodes and provide guidelines for selecting appropriate aggregation functions in a sensor network. Chan et al. [16] design an algorithm which the base station could detect if the computed aggregate was falsified. Conti et al. [17] present a scheme to preserve the privacy of robust data aggregation in wireless sensor network. However, these researches did not address the privacy issues of data aggregation which is the most likely to be attacked by adversary. In addition, as the devices are no longer owned by a single authority but belong to individuals in people-centric sensing network, the devices could be mobile and malicious. Therefore, these researches are not suitable for people-centric sensing applications.

Therefore, the novelty and the main differences with the existing work in the literature can be concluded in four points.

(i) PDA is looking at the new sensing network called people-centric sensing network. Most of the existing schemes cannot suit for the peculiarity of this network. For example, the devices are no longer owned by a single authority but belong to individuals; therefore, the devices could be mobile and malicious.

(ii) The problem that PDA solved contains security challenge and robust challenge. To the best of my knowledge, PDA is the first scheme that can solve these two problems of people-centric sensing network. Moreover, we not only consider the external internal threat, but also the internal threat. This is out of reach for a lot of existing work.

(iii) PDA is a clever use of $K$-anonymity, homomorphic encryption, and secret sharing. There is little research to use these schemes for robust challenge. Moreover, the combination of these schemes works very well.

## 3. Basic Schemes

In this section, we will introduce the basic schemes which are implemented in our scheme.

*3.1. K-Anonymity.* Society is experiencing exponential growth in the number and variety of data collections containing person-specific information as computer technology, network connectivity, and disk storage space become increasingly affordable. But the security of the database is a big challenge. Some researches propose that we can get a certain protection by removing some sensitive message. However, in most of these cases, the remaining data can be used to reidentify individuals by linking or matching the data to other data or by looking at unique characteristics found in the released data.

In order to solve this problem, Samarati [18] propose $K$-anonymity in 1998. In $K$-anonymity, an attacker cannot distinguish the specific individual who belongs to the private information by adding a certain amount of fake individuals. Therefore, we can prevent the disclosure of personal privacy.

Table 1: An example of $K$-anonymity.

| Num | Sex | Birth | Disease |
|---|---|---|---|
| 1 | F | 78/0∗/∗∗ | Obesity |
| 2 | F | 78/0∗/∗∗ | Chest pain |
| 3 | M | 83/0∗/∗∗ | Short breath |
| 4 | M | 83/0∗/∗∗ | Obesity |

$K$-anonymity: Let $RT(A_1, \ldots, A_n)$ be a table and $QI_{RT}$ the quasi-identifier associated with it. $RT$ is said to satisfy $K$-anonymity if and only if each sequence of data in $RT[QI_{RT}]$ appears with at least $K$occurrences in $RT[QI_{RT}]$.

Table 1 provides an example of a table $RT$ that adheres to $K$-anonymity. The quasi-identifier for the table is $QI_{RT}$ = {Sex, Birth} and $K = 2$. That is, for each sequence of data in $RT[QI_{RT}]$, there are at least 2 occurrences of those data in $RT[QI_{RT}]$ [19].

As the sensing data will be aggregated to aggregation servers and it could be malicious, we could guarantee security based on $K$-anonymity.

*3.2. Homomorphic Encryption.* Homomorphic encryption technology was originally developed in 1978 by Rivest [20]. It is an encryption transformation technology that allows operation of the ciphertext. Homomorphic encryption was first used to encrypt the statistics. It is ensured that the user can operate on sensitive data by the homogeneity of the algorithm without revealing the data information. That is, homomorphic encryption is a form of encryption which allows specific types of computations to be carried out on ciphertext and obtain an encrypted result which matches the decrypted result of operations performed on the plaintext. A cryptosystem which supports both addition and multiplication is known as fully homomorphic encryption. Using such a scheme, any circuit can be homomorphically evaluated, effectively allowing the construction of programs which may be run on encryptions of their inputs to produce an encryption of their output. This is equivalent to not knowing that the problem also gives the answer to the problem.

Homomorphic encryption is built on the foundations of algebra theory. Then we will introduce the basic idea of homomorphic encryption. Assuming that $E$ and $D$ are the encryption and decryption functions, plaintext data is a finite set $M = \{m_1, m_2, \ldots, m_n\}$, $\alpha$ and $\beta$ on behalf of computing. If $D(\alpha(E(m_1), E(m_2), \ldots, E(m_n))) = \beta\{m_1, m_2, \ldots, m_n\}$ is right, $(E, D, \alpha, \beta)$ is called homomorphism. Homomorphic encryption has attracted many attentions of many scholars in recent years, such as Yu. Yu proposed homomorphic encryption algorithm; Craig Gentry introduced an algorithm that $\alpha$ and $\beta$ are the same operation, and he referred to as the "ideal lattice" mathematical objects that allow people to make operation of the encryption data [21, 22].

*3.3. Secret Sharing.* Secret sharing distributes, preserves, and restores the secret method. And it is an important tool to achieve secure multiparty computation. Secret sharing refers to method for distributing a secret among a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient number, of possibly different types, of shares are combined together; individual shares are of no use on their own [23]. Secret sharing technology extends to many areas and practice systems, such as video, voice, and other fields. It is a technology which combines theory and practice.

The general type of secret sharing is threshold method, generally with $(m, n)$-threshold method. In this type, there are one dealer and n participants $(P_1, P_2, \ldots, P_n)$. The dealer slices the secret $S$ to $(s_1, s_2, \ldots, s_n)$ and shares the slices to the participants. The dealer accomplishes this by giving each participant a share in such a way that any group of $m$ or more participants can together reconstruct the secret, but no group of fewer than $m$ players can. That is, it takes $m$ points to define a polynomial of degree $m - 1$. The method is to create a polynomial of degree $m - 1$ with the secret as the first coefficient and the remaining coefficients picked at random. Next find $n$ points on the curve and give one to each of the players. When at least $m$ out of the $n$ players reveal their points, there is sufficient information to fit a $(m - 1)$th degree polynomial to them, the first coefficient being the secret [24, 25].

The basic method of secret sharing contains Shamir secret sharing method, Blakley secret sharing method, Karin-Greene-Hellman secret sharing method and so on.

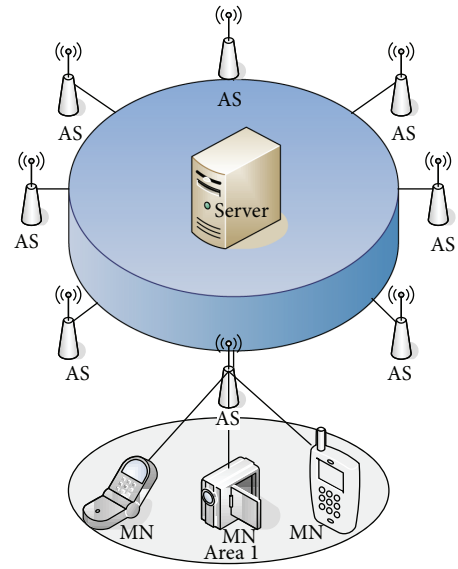



Figure 1: System model.

## 4. System Architecture

In this section, we present the system model, threat model, and design goal.

*4.1. System Model.* The architecture of people-centric sensing system is demonstrated in Figure 1.

Then, we give the components of people-centric sensing system.

(i) The mobile nodes (MNs) are all kinds of devices with sensing the environment and reporting the sensed data, such as PDAs, smartphones, and laptops. And these MNs are carried by person or vehicles. Each MN has a unique ID in this system.

(ii) The aggregation servers (ASs) provide network access services for MNs. Each AS is in charge of a certain region referred to as an area. Each MN in an area should send its sensed data to the AS which is in charge of the area. The AS should support a wide range of statistical additive and nonadditive aggregation functions such as Sum, Subtraction, Average, Count, Max/Min and Median.

(iii) The server should realize three main functions. The first one is that the server manages MNs' identity-related information and stores the fake IDs table for $K$-anonymity. The second one is that the server is responsible for registering MNs that wish to participate. The third one is that the server stores the data that was computed by the ASs. It also provides services for the applications based on the data.

Without loss of generality, we give the assumption of PDA. We assume that all the mobile nodes can access to an AS. We assume that all ASs can access to the server. That is to say, each MN can access to the server through ASs. We assume that MNs report the data to the ASs when they achieve a request from the server. The server sends the request which specify what sensors to report and when to sense.

*4.2. Threat Model.* In people-centric sensing networks, the data which is received from MNs unavoidably relates to a lot of personal information. If the information is known by adversary, the user could suffer many troubles. In this section, we will analyze the threat mode. As the procedure of data aggregator is the most likely to be attacked by adversary, we are mainly concerned with the threat of this procedure. We separate the threats into two classes: the internal threat and the external threat.

*(1) The Internal Threat.* The main threat of internal attackers is that the ASs and MNs could be malicious. That is to say, the ASs and MNs could be compromised and controlled by adversaries. This model is rational in that the number of ASs and MNs is enormous and the ASs and MNs are owned by third-parties. Therefore, it is hard to guarantee that the entire ASs and MNs are reliable. The malicious ASs and MNs could reveal the privacy easily if we take no measures.

*(2) The External Threat.* As the internet is an open channel, the communication link is unsafe and unreliable. The external threat mainly contains two parts.

*Part 1.* The adversary may eavesdrop on communications between MNs and AS.

*Part 2.* The adversary may attempt to insert some false data to the procedure of data aggregation. That is, the adversary may damage the data integrity.

*4.3. Design Goal.* Our design goal in our paper is satisfying the following requirements.

(i) As the aggregation server and nodes could be compromised and controlled by adversaries, we should prevent the sensed data of an individual node from being disclosed to the aggregation server and nodes.

(ii) As the communication link is unsafe and unreliable, we should prevent the sensed data being leaked. Moreover, we must guarantee the integrity of sensed data.

(iii) As the nodes could become offline or a message could be lost before reaching the aggregation server, we should ensure that these problems could not affect the correctness of the aggregate computed.

## 5. Scheme Design

In this section, we describe our PDA scheme in detail. As we focus on the privacy-preserving robust data aggregation of people-centric sensing networks, we should support a wide range of statistical additive and nonadditive aggregation functions safely. The additive and nonadditive aggregation functions include Sum, Subtraction, Average, Count, Max/Min and Median. We will use Sum as an example to declare the scheme's realization. That is, we should prevent the sensed data of an individual node from being disclosed to the AS and other MNs during the process of computing. Moreover, we should be robust to node or communication failure. We could get the issue model as follow.

*Issue Model.* There are $n$ MNs $\{P_1, P_2, \ldots, P_n\}$. Each of them has a sensed data $x_i$, $i \in [1, n]$. These data should be computed (such as Sum). And the result should be sent to the AS. However, these MNs do not want to leak anything about their data to the other MNs and AS. Through the analysis of the problem, we know that the above problem can be solved as long as we can ensure the security of two-party computing problem. Therefore, the above model can be simplified as follow. Suppose that there are two MNs (say, Alice and Bob); each of them has a sensed data (say, data_Alice and data_Bob). And one or more MNs (called third-party) $M = \{M_1, M_2, \ldots, M_n\}$ exist in the model. And these third-parties are untrusted. Alice and Bob send their sensed data to the AS though these third-parties. Then the AS computes the sum of $A$ and $B$. At the same time, $A$ and $B$ could not be disclosed to the AS and the third-parties. Moreover, we should be robust to node or communication failure. At last the result should be sent to the server.

In order to solve the issue, we present a novel scheme, PDA. The whole idea of scheme is shown in Figure 2. In order to understand the scheme intuitional, we will introduce an instance. In the instance, Alice and Bob have sensed data for computing. We assume data_Alice = 3, data_Bob = 4.

There are five steps to realize our scheme.

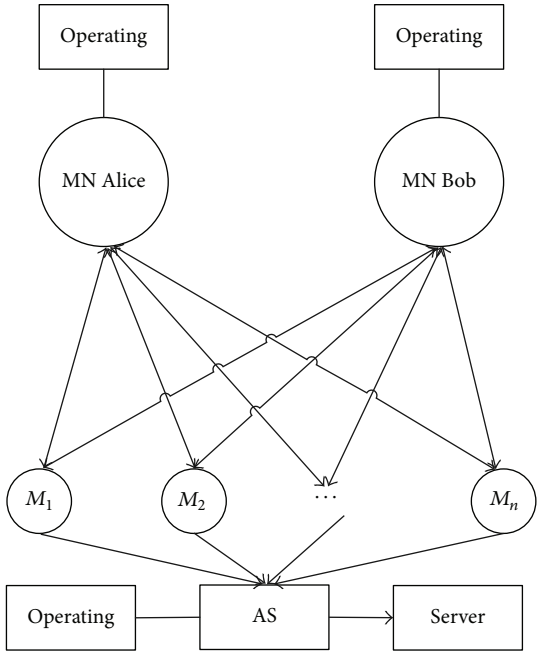



Figure 2: The idea of PDA.

*5.1. Service Registration.* Before the aggregation process of data, each MN needs to register for the service at our scheme. The service registration contains initialization and MN's registration.

During the initialization, the server should periodically generate fake IDs for $K$-anonymity and store them in a fake ID pool. The fake IDs can be generated using a cryptographic hash function, such as SHA-1. The equation is given as follows: fake $\mathrm{ID}_i = \mathrm{SHA}(\text{fake } \mathrm{ID}_{i-1} \oplus \text{salt})$.

During MN's registration, the MN should send a requested message to the server. After receiving the request, the server verifies if the MN is legal or not. After authentication, the server should pick $k$ fake IDs from its fake ID pool. And one of them is used to replace the identity of the MN. Let this fake ID be RID. The other $k-1$ fake IDs are used to confuse the adversaries. Therefore, the server stores an entry as $(\mathrm{RID}, \mathrm{ID}_1, \mathrm{ID}_2, \ldots, \mathrm{ID}_{k-1})$ in a fake ID table. The number of fake IDs ($k$) depends on the MN's reporting frequency. At last, the server sends an OK message which contains the fake ID entry to the MN. The function of fake ID is realizing the anonymous transmission which is based on $K$-anonymity.

*5.2. Authentication.* As the feature of people-centric sensing network, the MNs are always moving from an AS to another one. To let the AS authenticate their identity, the MNs should send an authentication request to the AS. The request contains the MN's ID table. After receiving the authentication request, the AS forwards this request to the server. Upon receiving the message, the MS should verify the ID table's legality. If the verification succeeds, it sends a reply to the AS. On the reception of the reply, the AS sends an OK message to the MN which contains some parameters. These parameters contain a prime number $P$ for homomorphic encryption, a prime number $Q$, and a random number $m$ for secret sharing. The meaning of these parameters will be recommended in later subsection. After authentication, the communication link is established between the MN and the AS. Then the MNs could send the sensed data to the AS after receiving a task.

In our instance, we take $P = 11$, $Q = 19$, and $m = 3$.

*5.3. Homomorphic Encryption.* After the MNs receive a task from the server, they should send the sensed data to the AS. Before sending data, the MNs will execute the data processing, homomorphic encryption. In this paper, the important point is proposing PDA, so we will use a simple algorithm to achieve homomorphic encryption. The implementation process is shown as follows. A large prime $P$ getting from the AS is used as the key. $a$ and $b$ are sensed data. The encryption algorithm is $A = a + (r * P)$, $r$ is a random integer number, $A$ is the ciphertext, and $a$ is the plaintext. Therefore, the data of $a$ and $b$ which have been encrypted can be expressed as $(A + B) = (a + (r_1 * P)) + (b + (r_2 * P))$. Then, the process of decryption is $P$ modulo $(A + B)$ and the result is $(a + b)$ [26, 27]. So we achieve a simple homomorphic encryption algorithm. In PDA, both participants should operate their own data using the homomorphic encryption algorithm before sending to the AS. Though the operation, the AS get the sensed data which was encrypted. However, it could compute the sum of the sensed data. That is, the aggregation functions (such as Sum) could be completed without leaking individual sensed data.

In our instance, Alice and Bob use homomorphic encryption algorithm to encrypt data_Alice and data_Bob according to the prime number $P$ which is negotiated. Then, we could get the ciphertext. For Alice, we take an integer $r = 3$; ciphertext $A$ is $A = a + (r * P) = 3 + (3 * 11) = 36$. For Bob, we take an integer $r = 2$; ciphertext $B$ is $B = 4 + (2 * 11) = 26$.

*5.4. Secret Sharing.* As the MNs could become offline or a message could be lost before reaching the AS, we should ensure that these problems could not affect the correctness of the aggregate computed. In our paper, we could achieve this goal though secret sharing.

The general type of secret sharing is threshold method, generally with $(m, n)$-threshold method. In this type, the dealer accomplishes this by giving each participants a share in such a way that any group of $m$ or more participants can together reconstruct the secret but no group of fewer than $m$ players can. In this scheme, in order to send the data after encryption to some third-parties and process the data by the third-party, we will improve the Shamir secret sharing algorithm. Shamir secret sharing algorithm is proposed by Shamir in 1979 [28]. The system relies on the idea that you can fit a unique polynomial of degree $m-1$ to any set of $m$ points that lie on the polynomial. It takes two points to define a straight line, three points to fully define a quadratic, four points to define a cubic curve, and so on. That is, it takes $m$ points to define a polynomial of degree $m-1$. The principle of the algorithm is shown in Figure 3. Any two points in a straight line can define a straight line. So we can take any

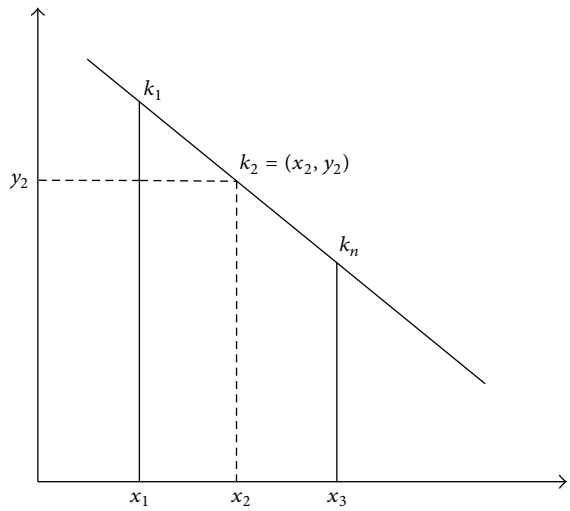



Figure 3: The principle of Shamir secret sharing.

of two points in $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ to obtain the equation of the line [29].

The procedure of the algorithm is (assuming the secret is $K$) as follows.

*Selecting a Polynomial.* At first, we should select a polynomial $F(x)$ based on a prime number $Q$. The prime number is got from the parameter negotiation. $F(x) = (a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + K) \bmod Q$, and $a_{m-1}, a_{m-2}, \ldots, a_1 \in [1, Q]$.

*Getting the Variables.* Then, we should select $n$ groups data $x_1, x_2, \ldots, x_n, x_i \neq 0$.

*Computing the Point.* At last, we should compute the point $(x_i, y_i)$, which meets the polynomial.

Therefore, we split the secret $K$ to $(x_i, y_i)$, $i \in [1, n]$. We can reconstruct the secret when getting more than $m$ parts.

After homomorphic encryption, the MN will operate the data which was encrypted by homomorphic encryption through secret sharing. That is, the MN will slice the data to $t$ points. Then, the MN will send these points to $t$ third-parties. At the same time, the MN should pick an ID from its fake ID table and send it to the third-parties.

In our instance, Alice and Bob will perform secret sharing operation for the ciphertext. We assume that the polynomial $F(x)$ for Alice is $F(x) = (2x^2 + 4x + 36) \bmod 19$. At first, Alice will compute the point of polynomial, based on $(x_1, x_2, x_3, x_4)$. Then, Alice could get four points $\{(1, 4), (2, 14), (3, 9), (4, 8)\}$. At last, Alice will send the four points to four third-parties. Similarly, we assume that the polynomial $F(x)$ for Bob is $F(x) = (x^2 + 2x + 26) \bmod 19$. Bob will compute the point of polynomial, $\{(1, 10), (2, 15), (3, 3), (4, 12)\}$ [30]. At last, Bob will send the four points to four third-parties. At the same time, Alice and Bob should send its fake ID to the third-parties.

*5.5. AS Operation.* After the third-parties receive the points and the ID, they should forward the information to the AS. Upon receiving the points and the ID, the AS should take three steps to get the sum of the sensed data.

*Step 1* (recovering the data). The AS picks the points which have the same ID as a group. And it recovers the data for each group based on secret sharing. The procedure of recovering is as follow. At first, the AS should pick $m$ points and select a polynomial $F(x) = (a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + K) \bmod Q$. Then, the AS substitutes the $m$ points into $F(x)$. So we can get the data of $(a_{m-1}, a_{m-2}, ..., K)$. At last, we could get the data when setting $x = 0$. In our instance, we pick three points $\{(1, 4), (2, 14), (3, 9)\}$ and substitute them into $F(x)$ for Alice. Then, we could get that $a_2 = 2$, $a_1 = 4$, and $K = 36$. So the data is 36 for Alice. Similarly, the data is 26 for Bob.

*Step 2* (computing the sum of data). After recovering the data, we could compute the sum of data. In our instance, the sum of data is $R = 36 + 26 = 62$.

*Step 3* (getting the result). As we conduct the data using homomorphic encryption, the data after computing is not the final result. In order to get the result, we should take the process of decryption which is using $P$ modulo of the data. In our instance, the result is $R\%P = 62\%11 = 7$. As data_Alice + data_Bob $= 3 + 4 = 7$, we know that the result is correct. At last, the result will be sent to the server. The server can provide data to some applications which provide guidance of the social environment or people's social activities.

Through the above analysis, we compute the sum of sensed data during the procedure of aggregation. Similarly, we could support a wide range of statistical additive and nonadditive aggregation functions [12].

## 6. Evaluation

In this section, we evaluate the security and performance of PDA.

*6.1. Security Evaluation.* In this part, we analyze the security of our scheme based on the threat model discussed in Section 3. We assume that the threats contain that (1) the ASs could be compromised by adversaries; (2) the MNs could be malicious; (3) the adversary could eavesdrop all of the communications in the network; (4) the adversary could attempt to insert some false data to the procedure of data aggregation.

For ease of threat, we explain the security features of our scheme.

(i) The ASs could be compromised by adversaries. In PDA, the AS only gets the sensed data which has been encrypted by homomorphic encryption algorithm. As the AS cannot get the parameter $r$, it could not decrypt the data. Moreover, the AS cannot get the MNs which own the sensed data real ID as fake IDs. Therefore, the attacker obtains no useful information from the AS to compromise a single node's privacy.

Table 2: Comparing the features with other schemes.

| | Additive and nonadditive aggregation | Privacy versus outsiders | Privacy versus MNs | Privacy versus AS | Node-failure and data-loss resilience | Suitable for people-centric sensing system |
|---|---|---|---|---|---|---|
| PriSense [12] | √ | √ | √ | √ | × | √ |
| AnonySense [11] | × | √ | √ | × | × | √ |
| Conti et al. scheme [17] | √ | √ | √ | √ | √ | × |
| Our scheme PDA | √ | √ | √ | √ | √ | √ |

(ii) The MNs could be malicious. In PDA, the MN acts as one of third-part in the procedure of data aggregation. Each MN can only get one part of the other MNs' sensed data based on secret sharing. And fewer than $m$ players cannot reconstruct the data. Moreover the sensed data is encrypted by homomorphic encryption algorithm. Therefore, we can compromise the privacy of a non-captured node leveraging the information.

(iii) The adversary could eavesdrop all of the communications in the network. In PDA, each link between AS and MN transfers only one part of sensed data based on secret sharing. And fewer than $m$ players cannot reconstruct the data. If the MN eavesdropped all of the communications in the network, it can reconstruct the sensed data. However, the sensed data is encrypted by homomorphic encryption algorithm. As the adversary cannot get the parameter $r$, it could not decrypt the data. Therefore, the adversary obtains no useful information to compromise a single node's privacy.

(iv) The adversary could attempt to insert some false data to the procedure of data aggregation. In PDA, each sensed data is attached with the owner's ID. And the ID is picked from the MN's fake ID table randomly. As the adversary cannot get the fake ID table, he does not know the valid ID. The false data which was sent by the adversary is not accepted by the AS. Therefore, the adversary cannot damage the data integrity.

Through the above analysis, PDA is able to meet the security requirements of privacy-preserving data aggregation.

*6.2. Performance Evaluation.* In this section, we focus on the efficiency of PDA from robustness, computational complexity, and communication complexity.

*6.2.1. Robustness.* As the MNs could become offline or a message could be lost before reaching the AS, we should ensure that these problems could not affect the correctness of the aggregate computed. In our paper, we could achieve this goal though secret sharing. As the AS get a share from $n$ third-parties, it can reconstruct the data but no group of fewer than $m$ ($m < n$) can. That is, the AS can reconstruct the sensed data when it received no less than $m$ parts. Therefore, if the number of node failure or data loss is less than $n-m$, we could ensure the correctness of the aggregate computed. Though the analysis, the AS can robust to node or communication failure.

*6.2.2. Computational Complexity.* For the people-centric sensing system, it is assumed that there are $n$ MNs for computing. In order to compute all of the MNs' data, we should compute each two MNs' data. Therefore, the computing complexity of this process is $O(n^2)$. For each computation processing, the computation complexity is mainly determined by homomorphic encryption and secret sharing. The operations of homomorphic encryption and secret sharing are the simple algebraic operations. So the computation complexity is $O(k)$ that is not relevant to the size of input. The analysis shows that computational complexity of PDA is $O(n^2)$.

*6.2.3. Communication Complexity.* The primary aspect to measure the performance of a scheme is its computational complexity. But, for the problem of privacy-preserving robust data aggregation, computational complexity cannot fully describe the performance of a scheme. As the participants and third-parties will communicate each other, the communication complexity is also an important aspect to measure the performance of PDA.

The communication complexity contains the cost of secret sharing and the third-party operation. It is assumed that the number of MNs is $m$ and the number of third-parties is $n$. As the MNs will send the secret input data to the third-parties and the third-parties will send the result to the AS, the communication complexity of PDA is $O(m * n)$.

*6.3. Comparison.* In this section, we summarize the features of our proposal compared with other mainly relevant algorithms in Table 2.

The feature additive and nonadditive aggregation indicates if the scheme support a wide range of statistical additive and nonadditive aggregation functions such as Sum, Subtraction, Average, Count, Max/Min, and Median. In columns 2, 3, and 4, these features indicate if the scheme protects privacy against outside eavesdropper, other MNs, or the AS. The features of node failure and data loss resilience refer to whether the AS can compute the correct aggregate when a few MNs become offline or a few messages could be lost. In columns 6 and 7, these features denote the scheme's computational and communication complexity. The last feature indicates if the scheme is suitable for the people-centric sensing networks.

Through the above analysis, our scheme is secure, scalable, and resilient to node failure and data loss.

## 7. Conclusion

We present PDA, a novel privacy-preserving robust data aggregation scheme in people-centric sensing system. Based on $K$-anonymity, homomorphic encryption, and secret sharing, PDA can support a wide range of statistical additive and nonadditive aggregation functions such as sum, subtraction, average, count, max/min and median without leaking individual sensed data. Moreover, PDA is robust to node failure and data loss. We also evaluate the efficacy and efficiency of PDA. The result shows that our scheme can achieve the security and robust goal under a reasonable cost. We believe a privacy-aware system will make people-centric sensing networks more acceptable.

## Acknowledgments

The work described in this paper is partially supported by the Grants of the National Basic Research Program of China (973 project) under Grants nos. 2009CB320503, 2012CB315906; the project of National Science Foundation of China under Grants nos. 61070199, 61103189, 61103194, 61103182, 61202488, and 61272482; the National High Technology Research and Development Program of China (863 Program) nos. 2011AA01A103, 2012AA01A506, and 2013AA013505; the Research Fund for the Doctoral Program of Higher Education of China under Grants nos. 20114307110006, 20124307120032; the program for Changjiang Scholars and Innovative Research Team in University (no. IRT1012); Science and Technology Innovative Research Team in Higher Educational Institutions of Hunan Province (network technology); and Hunan Province Natural Science Foundation of China (11JJ7003).

## References

[1] T. Abdelzaher, Y. Anokwa, P. Boda et al., "Mobiscopes for human spaces," *IEEE Pervasive Computing*, vol. 6, no. 2, pp. 20–29, 2007.

[2] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-centric urban sensing," in *Proceedings of the 2nd Annual International Workshop on Wireless Internet (WICON '06)*, ACM, 2006.

[3] A. T. Campbell, S. B. Eisenman, N. D. Lane et al., "The rise of people-centric sensing," *IEEE Internet Computing*, vol. 12, no. 4, pp. 12–21, 2008.

[4] I. Krontiris, F. C. Freiling, and T. Dimitriou, "Location privacy in urban sensing networks: research challenges and directions," *IEEE Wireless Communications*, vol. 17, no. 5, pp. 30–35, 2010.

[5] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "BikeNet: a mobile sensing system for cyclist experience mapping," *ACM Transactions on Sensor Networks*, vol. 6, no. 1, article 6, 2009.

[6] M. Welsh and J. Bers, "CitySense: an urban-scale sensor network," in *Ecological Urbanism*, M. Mostafavi and G. Doherty, Eds., Harvard Graduate School of Design, pp. 164–165, Lars Müller, Zurich, Switzerland, 2010.

[7] Y. Malinovskiy and Y. Wang, "Pedestrian travel pattern discovery using mobile bluetooth sensors," in *Proceedings of the 91st Annual Meeting of the Transportation Research Board*, 2012.

[8] M. Mun, S. Reddy, K. Shilton et al., "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'09)*, pp. 55–68, ACM, Wroclaw, Poland, June 2009.

[9] "Microsoft Research Sense Web Project," March 2013, http://research.microsoft.com/en-us/projects/senseweb/.

[10] R. Crepaldi, R. Beavers, B. Ehrat, and R. Kravets, "Illinois vehicular project, live data sampling and opportunistic internet connectivity," in *Proceedings of the 3rd ACM International Workshop on Mobile Opportunistic Networks (MobiOpp '12)*, pp. 85–86, ACM, Zurich, Switzerland, March 2012.

[11] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "AnonySense: privacy-aware people-centric sensing," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (Mobisys '08)*, pp. 211–224, ACM, Breckenridge, Colo, USA, June 2008.

[12] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: privacy-preserving data aggregation in people-centric urban sensing systems," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '10)*, San Diego, Calif, USA, March 2010.

[13] T. Feng, C. Wang, W. Zhang, and L. Ruan, "Confidentiality protection for distributed sensor data aggregation," in *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM '08)*, pp. 475–483, Phoenix, Ariz, USA, April 2008.

[14] W. Zhang, C. Wang, and T. Feng, "GP2S: generic privacy-preservation solutions for approximate aggregation of sensor data," in *Proceedings of the 6th IEEE Annual International Conference on Pervasive Computing and Communications (PerCom '08)*, pp. 179–184, Hong Kong, March 2008.

[15] D. Wagner, "Resilient aggregation in sensor networks," in *Proceedings of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '06)*, pp. 71–82, Alexandria, Va, USA, October 2006.

[16] H. Chan, A. Perrig, and D. Song, "Secure hierarchical in-network aggregation in sensor networks," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 278–287, ACM, Alexandria, Va, USA, November 2006.

[17] M. Conti, L. Zhang, S. Roy, R. di Pietro, S. Jajodia, and L. V. Mancini, "Privacy-preserving robust data aggregation in wireless sensor networks," *Security and Communication Networks*, vol. 2, no. 2, pp. 195–213, 2009.

[18] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," Tech. Rep., SRI International, 1998.

[19] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[20] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," *Foundations of Secure Computation*, vol. 32, no. 4, pp. 169–178, 1978.

[21] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Advances in Cryptology—EUROCRYPT, 2010*, H. Gilbert, Ed., vol. 6110 of *Lecture Notes in Computer Science*, pp. 24–43, Springer, Berlin, Germany, 2010.

[22] C. Fontaine and F. Galand, "A survey of homomorphic encryption for nonspecialists," *Eurasip Journal on Information Security*, vol. 2007, Article ID 13801, 2007.

[23] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *Proceedings of the 28th IEEE Annual Symposium on Foundations of Computer Science (FOCS '87)*, 1987.

[24] L. Harn, "Secure secret reconstruction and multi—secret sharing schemes with unconditional security," *Security and Communication Networks*, 2013.

[25] Y. Yan and H. Hu, "Research and realization of security electronic voting plan based on homomorphic commitment verifiable secret sharing," *Applied Mechanics and Materials*, vol. 263–266, pp. 1673–1676, 2013.

[26] E. Prouff and T. Roche, "Higher-order glitches free implementation of the AES using secure multi-party computation protocols," in *Cryptographic Hardware and Embedded Systems—CHES*, B. Preneel and T. Takagi, Eds., vol. 6917 of *Lecture Notes in Computer Science*, pp. 63–78, Springer, Berlin, Germany, 2011.

[27] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Advances in Cryptology—CRYPTO, 2010*, T. Rabin, Ed., vol. 6223 of *Lecture Notes in Computer Science*, pp. 483–501, Springer, Berlin, Germany, 2010.

[28] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[29] V. A. Oleshchuk and Z. Vladimir, "Secure multi-party computations and privacy preservation: results and open problems," *Telektronikk*, vol. 103, no. 2, article 20, 2007.

[30] S. L. From and T. Jakobsen, *Secure multi-party computation on integers [M.S. dissertation]*, Datalogisk Institut, Aarhus Universitet, Aarhus, Denmark, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 165498, 15 pages
http://dx.doi.org/10.1155/2013/165498

*Research Article*

# Distributed Voronoi-Based Self-Redeployment for Coverage Enhancement in a Mobile Directional Sensor Network

**Tien-Wen Sung and Chu-Sing Yang**

*Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, No.1, University Road, Tainan City 701, Taiwan*

Correspondence should be addressed to Tien-Wen Sung; tienwen.sung@gmail.com

Received 28 June 2013; Revised 16 September 2013; Accepted 17 September 2013

Academic Editor: Shengming Jiang



A directional sensor network is different from conventional wireless sensor networks. It uses directional sensors instead of omnidirectional ones in the network for different applications, and the effective sensing range is characterized by directionality and size-specific sensing angle. Therefore, conditions of directional sensor networks are dissimilar to those of generic wireless sensor networks for researches, especially on the sensing coverage. This study proposed a distributed approach to enhance the overall field coverage by utilizing mobile and direction-rotatable sensors in a directional sensor network. The algorithm makes sensors self-redeploy to the new location and new direction without global information by utilizing the features of geometrical Voronoi cells. Simulations were used to evaluate and prove the effectiveness of the proposed algorithm. The results show that the approach contributes to significant field coverage improvement in directional sensor networks.

## 1. Introduction

In a wireless sensor network (WSN) [1], the sensing coverage is always one of the key factors to ensure that sensing tasks will be well performed. The coverage ratio becomes a fundamental index of the measurement for WSN quality of service (QoS) [2]. Many studies related to the subject of WSN coverage have been successively proposed [3]. In recent years, directional sensor networks (DSNs) have drawn the attention of researchers. Differing from the conventional WSNs which use omnidirectional scalar sensors such as temperature or humidity sensors, a DSN is likely to be equipped with directional sensors such as image or video sensors. There are different problems and conditions regarding the coverage issues should be considered in the researches for DSNs due to the limited effective sensing range characterized by directionality and size-specific sensing angle [4, 5]. In the sensing field of a DSN, the sensing coverage depends on not only the location but also on the sensing direction and sensing angle of each sensor [6].

Although there are some studies concerning coverage of DSN [7], the problems and solutions are different. For example, several of the studies focused on the target coverage and several focused on the overall field coverage; several aimed at solving $k$-coverage problem and several aimed at solving prioritized region coverage. For another example, several studies considered the obstacles in a sensing field and several considered an obstacle-free sensing environment. This paper proposes a distributed self-redeployed algorithm to deal with the enhancement of overall coverage ratio in the sensing field of a DSN which consists of mobile and rotatable directional sensors. The characteristics of geometrical Voronoi diagram were utilized to determine the target positions and sensing directions of sensors and reduce the decision-making complexity. Using Voronoi diagram in this study was motivated by the following advantages: (1) Voronoi diagram exactly can be generated from a set of sensors and each sensor is associated with only one Voronoi cell; (2) the cell-based structure can help design a localized/distributed algorithm without global information; (3) each sensor can only check its own cell to examine the coverage hole; (4) both Voronoi vertices and edges can help make the decision of sensor position and sensing direction; (5) for a mobile sensor, the moving distance can be confined to the cell;

and (6) the construction of a Voronoi cell is independent of sensing radii and angles; this helps a distributed algorithm be applicable to those sensors which have varied radii and angles. More detailed introduction of Voronoi diagram is described in Section 3.2.

The rest of the paper is organized as follows. Section 2 briefs the previous works related to DSN coverage. Section 3 describes the preconditions and assumptions. In Section 4, we provide the readers with our proposed scheme. Section 5 evaluates and analyzes the efficiency of the scheme. Finally, we conclude this paper in Section 6.

## 2. Related Works

There are four major categories of research subjects in WSNs [8]: (1) sensing coverage, (2) network connectivity, (3) network longevity, and (4) data fidelity. These subjects are also the fundamental requirements in DSN, but the existing WSN solutions or results are not completely suitable for DSNs, particularly the coverage issue. This is mainly due to that the directional sensors are characterized by working direction, angle of view (AOV), and line of sight (LOS) [9], and new solutions for the dissimilar problems are required [10].

In regard to DSN coverage problems, the related studies can be divided into two categories [11]: (1) known targets coverage and (2) region coverage. The former determines a subset of the sensors to attain coverage on the specific targets or positions, and the latter makes the sensors attain a full or certain ratio of sensing coverage in the task region. Concerning known-targets coverage, Chow et al. [12] proposed an algorithm to select multiple directional sensors for providing 360° complete monitoring of a target while one sensor can only cover a part of the target. Wang et al. [13] aimed at prioritized targets and used the genetic algorithm to obtain a minimum subset of directional sensors for covering all of the targets. Hsu et al. [14] proposed an algorithm to solve the problem of which the number of covered targets is maximized whereas the rotated angle of sensors is minimized. And concerning region coverage, Jing and Jian-Chao [15] used the coverage overlap between the adjacent sensors as the quantity of electric charge and then used Coulomb's inverse-square law to change the sensor direction for reducing the overlapped coverage. Liang et al. [16] aimed at DSNs with mobile sensors, so that sensors could move to appropriate positions and obtain a good total coverage. Guvensan and Yavuz [17] used a hybrid solution of mixing stationary sensors, motile sensors, and mobile sensors in a DSN to increase the sensing field coverage. Tezcan and Wang [18] considered the condition of obstacles within the sensing field and proposed an algorithm utilizing rotatable sensors to reduce the influence of obstacles on the coverage. Huang et al. [19] focused on the multimedia image sensor networks and proposed a virtual potential field-based method with the considerations of sensor direction and movement for the coverage enhancement. The virtual force causes the adjustment of angular magnitude to be a trouble in coverage problem; therefore, linear-relation-based algorithm (LRBA) and mechanism-based approximate algorithm (MBAA) was proposed. These algorithms need the process of pairing between two adjacent nodes and a defect exists in the algorithms. That is, some nodes could not find their respective paired partner. These isolated sensors cannot perform the algorithms for the coverage contribution. Ma and Liu [20] analyzed deployment strategies and proposed a group-based strategy, namely, grouping scheduling protocol (GSP) for satisfying given coverage probability requirement in a directional sensor network. It needs repairing processes if certain grouped sensors are incommunicable to the sink. This needs the deployment of more sensors; therefore, the average coverage ratio of the grouped sensors could be decreased.

Voronoi-based method has been utilized in WSNs, but it has not drawn much attention of researchers on the DSN coverage problems. Li et al. [21] proposed the Voronoi-based distributed approximation (VDA) algorithm to make sensors cover the Voronoi edges as more as possible. The study approximately considers that if most Voronoi edges are covered, then most area will be covered; however, this is not definite and may cause more coverage overlap. This paper proposes a new distributed approach to determine location and direction movements of the mobile and rotational sensors for obtaining significant coverage contribution in the DSN. In addition, the geometrical features of Voronoi cells with its vertices, edges, and included angles were utilized in the proposed approach to assist in the determination of sensor self-redeployment.

## 3. Preliminaries

*3.1. Assumptions.* There are several related studies [16, 21, 22] assume that all the directional sensors have the same sensing radius and sensing angle range, which means that the sensors are homogeneous. The algorithm proposed in this paper doses not have this limitation. It is applicable to sensors with different sensing radii and angles. But the general assumptions are listed as follows:

(1) Each sensor is well aware of its coordinate by utilizing a certain localization technology [23].

(2) Each sensor has enough communication range or multihop transmission capacity to transmit information to neighbor sensors.

(3) Sensors are rotatable; they can do a clockwise or counterclockwise rotation to change the working direction.

(4) Sensors are mobile; they can move within the sensing field. This assumption is not unrealistic in the real world [24, 25].

*3.2. Voronoi Diagram.* The Voronoi diagram is a computational geometry data structure with special characteristics [26], which is applicable to be utilized in the proposed algorithm to divide the sensing field into cells. The sensing field will be divided into Voronoi cells according to the initial positions of the deployed sensors, as shown in Figure 1. Given a set of $n$ sensors $s_1, s_2, \ldots, s_n$ in the sensing field, the one and only one (unique) Voronoi diagram can be constructed

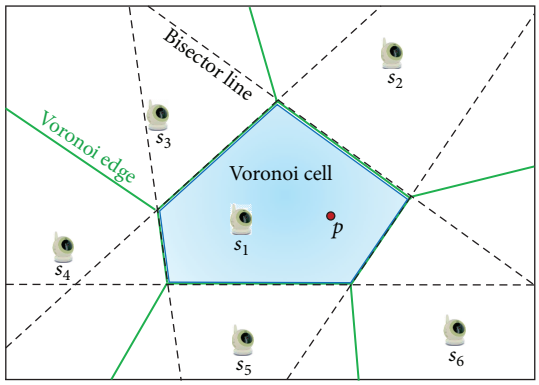



Figure 1: Construction of the Voronoi cells.

by drawing the perpendicular bisector of line segment of each sensor pair. Those bisector line segments form the boundaries of Voronoi cells and are called Voronoi edges. The endpoints of these edges are called Voronoi vertices. Finally, the sensing field is divided into $n$ Voronoi cells, which meets the following two major properties.

(1) Each sensor $s_i$ lies in exactly one cell.

(2) If a point $p$ lies in the same cell as $s_i$, then the Euclidian distance from $p$ to $s_i$ will be shorter than the one from $p$ to $s_j$, where $s_j$ is any other sensor.

Regarding the construction of a Voronoi Diagram, there are several algorithms to generate the diagram from a set of points, such as Bowyer-Watson algorithm with an $O(n^2)$ time complexity and the best known Fortune algorithm [27] with an $O(n \log n)$ time complexity. They can used to construct a global Voronoi diagram. In this study, we do not need to construct a global Voronoi diagram. Our proposed algorithm is a distributed method and each sensor only needs to construct one cell of the Voronoi diagram. The Voronoi diagram of a set of points (sensors) is unique (one and only one) [28], which consists of Voronoi cells. In our study, each distributed sensor broadcasts its coordinates as local information to the neighbors. A sensor constructs a bisector line between itself and the neighbor whenever it receives the coordinates of the neighbor. And finally, the sensor will be in a convex polygon, in which the polygon consists of certain line segments of those bisector lines. And the polygon is called a Voronoi cell associated with the sensor. No other sensors will appear in the cell. Any edge of the cell is a bisector line between the associated sensor and its one neighbor, and this edge is also an edge of the cell of that neighbor. It should be noted that a Voronoi diagram consists of the Voronoi cells associated with a given set of points is unique, which had been proven in [28].

*3.3. Directional Sensing Model.* Figure 2 shows the direction-rotatable sensing model for the directional sensors in the proposed algorithm of this paper. The notations and parameters are listed in Table 1.

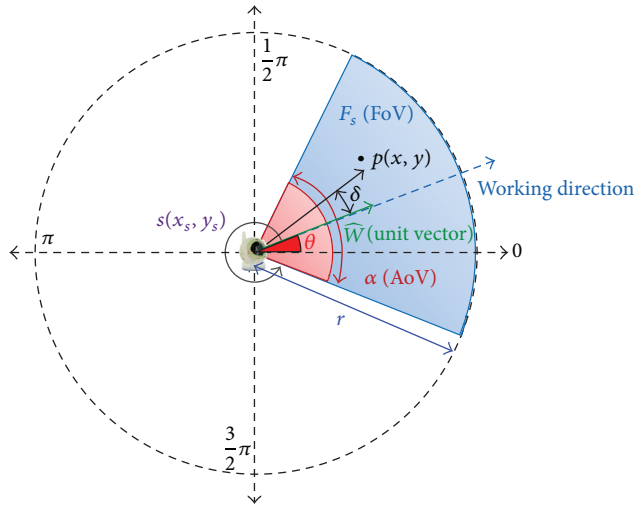



Figure 2: Sensing model for direction-rotatable sensors.

Table 1: Notations and parameters for the directional sensing model.

| Notation/parameter | Description |
|---|---|
| $s(x_s, y_s)$ | The directional sensor with its coordinate |
| $F_s$ | The effective sensing field of $s$; the region covered by $s$; field of view (FoV) |
| $\alpha$ | The angular size of sensing range of $s$; angle of view (AoV) |
| $r$ | The sensing radius of $s$ |
| $\theta$ | The working direction of $s$, defined as the angle value relative to the positive $x$-axis of the sensing direction vector; $\theta$ is between 0 and $\pm\pi$. |
| $\widehat{w}$ | The unit vector of the working direction |
| $p(x, y)$ | A point with its coordinate covered by $s$ |
| $\delta$ | The included angle between $\vec{sp}$ and $\widehat{w}$ |

The effective sensing field $F_s$ is in sector shape. A point $p$ is said to be covered by the sensor $s$ if and only if the following two conditions are satisfied.

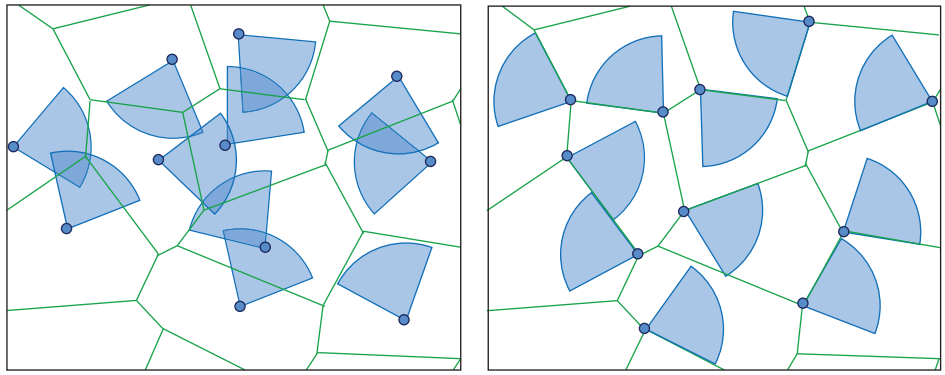
(1) The Euclidean distance between $p(x, y)$ and $s(x_s, y_s)$ is less than or equal to the sensing radius ($r$) of $s$. Consider

$$\sqrt{(x - x_s)^2 + (y - y_s)^2} \leq r. \tag{1}$$

(2) The included angle ($\delta$) between $\vec{sp}$ and the working direction unit vector ($\widehat{w}$) is less than or equal to the half of the AoV ($\alpha$). Consider

$$\vec{sp} \cdot \widehat{w} = \left\|\vec{sp}\right\| \|\widehat{w}\| \cos\delta \tag{2}$$

$$\Rightarrow (x - x_s)\cos\theta + (y - y_s)\sin\theta \geq \sqrt{(x - x_s)^2 + (y - y_s)^2}\cos\left(\frac{\alpha}{2}\right). \tag{3}$$



(a) Initial coverage (b) Enhanced coverage

Figure 3: A schematic example of results before and after execution of the algorithm.

In brief, the directional sensing model shown in Figure 2 illustrates that a point $p(x, y)$ is covered by the sensor $s$; that is, to say, $(x, y) \in F_s$, if and only if (1) and (3) are satisfied with the given parameters $r$, $\theta$, $\alpha$, and sensor coordinate $(x_s, y_s)$.

## 4. Distributed Voronoi-Based Self-Redeployment Algorithm

*4.1. Problem Description.* In a sensing field on which directional sensors were numerously and randomly deployed, it is difficult to obtain the optimal field coverage since partial regions are not covered and some regions are overlapped. The chief defect in using centralized algorithms to solve this problem is that the multihop transmissions over the network for collecting global information will be heavy on resources. On the contrary, a distributed algorithm does not need the global information, but only local information used for solutions. In addition, a distributed algorithm is more likely to make a real-time response or quick adjustment in changes of environment. Therefore, this paper proposes a distributed Voronoi-based self-redeployment algorithm, "DVSA" for short, to improve the field coverage ratio for a mobile and rotatable directional sensor network. And, it can be applied to sensors with different sensing radii and AoVs.

Each sensor constructs its own Voronoi cell according to the position information of their neighbor sensors after the initial deployment. By evaluating the suitability of each vertex with the parameters of cell structure and sensing model, one will be selected as the new location of the sensor (associated with the cell), and a new working direction also will be decided. The suitability evaluation aims to reduce the probability of coverage overlap across the cells. Figure 3 shows a schematic example of coverage comparison between the results before and after execution of the algorithm.

*4.2. Target Location of Sensor Movement.* Figure 4 shows an initially deployed directional sensor $s$ and its associated Voronoi cell $C_s$. The sensor coordinate $(x_s, y_s)$ will be known after initial sensor deployment and the sensing parameters $\alpha$ and $r$ are also given. In addition, the coordinates of all vertices of the cell will be obtained after the construction of local

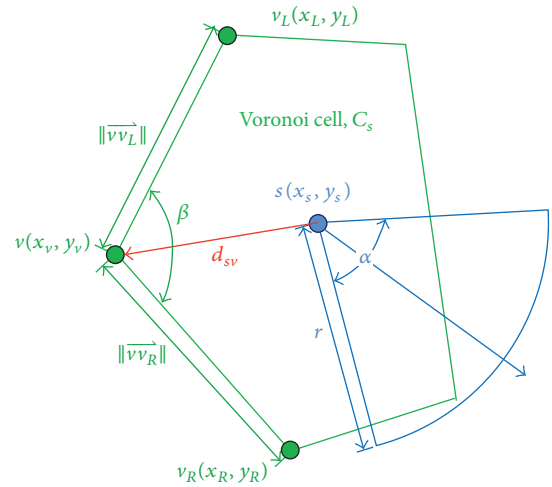



Figure 4: Included angle of a Voronoi vertex and related parameters.

Voronoi cell. Let $V$ be the set of all vertices of $C_s$. For each $v \in V$, $v_L$, and $v_R$ means the left adjacent vertex and right adjacent vertex of $v$, and their coordinates $(x_L, y_L)$ and $(x_R, y_R)$ are known. According to the geometric definition of inner product of two vectors in linear algebra, the included angle of $\overrightarrow{vv_L}$ and $\overrightarrow{vv_R}$, notated as $\beta$, can be derived from

$$\overrightarrow{vv_L} \cdot \overrightarrow{vv_R} = \left\|\overrightarrow{vv_L}\right\| \left\|\overrightarrow{vv_R}\right\| \cos\beta$$

$$\beta = \cos^{-1} \frac{\overrightarrow{vv_L} \cdot \overrightarrow{vv_R}}{\left\|\overrightarrow{vv_L}\right\| \left\|\overrightarrow{vv_R}\right\|}$$

$$= \cos^{-1} \frac{(x_L - x_v)(x_R - x_v) + (y_L - y_v)(y_R - y_v)}{\sqrt{(x_L - x_v)^2 + (y_L - y_v)^2}\sqrt{(x_R - x_v)^2 + (y_R - y_v)^2}}. \quad (4)$$

The desired choice of a target vertex for the sensor movement is the vertex that its corresponding included angle is larger than or equal to the AoV of the sensor and the lengths of its two edges are larger than or equal to the sensing radius of the sensor. If there is more than one choice in this case,

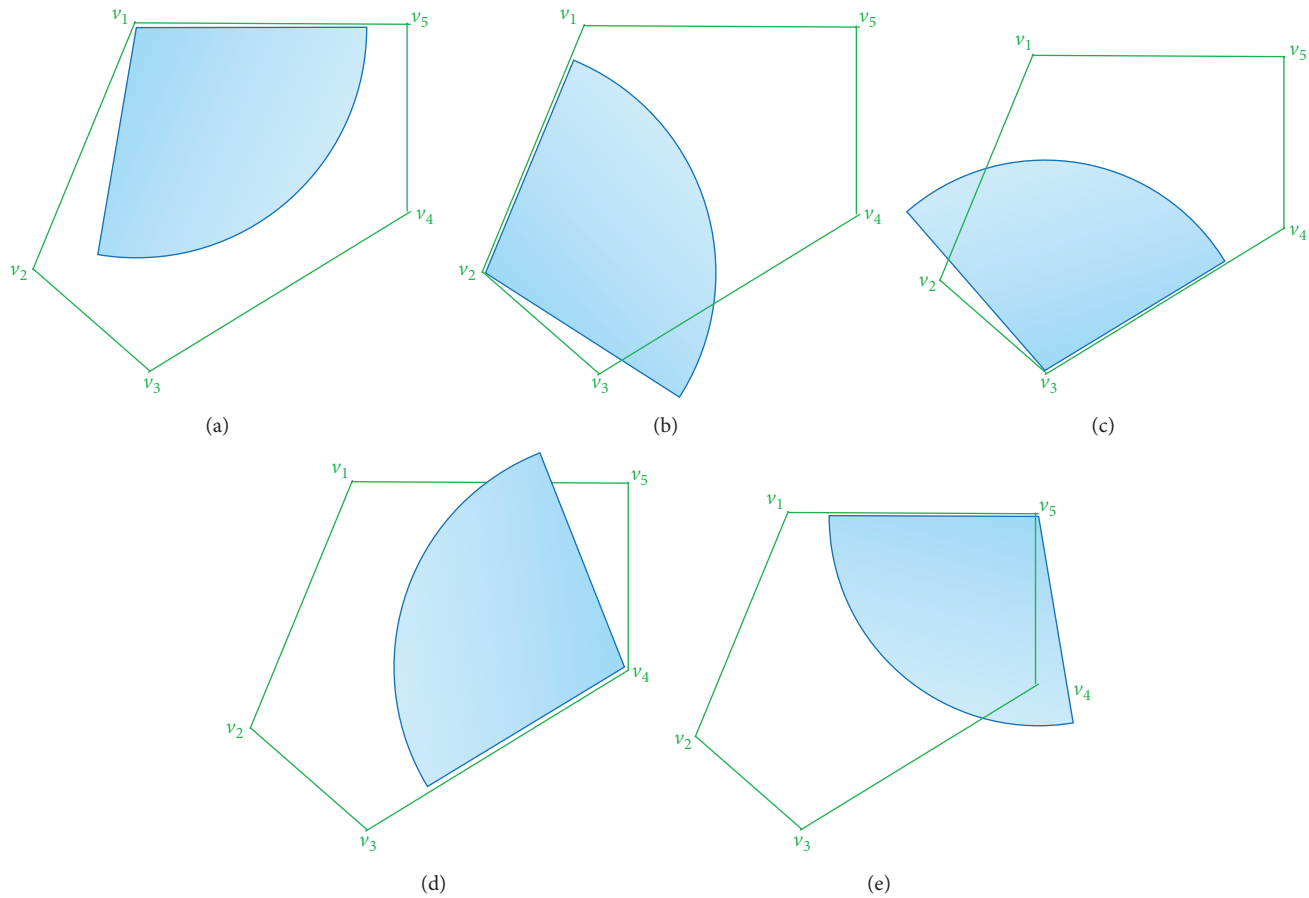



Figure 5: Results of choosing different vertices as the target location.

the vertex with the largest included angle is selected from the candidates, as shown as follows:

$$T = \left\{ v \in V \mid \beta \geq \alpha,\ \left\| \overrightarrow{vv_L} \right\| \geq r,\ \left\| \overrightarrow{vv_R} \right\| \geq r \right\}$$
$$t = \arg\max_{v \in T} \beta, \tag{5}$$

where $T$ is the set of target vertex candidates with the highest priority, and $t$ is the selected target vertex. If $T$ is empty, the second preferred choice of target vertex is the vertex that its included angle is larger than or equal to the AoV of the sensor, but only one edge is larger than or equal to the sensing radius $r$. Similarly, if there is more than one choice in this priority level, the vertex with largest included angle is selected from the new candidates. On the contrary, if there is neither vertex has an included angle $\beta \geq \alpha$ nor an edge larger than or equal to $r$, the one with largest included angleis selected from these lowest-priority candidates. Figure 5 shows an example with the results of choosing different vertices as the target location. The vertex $v_1$ will be selected as the new location of the sensor. The moving distance of the sensor for moving to the target vertex $v$ is as (6), and this movement will certainly be within the limitation of cell boundary. Consider

$$d_{sv} = \sqrt{(x_s - x_v)^2 + (y_s - y_v)^2}. \tag{6}$$

In this phase, each sensor in the DSN selects a suitable vertex from its own local Voronoi cell for obtaining a higher *intracell* coverage ratio. This will be beneficial to avoid coverage overlap across the cells as much as possible and keep higher overall coverage ratio of the sensing field. The detail of the algorithm for this phase is shown as Algorithm 1.

*4.3. Rotation of Working Direction.* The target location of sensor movement is selected under the criteria described in previous section. After the directional sensor has moved to the suitable vertex on its local Voronoi cell, the next is to decide a working direction that the direction should be controlled to make sensor contribute to full *intracell* coverage as much as possible. By this rule, the coverage overlaps across the cells will be decreased, and it benefits the overall filed coverage.

Figure 6 shows an example model to control the working direction. The sensor is at the position of vertex $v$ with coordinate $(x_v, y_v)$. The working direction, notated as $\theta$, should be controlled in the range limited by the angle values $\theta_1$ and $\theta_2$, and then the sensing coverage of the sensor will not go beyond the cell edges $e_R$ or $e_L$. $\theta_1$ is the most right limit of direction for reserving the *intracell* coverage, and $\theta_2$ is the most left limit. Because the sensor is direction rotatable and the FoV of the sensor should not go beyond the cell edges of

get self-coordinate $(x_s, y_s)$
broadcast $(x_s, y_s)$ to neighbor sensors
receive coordinates from neighbors
construct local Voronoi cell $C_s = (V, E)$
  $V = \{v_1, v_2, \ldots, v_m\}$ is the set of vertices of the cell
  $E = \{e_1, e_2, \ldots, e_n\}$ is the set of edges of the cell
$T_2 = \phi$; $T_1 = \phi$; $T_0 = \phi$;
for each $v(x_v, y_v) \in V$
{
  let $v_L(x_L, y_L)$ is the left adjacent vertex of $v$
  let $v_R(x_R, y_R)$ is the right adjacent vertex of $v$
  $\beta_v = \dfrac{(x_L - x_v)(x_R - x_v) + (y_L - y_v)(y_R - y_v)}{\sqrt{(x_L - x_v)^2 + (y_L - y_v)^2}\sqrt{(x_R - x_v)^2 + (y_R - y_v)^2}}$ is the included angle of $\overrightarrow{vv_L}$ and $\overrightarrow{vv_R}$
  if $(\beta_v \geq \alpha)$
    if $\left(\left\|\overrightarrow{vv_L}\right\| \geq r \text{ and } \left\|\overrightarrow{vv_R}\right\| \geq r\right)$
      $T_2 = T_2 \cup \{v\}$
    else
      if $\left(\left\|\overrightarrow{vv_L}\right\| < r \text{ and } \left\|\overrightarrow{vv_R}\right\| < r\right)$
        $T_0 = T_0 \cup \{v\}$
      else
        $T_1 = T_1 \cup \{v\}$
      endif
    endif
  else
    $T_0 = T_0 \cup \{v\}$
  endif
}
if $(T_2 \neq \phi)$
  $t = \arg\max_{v \in T_2} \beta_v$
else
  if $(T_1 \neq \phi)$
    $t = \arg\max_{v \in T_1} \beta_v$
  else
    $t = \arg\max_{v \in T_0} \beta_v$
  endif
endif

Algorithm 1: Target location selection for sensor movement.

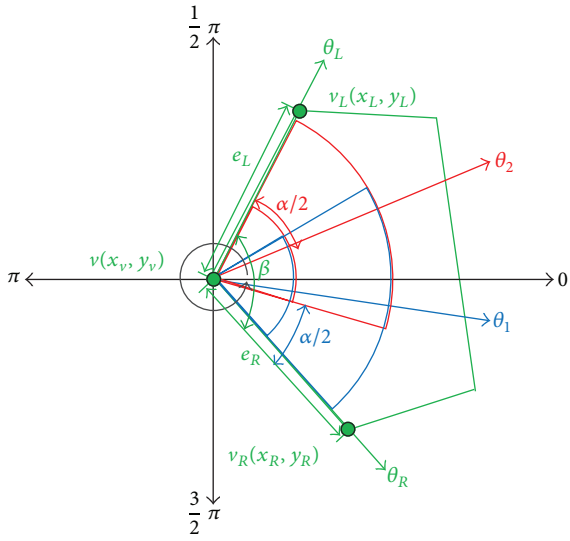



Figure 6: Rotation of working direction.

$e_R$ and $e_L$, the angle values $\theta_R$ and $\theta_L$ of the vectors $\overrightarrow{vv_R}$ and $\overrightarrow{vv_L}$ relative to the positive $x$-axis are calculated as the following (7), respectively. The values of $\theta_R$ and $\theta_L$ are between 0 and $\pm\pi$. Consider

$$\theta_R = \begin{cases} \sin^{-1}\dfrac{y_R - y_v}{\sqrt{(x_R - x_v)^2 + (y_R - y_v)^2}}, & x_R > x_v \\ \pi - \sin^{-1}\dfrac{y_R - y_v}{\sqrt{(x_R - x_v)^2 + (y_R - y_v)^2}}, & x_R < x_v,\ y_R > y_v \\ -\pi - \sin^{-1}\dfrac{y_R - y_v}{\sqrt{(x_R - x_v)^2 + (y_R - y_v)^2}}, & x_R < x_v,\ y_R < y_v \end{cases}$$

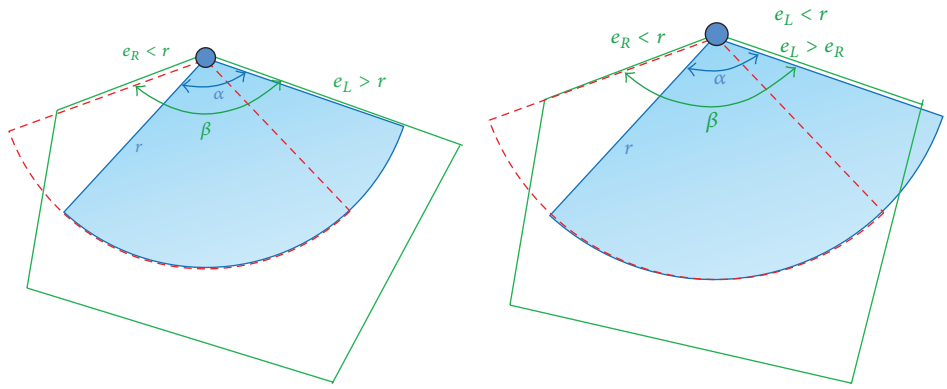



Figure 7: Two other cases of direction rotation.

$$\theta_L = \begin{cases} \sin^{-1}\dfrac{y_L - y_v}{\sqrt{(x_L - x_v)^2 + (y_L - y_v)^2}}, & x_L > x_v \\ \pi - \sin^{-1}\dfrac{y_L - y_v}{\sqrt{(x_L - x_v)^2 + (y_L - y_v)^2}}, & x_L < x_v,\ y_L > y_v \\ -\pi - \sin^{-1}\dfrac{y_L - y_v}{\sqrt{(x_L - x_v)^2 + (y_L - y_v)^2}}, & x_L < x_v,\ y_L < y_v. \end{cases} \tag{7}$$

If the working direction rotates right toward the $\theta_R$ and the right sideline of the sector is lying on the cell edge $e_R$, then the angle value of working direction at the moment is $\theta_1$. Similarly, $\theta_2$ is the angle value of working direction while the sensor rotates left toward the $\theta_L$ and the left sideline of the sector is lying on the cell edge $e_L$. Because the AoV of the sensor is $\alpha$, $\theta_1$ and $\theta_2$ can be calculated as (8). The values of $\theta_1$ and $\theta_2$ are between 0 and $\pm\pi$. Consider

$$\theta_1 = \begin{cases} \theta_R + \dfrac{\alpha}{2}, & \theta_R + \dfrac{\alpha}{2} \le \pi \\ \theta_R + \dfrac{\alpha}{2} - 2\pi, & \text{otherwise} \end{cases}$$
$$\theta_2 = \begin{cases} \theta_L - \dfrac{\alpha}{2}, & \theta_L - \dfrac{\alpha}{2} \ge -\pi \\ \theta_L - \dfrac{\alpha}{2} + 2\pi, & \text{otherwise.} \end{cases} \tag{8}$$

Figure 7 shows other two different cases in which $e_L$ is longer than $e_R$. However in Figure 7(a), sensing radius $r$ is longer than $e_R$ and shorter than $e_L$; in Figure 7(b), sensing radius is longer than both $e_R$ and $e_L$. In both of the two cases, making the left sideline of the sector lie on $e_L$ and making the right sideline lie on $e_R$ will obtain different *intracell* coverages. In the phase of working direction rotation, the proposed algorithm makes one of the sidelines of the sector lie on the longer one of $e_L$ and $e_R$; for example, the $e_L$ in Figure 7. This is mostly beneficial to reserve larger *intracell* coverage. Therefore, if $\theta_0$ is the original working direction after the sensor was initially deployed and moved to the target vertex, the rotation range $\varphi$ of the sensor is shown as (9). The sensor will make a clockwise or counterclockwise rotation with a range size of $\varphi$. Consider

$$\varphi = \begin{cases} \begin{cases} |\theta_0 - \theta_1|, & |\theta_0 - \theta_1| \le \pi \\ 2\pi - |\theta_0 - \theta_1|, & |\theta_0 - \theta_1| > \pi \end{cases}, & e_R \ge e_L \\ \begin{cases} |\theta_0 - \theta_2|, & |\theta_0 - \theta_2| \le \pi \\ 2\pi - |\theta_0 - \theta_2|, & |\theta_0 - \theta_2| > \pi \end{cases}, & e_L > e_R. \end{cases} \tag{9}$$

The detail of the algorithm for this phase is shown as Algorithm 2.

*4.4. Summary of the Procedures.* This subsection summarizes the procedures of the proposed algorithm. As shown in Figure 8, there are seven major procedures in the proposed distributed self-redeployment algorithm. The procedures are performed in each sensor with only few local information (the coordinates). Global information is not required. After the procedures are finished, overall coverage of sensing field is improved and then the sensors begin to do their sensing works.

Regarding the time complexity, the best known Fortune algorithm [27] can generate a Voronoi diagram from a given set of points in $O(n \log n)$ time. However, in our proposed DVSA algorithm, each distributed sensor only constructs a local Voronoi cell by itself after the initial deployment according to the coordinates of its neighbor sensors. Therefore, the time complexity is $O(n)$ for the initialization phase. In the decision phase, each sensor selects one of the cell vertices as the target position of movement. The time complexity is also $O(n)$. Finally, the sensor decides the working direction according to the two edges connected at the target vertex. The time complexity is $O(1)$. Therefore, the proposed DVSA algorithm has an overall time complexity of $O(n)$. This is better than the centralized algorithm of $O(n^2 \log n + n^3)$ and the distributed algorithm of $O(n \log n)$ proposed in [21].

$v\ (x_v, y_v)$ is the coordinate of new sensor position
let $v_R(x_R, y_R)$ is the right adjacent vertex of $v$
let $v_L(x_L, y_L)$ is the left adjacent vertex of $v$
let $\theta_0$ is the initial direction of the sensor
function get_edge_angle_value $(x, y)$
{

$$\theta' = \begin{cases} \sin^{-1}\dfrac{y - y_v}{\sqrt{(x - x_v)^2 + (y - y_v)^2}}, & x > x_v \\ \pi - \sin^{-1}\dfrac{y - y_v}{\sqrt{(x - x_v)^2 + (y - y_v)^2}}, & x < x_v \text{ and } y > y_v \\ -\pi - \sin^{-1}\dfrac{y - y_v}{\sqrt{(x - x_v)^2 + (y - y_v)^2}}, & x < x_v \text{ and } y < y_v \end{cases}$$

return $\theta'$
}
function get_rotation_range $(\theta_i)$
{

$$\theta' = \begin{cases} |\theta_0 - \theta_i|, |\theta_0 - \theta_i| \leq & \pi \\ 2\pi - |\theta_0 - \theta_i|, |\theta_0 - \theta_i| > & \pi \end{cases}$$

return $\theta'$
}
$\theta_R$ = get_edge_angle_value $(x_R, y_R)$
$\theta_L$ = get_edge_angle_value $(x_L, y_L)$

$$\theta_1 = \begin{cases} \theta_R + \alpha/2, & \theta_R + \alpha/2 \leq \pi \\ \theta_R + \alpha/2 - 2\pi, & \text{otherwise} \end{cases}$$

$$\theta_2 = \begin{cases} \theta_L - \alpha/2, & \theta_L - \alpha/2 \geq -\pi \\ \theta_L - \alpha/2 + 2\pi, & \text{otherwise} \end{cases}$$

$e_R = \left\|\overrightarrow{vv_R}\right\|$
$e_L = \left\|\overrightarrow{vv_L}\right\|$
if $(e_R \geq e_L)$
$\varphi$ = get_rotation_range $(\theta_1)$
rotate the working direction from $\theta_0$ to $\theta_1$ with rotation range $\varphi$
else
$\varphi$ = get_rotation_range $(\theta_2)$
rotate the working direction from $\theta_0$ to $\theta_2$ with rotation range $\varphi$
endif

Algorithm 2: Working direction rotation for directional sensor.

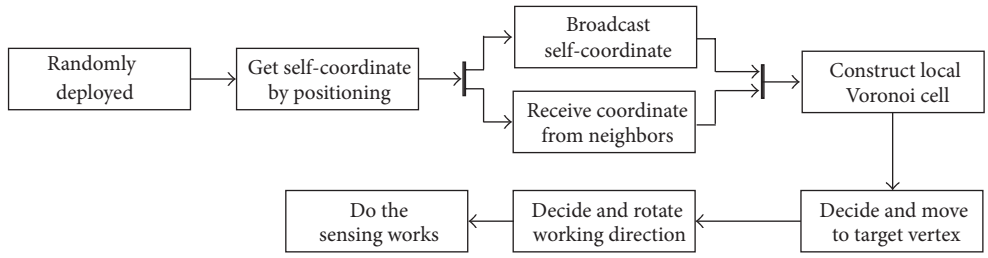


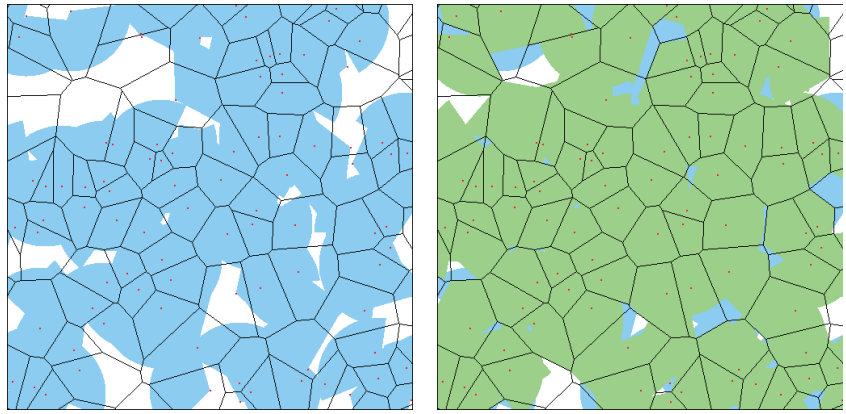

Figure 8: Procedures of the proposed self-redeployment algorithm.

## 5. Performance Evaluation

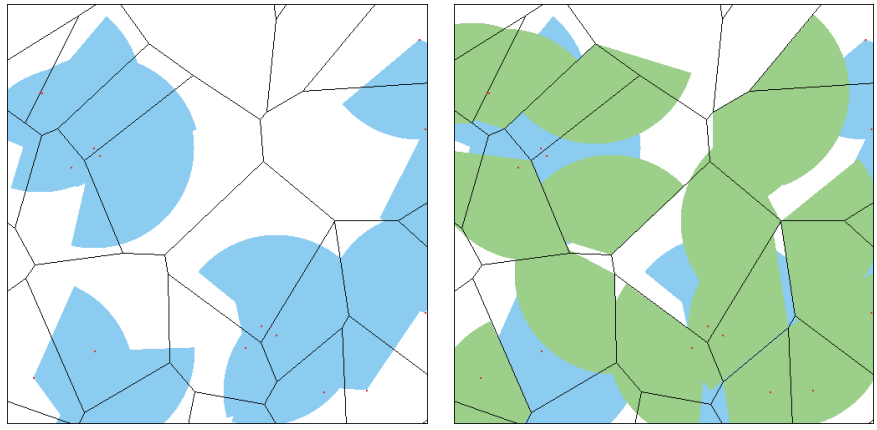
This study evaluated the performance of the proposed distributed Voronoi-based self-redeployment algorithm (DVSA) by simulations. Field coverage ratio is the problem concerned in the algorithm and it is the key point of the performance evaluation. In the simulations, mobile and direction-rotatable sensors are randomly deployed in a sensing field at the initial phase. The parameter of sensing field size is fixed at 500 m × 500 m. The other key parameters of the simulations comprise (1) number of sensors, (2) AoV of the sensor, and (3) sensing radius of the sensor.

Table 2: Notations and parameters for the directional sensing model.

| Parameters | Notations | Default value | Variational values |
|---|---|---|---|
| Size of sensing field | $A$ | 500 m × 500 m | None |
| Number of sensors | $n$ | 150 | 50, 70, 90, 110, 130, 150 |
| Sensing radius | $r$ | 60 m | 10 m, 20 m, 30 m, 40 m, 50 m, 60 m |
| Angle of view | $\alpha$ | 180° ($\pi$) | 30°, 60°, 90°, 120°, 150°, 180° |



(a) Case 1: Initial versus DVSA



(b) Case 2: Initial versus DVSA

Figure 9: Pictures captured from the simulation program.

There are variations in values of the parameters in the simulations. As shown in Table 2, the number of sensors is 50~150 with an interval of 20, the sensing radius is 10 m~60 m with an interval of 10 m, and the angle of view (AoV) is 30°~180° with an interval of 30°. Each case in the simulations was repeated 100 times, and the average was taken as the result data. The simulation results are described in the following subsections. And Figure 9 illustrates a result with pictures captured from the simulation program.

Sections 5.1, 5.2, and 5.3 show the coverage performance of the proposed DVSA and the relationship among the number of sensors, sensing radius, and AoV. Then Section 5.4 shows the comparison with the other algorithms so as to prove that the proposed DVSA can improve the sensing field coverage well. Finally, Section 5.5 shows the performance result of sensors with different radii and AoVs; in other words, each sensor has a random radius and a random AoV different from the ones of the other sensors.

*5.1. Coverage Ratio with Various Sensing Radii and AoVs.* Figure 10 shows the results of simulation by changing the sensing radius ($r$) and the AoV ($\alpha$), while fixing the number of sensors of $n = 150$. It illustrates the relationship among the coverage ratio, the sensing radius, and the AoV. For each AoV curve, the coverage ratio is increased with the increase of sensing radius. In addition, the larger the AoV is, the larger the slope is. In other words, this indicates that the larger the AoV is, the larger the increment of coverage ratio is. For instance, there is a coverage ratio increment of 43.72% from 1.52% (at $r = 10$ m) to 45.24% (at $r = 60$ m) on the curve of $\alpha = 60°$, but an increment of 89.17% from 8.22% (at $r = 10$ m) to 97.39% (at $r = 60$ m) on the curve of $\alpha = 180°$. On

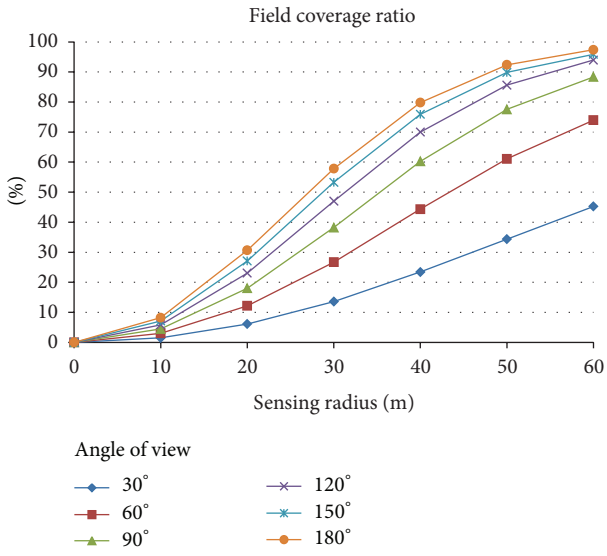



Figure 10: Coverage ratio (%) with fixed number of sensors ($n$ = 150).

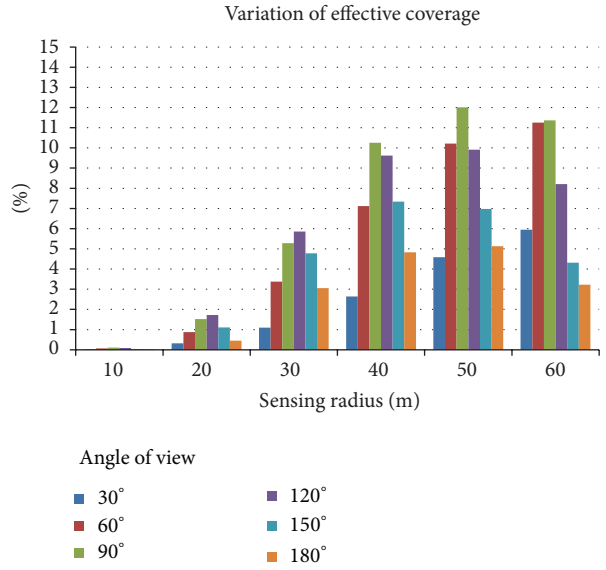



Figure 11: Variation of effective coverage (%) with fixed number of sensors ($n$ = 150).

the other hand, the figure shows that the larger the AoV is, the smaller the interval between two adjacent AoV curves is. This indicates that the increment of coverage contribution caused by the increment of AoV becomes smaller because the coverage overlap becomes larger.

Figure 11 shows the variation of effective coverage, which indicates the increments of coverage ratio relative to the initial ratio after deployment. It can be found that when $r$ = 10 m (one-fiftieth of the side length of the sensing field), there is a very small (almost zero) increment of coverage. This may means the situation of almost no coverage overlap among sensors. In the figure, it also can be found that the peak value of increment occurs at $r$ = 50 m and $\alpha$ = 90°, which

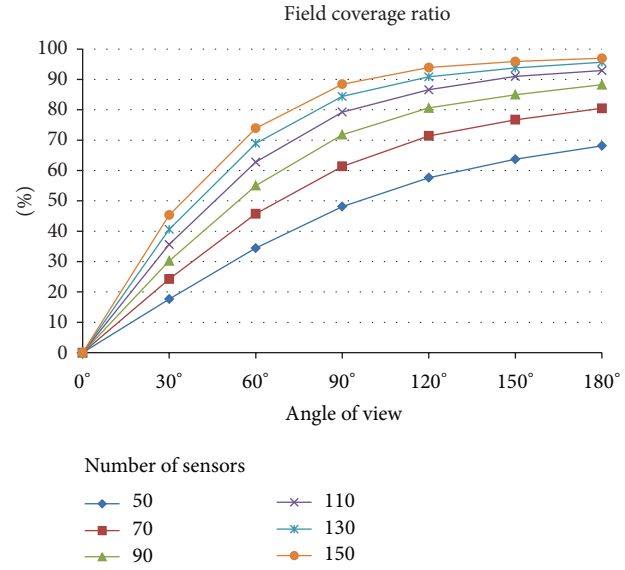



Figure 12: Coverage ratio (%) with fixed sensing radius ($r$ = 60 m).

means that these two parameter values can help obtain a best performance under the fixed number of sensors ($n$ = 150). Furthermore, for the $\alpha$ = 30° and 60°, the larger the sensing radius is, the larger the increment of coverage ratio is; but for $\alpha$ = 120°, 150°, and 180°, the increments at $r$ = 60 m become smaller than the ones at $r$ = 50 m and at $r$ = 40 m. This means that the sensing radius larger than 60 m with the AoV larger than 120° will not keep raising the increment of coverage ratio. In other words, although the increment of coverage will become smaller under the cases, it will still be larger than the increment of overlap and the proposed algorithm will still obtain an improved coverage.

*5.2. Coverage Ratio with Various AoVs and Numbers of Sensors.* Figure 12 shows the results of simulation by changing the AoV ($\alpha$) and the number of sensors ($n$), while fixing the sensing radius of $r$ = 60 m. It illustrates the relationship among the coverage ratio, the AoV, and the number of sensors. For each curve, the coverage ratio is increased with the increase of AoV. All of the curves have a larger slope at the beginning of the curve and a smaller slope at the end of the curve. This means that any curve does not keep the increment of coverage ratio equivalent every time the AoV is increased. For instance, there is a coverage ratio increment of 16.77% from 17.67% (at $\alpha$ = 30°) to 34.44% (at $\alpha$ = 60°) on the curve of $n$ = 50, but only an increment of 4.42% from 63.71% (at $\alpha$ = 150°) to 68.13% (at $\alpha$ = 180°) on the same curve. The larger the AoV is, the smaller the coverage contribution is smaller. Also, the figure shows that the larger the number of sensors is, the smaller the interval between two adjacent curves is. This indicates that the increment of coverage contribution caused by the increment of number of sensors becomes smaller because the coverage overlap becomes larger.

Figure 13 shows the variation of effective coverage of fixed sensing radius ($r$ = 60 m) as AoV varies from 30° to 180°

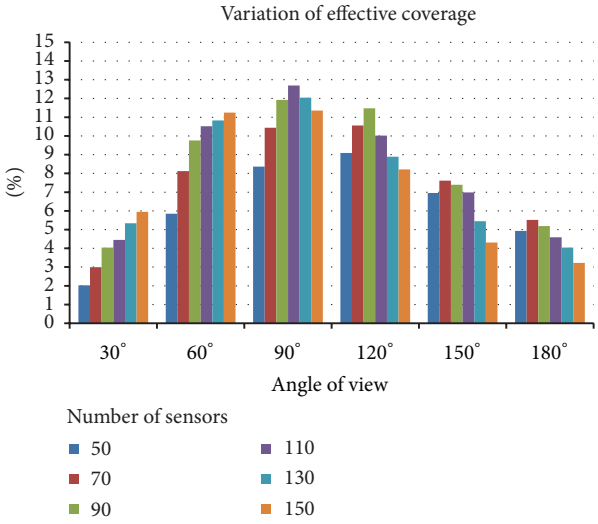



Figure 13: Variation of effective coverage (%) with fixed sensing radius ($r$ = 60 m).

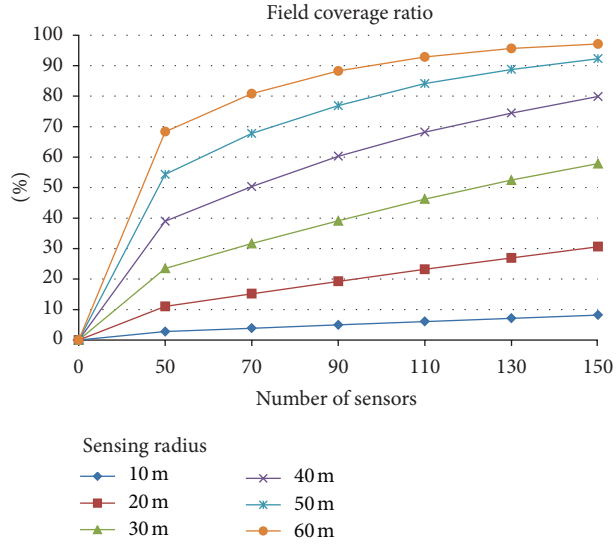



Figure 14: Coverage ratio (%) with fixed AoV ($\alpha$ = 180°).

and number of sensors varies from 50 to 150. For all of these cases, the proposed algorithm obtained positive increments of the coverage relative to the initial values significantly. The peak value of increment occurs at $\alpha = 90^\circ$ and $n = 110$, which means that these two parameter values can help obtain a best performance under the fixed sensing radius ($r$ = 60 m). And in the figure, it can be found that the larger increments occur around $\alpha = 60^\circ$~$120^\circ$. The AoV larger than 120° cannot bring more increment. This should be caused by the increase of coverage overlap. Furthermore, it also can be found that at $\alpha = 30^\circ$ (smallest AoV), the bar of $n = 150$ is the highest; on the contrary, at $\alpha = 180^\circ$ (the largest AoV), the bar of $n = 150$ is the lowest. Although the largest $n$ with largest $\alpha$ brings considerably large coverage overlap, it still have positive increment of coverage ratio due to the larger coverage contribution.

*5.3. Coverage Ratio with Various Numbers of Sensors and Sensing Radii.* Figure 14 shows the results of simulation by changing the number of sensors ($n$) and the sensing radius ($r$), while fixing the AoV of $\alpha = 180^\circ$. It illustrates the relationship among the coverage ratio, the number of sensors, and the sensing radius. The coverage ratio is low while the sensing radius is the shortest (at $r$ = 10 m); however, the increment of coverage ratio is significant if the sensing radius is increased. But the slopes of the curves seem smaller than the ones in the figures illustrated in previous subsections. The figure looks like that the increase of sensing radius brings more significant coverage ratio increment than the increase of number of sensors.

Figure 15 shows the variation of effective coverage of fixed AoV ($\alpha = 180^\circ$) as number of sensors varies from 50 to 150 and sensing radius varies from 10 m to 60 m. It can be found that in certain cases even if the sensing radius is small and the number of sensors is not many, the increment of coverage relative to the initial value after deployment is a negative value

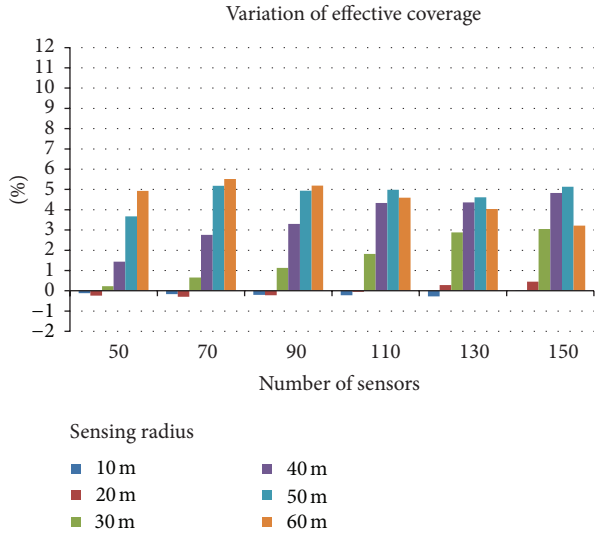



Figure 15: Variation of effective coverage (%) with fixed AoV ($\alpha$ = 180°).

(around −0.1%~−0.3%). This should be caused by the large AoV ($\alpha = 180^\circ$), and the coverage overlap is a little larger than the coverage contribution in these cases. In addition, the difference of the increment between different numbers of sensors with same sensing radius is averagely not significant. This may mean that the increment of coverage is near the increase of overlap in these cases.

*5.4. Comparison with Different Algorithms.* The following show the results of comparing our proposed DVSA with the algorithms of VDA proposed in [21] and mentioned in Section 2. The RND (random) method has also been compared, which the RND means the initial value after the random deployment of sensors (with random position and random direction). Firstly, Figure 16 is the result by changing

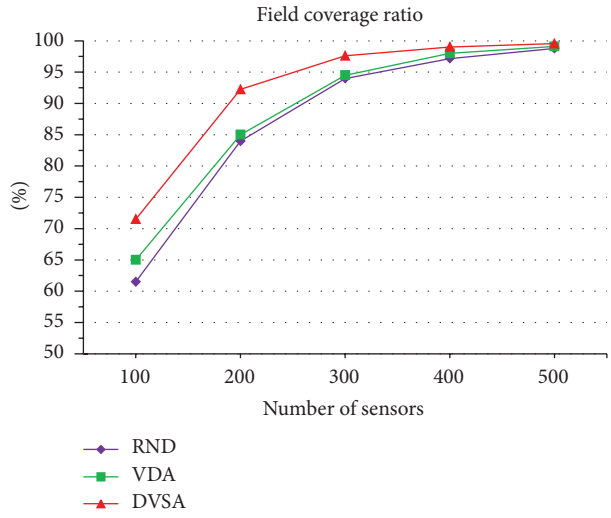



Figure 16: DVSA Compared with RND and VDA ($n$ = 100 ~500, $r$ = 50 m, $\alpha$ = 120°).

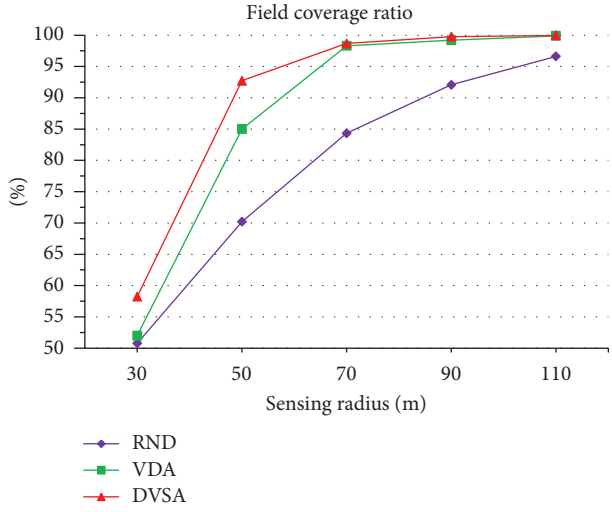



Figure 17: DVSA Compared with RND and VDA ($n$ = 200, $r$ = 30 m~110 m, $\alpha$ = 120°).

the number of sensors from 100 to 500, and the sensing radius and AoV are fixed at 50 m and 120°, respectively. The DVSA curve shows a significant improvement of coverage performance, which is better than both VDA and RND. At the value of $n$ = 100, the improvement is the largest, but the larger the $n$ is, the lesser the improvement is. In particular at $n$ = 500, DVSA, VDA and RND almost have the same performance; more sensors cannot bring more coverage contribution. Figure 17 shows the result of setting parameters as $n$ = 200, $r$ = 30 m~110 m, and $\alpha$ = 120°. The DVSA performs better than VDA at smaller number of sensors and performs almost equally at larger number of sensors. They converge on the point of $r$ that is larger than 70 m while the performance RND is still significantly lower. Figure 18 shows the result of setting parameters as $n$ = 200, $r$ = 50 m, and $\alpha$ = 60° ~ 240°. The DVSA still performs significantly better than

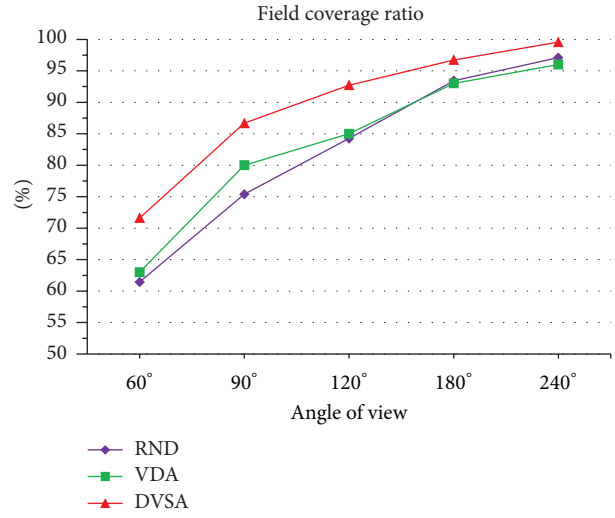



Figure 18: DVSA Compared with RND and VDA ($n$ = 200, $r$ = 50 m, $\alpha$ = 60°~240°).

VDA. The VDA performs almost the same as RND except at the point of $\alpha$ = 90°. From these results, the proposed DVSA is proved that it performs very well under the situations of varied numbers of sensors, sensing radii, and angles of view.

Figure 19 shows the coverage ratio of the proposed DVSA algorithm compared with the result presented in the related study [19]. The simulation parameters: the size of sensing field is 500 m × 500 m; the number of sensors is 100; the sensing radius is 60 m; and the AoV is 60°, which are the same with the ones in that study. In the Figure 19, it shows that the coverage ratio of the sensing field can be well improved by the proposed DVSA algorithm while being compared with the related algorithms.

Figure 20 shows the result of the proposed DVSA algorithm compared with the GSP algorithm presented in the related study [20]. This simulation uses a large number of sensors of $n$ = 1000. The sensing radius is fixed at $r$ = 20 m in Figure 20(a), and the AoV is fixed at $\alpha$ = 120° in Figure 20(b). The results also show that the DVSA obtained better coverage ratios than the GSP. The DVSA algorithm can be performed well.

In summary, the proposed DVSA method utilizes the vertices, edges, and included angles in each Voronoi cell to precisely compute the most suitable location and working direction for each sensor according to the algorithms described in Sections 4.2 and 4.3. On the contrary, the VDA approximately considers that if most Voronoi edges are covered, then most area will be covered; this is not definite and may cause more coverage overlap. In LRBA and MBAA algorithms, some nodes could not find their respective paired partner and cannot perform the algorithms for coverage contribution. And in the GSP algorithm, it needs the deployment of more sensors for repairing processes if certain grouped sensors are incommunicable to the sink, and the average coverage ratio of the grouped sensors will be decreased. Accordingly, the above performance results show

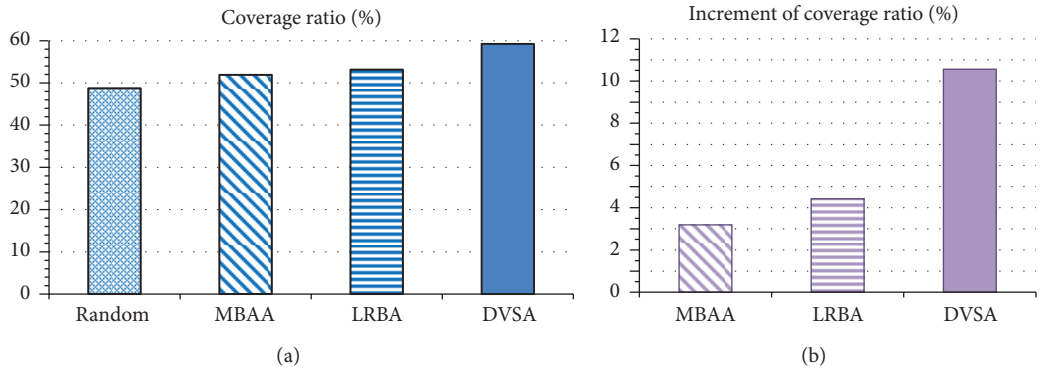



Figure 19: Simulation result compared with different algorithms ($n = 100$, $r = 60$ m, $\alpha = 60^{\circ}$).

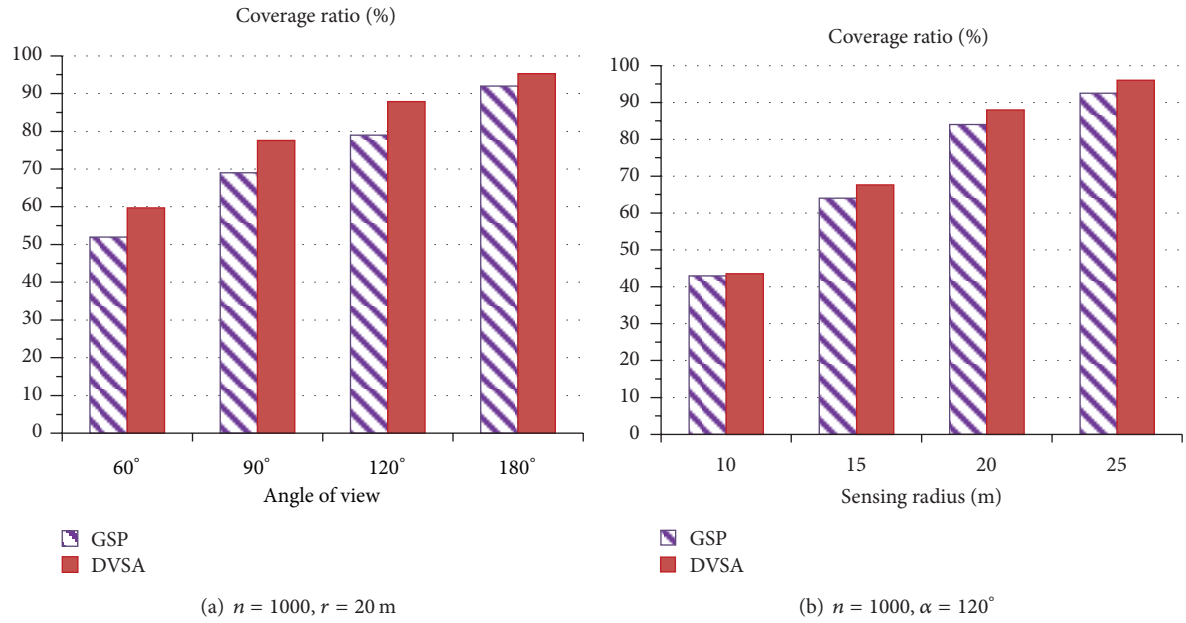



Figure 20: Simulation result compared with the GSP algorithm.

that our proposed DVSA performs better than the algorithms of VDA, LRBA, MBAA, and GSP.

*5.5. Performance of Sensors with Different Sensing Radii and AoVs.* In this subsection, the result of coverage performance of sensors with different radii and different AoVs is shown. In other words, the sensing radius and AoV of a sensor are randomly generated and different from the ones of the other sensors. Figure 21 illustrates this scenario with the pictures captured from the screenshot of the simulation program.

The parameter of the number of sensors varies from 30 to 200. The sensing radius of a sensor is randomly generated with a range between $r_{\min} = 10$ m and $r_{\max} = 100$ m, and the AoV is randomly generated with a range between $\alpha_{\min} = 30^{\circ}$ and $\alpha_{\max} = 150^{\circ}$. Figure 22 shows the performance of DVSA compared with RND (random deployment). DVSA performs very well with an improved coverage ratio, no matter what the number of sensors is. And the increment of coverage ratio is kept around 5%. This result proves that our proposed DVSA is applicable to sensors with different sensing radii and angles (we mentioned this in the beginning of Section 3.1).

## 6. Conclusion and Future Work

This study utilized the geometrical features of Voronoi diagram and the advantages of a distributed algorithm to propose the distributed Voronoi-based self-redeployment algorithm (DVSA), aiming to improve the overall field coverage of directional sensor networks effectively. The performance of the proposed algorithm and comparison with the different algorithm are also presented in the paper. The simulations prove that the DVSA method can improve the sensing coverage performance well.

Our future work will focus on combining the algorithm with an energy consumption model to give consideration to both coverage and lifetime performance in mobile and direction-rotatable directional sensor networks.

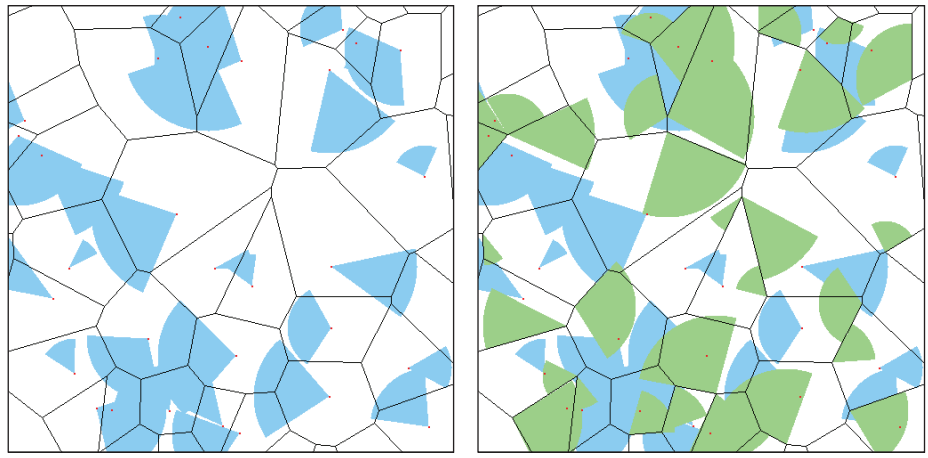


Figure 21: Pictures captured from the simulation program (initial versus DVSA).

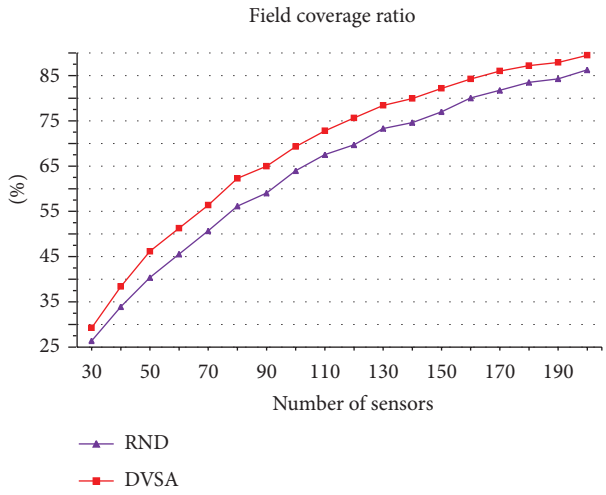



Figure 22: Performance ($n = 30 \sim 200$; $r_{min} = 10$ m, $r_{max} = 100$ m; $\alpha_{min} = 30^{\circ}$, $\alpha_{max} = 150^{\circ}$).

## Acknowledgments

This work is supported by National Science Council of Taiwan under Grant no. NSC 102-2219-E-006-001 and the Research Center for Energy Technology of National Cheng Kung University under Grant no. D102-23015.

## References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] C. Zhu, C. Zheng, L. Shu, and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619–632, 2012.

[3] G. J. Fan and S. Y. Jin, "Coverage problem in wireless sensor network: a survey," *Journal of Networks*, vol. 5, no. 9, pp. 1033–1040, 2010.

[4] Y. Charfi, N. Wakamiya, and M. Murata, "Challenging issues in visual sensor networks," *IEEE Wireless Communications*, vol. 16, no. 2, pp. 44–49, 2009.

[5] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Advances in Multimedia*, vol. 2009, Article ID 640386, 22 pages, 2009.

[6] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 1, pp. 21–41, 2006.

[7] M. G. Guvensan and A. G. Yavuz, "On coverage issues in directional sensor networks: a survey," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1238–1255, 2011.

[8] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[9] M. A. Guvensan and A. G. Yavuz, "A new coverage improvement algorithm based on motility capability of directional sensor nodes," in *Ad-Hoc, Mobile, and Wireless Networks*, vol. 6811 of *Lecture Notes in Computer Science*, pp. 206–219, Springer, Berlin, Germany, 2011.

[10] H. Ma and Y. Liu, "On coverage problems of directional sensor networks," in *Mobile Ad-Hoc and Sensor Networks*, vol. 3794 of *Lecture Notes in Computer Science*, pp. 721–731, Springer, Berlin, Germany, 2005.

[11] C. Pham, A. Makhoul, and R. Saadi, "Risk-based adaptive scheduling in randomly deployed video sensor networks for critical surveillance applications," *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 783–795, 2011.

[12] K.-Y. Chow, K.-S. Lui, and E. Y. Lam, "Wireless sensor networks scheduling for full angle coverage," *Multidimensional Systems and Signal Processing*, vol. 20, no. 2, pp. 101–119, 2009.

[13] J. Wang, C. Niu, and R. Shen, "Priority-based target coverage in directional sensor networks using a genetic algorithm," *Computers and Mathematics with Applications*, vol. 57, no. 11-12, pp. 1915–1922, 2009.

[14] Y.-C. Hsu, Y.-T. Chen, and C.-K. Liang, "Distributed coverage-enhancing algorithms in directional sensor networks with rotatable sensors," in *Distributed Computing and Networking*, vol. 7129 of *Lecture Notes in Computer Science*, pp. 201–213, Springer, Berlin, Germany, 2012.

[15] Z. Jing and Z. Jian-Chao, "A virtual centripetal force-based coverage-enhancing algorithm for wireless multimedia sensor

networks," *IEEE Sensors Journal*, vol. 10, no. 8, pp. 1328–1334, 2010.

[16] C.-K. Liang, M.-C. He, and C.-H. Tsai, "Movement assisted sensor deployment in directional sensor networks," in *Proceedings of the 6th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN '10)*, pp. 226–230, Hangzhou, China, December 2010.

[17] M. A. Guvensan and A. G. Yavuz, "A hybrid solution for coverage enhancement in directional sensor networks," in *Proceedings of the 7th International Conference on Wireless and Mobile Communications*, pp. 134–138, IARIA, Luxembourg City, Luxembourg, 2011.

[18] N. Tezcan and W. Wang, "Self-orienting wireless multimedia sensor networks for occlusion-free viewpoints," *Computer Networks*, vol. 52, no. 13, pp. 2558–2567, 2008.

[19] H. Huang, L. Sun, R. Wang, and J. Li, "A novel coverage enhancement algorithm for image sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 370935, 11 pages, 2012.

[20] H. Ma and Y. Liu, "Some problems of directional sensor networks," *International Journal of Sensor Networks*, vol. 2, no. 1-2, pp. 44–52, 2007.

[21] J. Li, R. Wang, H. Huanh, and L. Sun, "Voronoi-based coverage optimization for directional sensor networks," *Wireless Sensor Network*, vol. 1, no. 5, pp. 417–424, 2009.

[22] W. Cheng, S. Li, X. Liao, S. Changxiang, and H. Chen, "Maximal coverage scheduling in randomly deployed directional sensor networks," in *Proceedings of International Conference on Parallel Processing Workshops (ICPPW '07)*, Xian, China, September 2007.

[23] Z. Li, R. Li, Y. Wei, and T. Pei, "Survey of localization techniques in wireless sensor networks," *Information Technology Journal*, vol. 9, no. 8, pp. 1754–1757, 2010.

[24] G. T. Sibley, M. H. Rahimi, and G. S. Sukhatme, "Robomote: a tiny mobile robot platform for large-scale ad-hoc sensor networks," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1143–1148, Washington, DC, USA, May 2002.

[25] G. Wang, M. J. Irwin, P. Berman, H. Fu, and T. La Porta, "Optimizing sensor movement planning for energy efficiency," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 215–220, San Diego, Calif, USA, August 2005.

[26] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.

[27] S. Fortune, "A sweepline algorithm for Voronoi diagrams," *Algorithmica*, vol. 2, no. 1-4, pp. 153–174, 1987.

[28] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations, Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, 2nd edition, 2000.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 186720, 11 pages
http://dx.doi.org/10.1155/2013/186720

*Research Article*

# A Jigsaw-Based Sensor Placement Algorithm for Wireless Sensor Networks

**Shih-Chang Huang,[1] Hong-Yi Chang,[2] and Kun-Lin Wu[1]**

[1] *Department of Computer Science and Information Engineering, National Formosa University, Taiwan*
[2] *Department of Management Information Systems, National Chiayi University, Taiwan*

Correspondence should be addressed to Hong-Yi Chang; alanc68@gmail.com

Received 26 June 2013; Revised 9 September 2013; Accepted 17 September 2013

Academic Editor: Shengming Jiang



Current deterministic sensor deploying methods always include the uncovered space greedily to reduce the number of deployed sensors. Because the sensing area of each sensor is circle-like, these greedily methods often divide the region of interest to multiple tiny and scattered regions. Therefore, many additional sensors are deployed to cover these scattered regions. This paper proposes a Jigsaw-based sensor placement (JSP) algorithm for deploying sensors deterministically. Sensors are placed at the periphery of the region of interest to prevent separating the region of interest to isolated regions. An enhanced mechanism is also proposed to improve the time complexity of the proposed method. The scenarios with and without obstacles are evaluated. The simulation results show that the proposed method can cover the whole region of interest with fewer deployed sensors. The effective coverage ratio of JSP method is less than 2. It is better than the maximum coverage method and the Delaunay triangulation method. The deploying sensors have more efficient coverage area, and the distribution of the incremental covered area is close to normal distribution.

## 1. Introduction

Sensor networks are applied for monitoring environment. Sensors are similar to the guard men standing at the specific locations for scouting. Usually, sensors have the wireless communication components to deliver their collected data back to the *sink*, which is a data collector in the network. Sensors can be deployed in the field for the environmental habitat monitoring [1], the natural hazards monitoring [2], the forest-fire detection [3], or the nuclear radiation detection [4]. Sensors are also used in the medical center to quicken the emergency response [5, 6]. Deployed sensors can also be used for localizing the users [7] or detecting the temperature of devices [8].

The placement of sensor nodes can be deterministic or random. The random deployment cannot guarantee the region of interest completely covered by the deployed sensors. Thus, more sensors need to be deployed to include the uncovered space [9, 10]. Giving the mobile ability for sensors to adjust their initial positions is the common used solution [11, 12]. The deterministic deploying methods consider covering the whole region of interest in the first time that sensors are deployed [13–17]. The object is determining the locations of sensors to minimize the number of deployed sensors. Finding minimum number of sensors to cover the whole space is an NP-complete problem similar to the coloring problem in a graph. Thus, finding a polynomial time solvable problem to approximate the optimal solution is the object of this research problem. The existing methods always greedily place a sensor to the place that can cover the largest uncovered space. So, the number of deployed sensors can be reduced. However, the sensing area of a sensor is circle-like. Using greedy methods to cover the whole region of interest introduces multiple small isolated regions. Sensors must be deployed in these scattered regions to achieve full coverage. Therefore, more sensors are deployed.

This paper proposes an enhanced deterministic deploying method named as the Jigsaw-based sensor placement (JSP) algorithm. Sensors are deployed from the periphery of the region of interest such as solving the jigsaw puzzle. The proposed JSP algorithm can prevent introducing the isolated

regions. The open-space scenario and center obstacle scenario are evaluated. The JSP can use fewer numbers of sensors to cover the region of interest. The overlapping coverage between sensors can also be reduced.

The rest of this paper is organized as follows. The related studies are reviewed in Section 2. Section 3 gives the main idea of JSP and the methodology to reduce the time complexity of JSP. The evaluation results are shown in Section 4. Finally, the conclusion is given in Section 5.

## 2. Related Works

The deterministic deploying methods focus on using minimum numbers of sensors to cover the whole region of interest. Dhillon et al. presented two methods, called as MAX_AVG_COV and MIN_AVG_COV [13, 14]. The region of interest is divided into $n \times n$ grids, and an $n^2 \times n^2$ matrix is used to represent the detection probability between any two grids. Each grid $g$ is scored by accumulating the probabilities of all neighbor grids covered by a sensor centered at $g$. The MAX_AVG_COV method chooses the highest score grid as the candidate position to place the next sensor. After a sensor is placed, the matrix is updated. The procedure continues until all grids are covered.

The MAX_AVG_COV method scans all grids and computes their scores. The time complexity to determine the deploying location is $O(n^4)$. In a large deploying area, the computation overhead on scoring the grids is enormous. In addition, reducing the grid size also generates high computation overhead. Greedily placing the next sensor at the location which can include the most uncovered space will introduce many small isolated regions as shown in Figure 1 and each isolated region needs a sensor to cover it. To prevent placing addition sensors on the tiny isolated regions, the MIN_AVG_COV method is proposed. The candidate uncovered grid to place the next sensor is the one which can include minimum uncovered space. MIN_AVG_COV method is proposed to prevent generating the isolated region, but the cost is to deploy more sensors. Xu and Sahnialso proposed a greedy method as the MIN_AVG_COV but using the integer linear programming formulation to find the minimum cost position for deploying of sensors [15] with heterogeneous sensors.

Wu et al. proposed a DT-Score algorithm [16] which utilizes the Delaunay triangulation to determine the location for placing sensors. The DT-Score algorithm consists of two phases. In the first phase, the sensors are placed along the contour of the boundary and the obstacles to eliminate the coverage holes are generated near them. All grids in the region of interest are scanned to identify the contours of the obstacles. The time complexity of this phase is $O(n^2)$. The second phase is to refine the deployment. New candidate positions to place sensors are the centers of the circumcircle of all triangles generated through the Delaunay triangulation algorithm. All candidate points are scored, and the next sensor is placed at the location which gets the most coverage gains. The Delaunay triangulation algorithm is applied to add vertices one by one continuously to include the uncovered space so that the time complexity to triangulate the graph is $O(m^3)$. The $m$ is the number of sensors deploying along the contour of obstacles. After that the DT-Score algorithm needs $O(n^2 \log n)$ time complexity to obtain the candidate location for the placing sensors. The procedure is repeated until the predefined number of deployable sensors is reached.

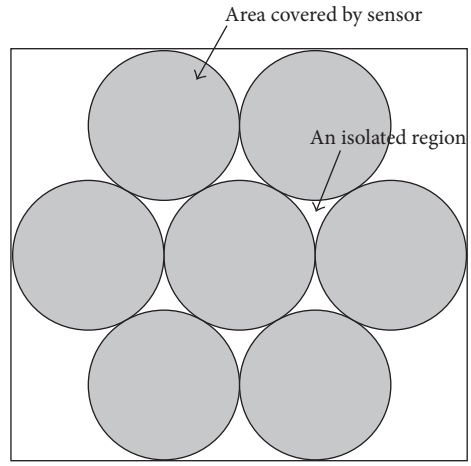



Figure 1: The generated isolated regions in greedy deployed methods.

At first, the DT-Score algorithm computes the centers of the circumcircles of the triangles generated by the Delaunay triangulation algorithm as the candidate positions. The center of the circum-circles with maximum radius is the position to place the next sensor. It also generates multiple scattered isolated regions as the MAX_AVG_COV method. The sensors placed near the obstacles only slightly moderated this problem. Therefore, Dhillon et al. also proposed the MIN_AVG_COV method for reducing the isolated regions. The MIN_AVG_COV method always places the next sensor at the location which includes the least uncovered space. The number of deployed sensors is also worse than MAX_AVG_COV. However, it gives the motivation to improve the deterministic deploying methods. If sensors can be added one by one from the periphery of the uncovered space, the probability to generate an isolated region will be reduced. It is similar to the strategy of playing a Jigsaw puzzle game. A similar method is proposed in [17] by considering a different coverage ratio.

## 3. The Jigsaw-Based Sensor Placement (JSP) Algorithm

*3.1. Basic JSP Algorithm.* Assume all sensors have the same sensing radius $R$. $V$ is the set containing all uncovered grids in the region of interest. The set $B \in V$ includes all boundary grids. Similar to the MAX_AVG_COV method, the region of interest is divided into $n \times n$ grids. Each grid $P_{ij}$ is initially assigned a token $T$ which is used for computing the score of each grid. The value of $T$ is set as

$$T\left(P_{ij}\right) = \begin{cases} 0, & \text{otherwise} \\ 1, & P_{ij} \in B, \end{cases} \quad \{i, j\} = [1 \cdots n]. \tag{1}$$

```
(1) V: The set of uncovered grids
(2) B: The set of the boundary grids
(3) P*: The highest score grid
(4) T(P_ij): Token of a grid point
(5) R: Sensing radius of a sensor node
(6) //Initial
(7) For all P_ij ∈ V
(8)       If P_ij ∉ B then
(9)           set T(P_ij) = 0
(10)      Else
(11)          set T(P_ij) = 1, add P_ij into B
(12) End
(13) //Scoring
(14) Do while V is not empty
(15)      Use (2) to compute the score of P_ij ∈ V
(16)      Get the highest score P* and place a sensor in it, remove P* from V
(17)      For all P_ij ∈ V
(18)           Case: |P*P_ij| < R
(19)               Set T(P_ij) = 0
(20)               Remove P_ij from V
(21)           Case: |P*P_ij| = R
(22)               Set T(P_ij) = 1
(23)               Add P_ij into B
(24)      End
(25)      If P_ij ∉ V
(26)          Remove P_ij from B
(27) End
```

Algorithm 1: The scoring algorithm of the JSP.

Let $C_{ij}$ be the set containing all grids which are covered by the sensor centered at $P_{ij}$. Thus, the grids $P_{xy}$ included by $C_{ij}$ must satisfy $|\overline{P_{xy}P_{ij}}| < R$. The score of $P_{ij}$ is the accumulated tokens of the grids in $C_{ij}$. We can represent it as

$$S\left(P_{ij}\right) = \sum_{\forall P_{xy} \in C_{ij}} T\left(P_{xy}\right), \quad \{x, y\} = [1 \cdots n]. \tag{2}$$

And the next candidate grid to place a sensor will be the one which has the highest score:

$$P^* = \max\left\{S\left(P_{ij}\right)\right\}. \tag{3}$$

After placing a sensor at the grid $P^*$, the grids $g \in C_{ij}$ are removed from $V$. The set $B$ is also updated for refreshing the boundary. The deploying procedure continues until all elements in $V$ are exhausted. The scoring algorithm is given in Algorithm 1.

The time complexity of JSP is also proportional to the number of grids. Let the grid size be $g^2$. All grids in $V$ need to be scored. Each grid requires scanning the rectangle that includes the sensing area of a sensor to accumulate the tokens. The rectangle is $(R^2/g^2) \in O(R^2)$ area. The initial number of grids in set $V$ is $n \times n$ and decreases after sensors are deployed. The average grid decreasing rate in $V$, denoted as $\delta$, is ranged from 1 to $R^2/g^2$. The time complexity can be represented as the recursive relation, $T(n^2)$. The time complexity is $O(n^4R^2)$. Consider the following:

$$\begin{aligned} T\left(n^2\right) &= T\left(n^2 - \delta\right) + n^2\frac{\left(R^2\right)}{g^2} \\ &= T(0) + \left(n^2 - \left(\frac{n^2}{\delta}\right)\delta\right)\frac{\left(R^2\right)}{g^2} + \cdots + \left(n^2 - 2\delta\right) \\ &\quad \times \frac{\left(R^2\right)}{g^2} + \left(n^2 - \delta\right)\frac{\left(R^2\right)}{g^2} + n^2\frac{\left(R^2\right)}{g^2} \\ &= \left(\frac{n^2}{2\delta}\right)\left(n^2\frac{\left(R^2\right)}{g^2}\right) \in O\left(n^4R^2\right). \end{aligned} \tag{4}$$

A little example is given in Figure 2 to show the operation of the JSP algorithm. Initially, all boundary grids are set to 1 and others are set to 0 as shown in Figure 2(a). Next, each grid is rescored according to the number of included boundary grids if a sensor is placed in it. As the grid (4, 3) in the Figure 2(b), it includes the boundary grids (1, 2), (1, 3), (1, 4), (1, 5), and (1, 6). So, its score is 5 (note that a grid is counted as included if more than two-thirds of the grids are covered by the sensor). Figure 2(b) shows that four grids (3, 3), (6, 3), (3, 6), and (6, 6) have the highest score 9, and the first one (3, 3) is selected as the next candidate position to place a sensor.

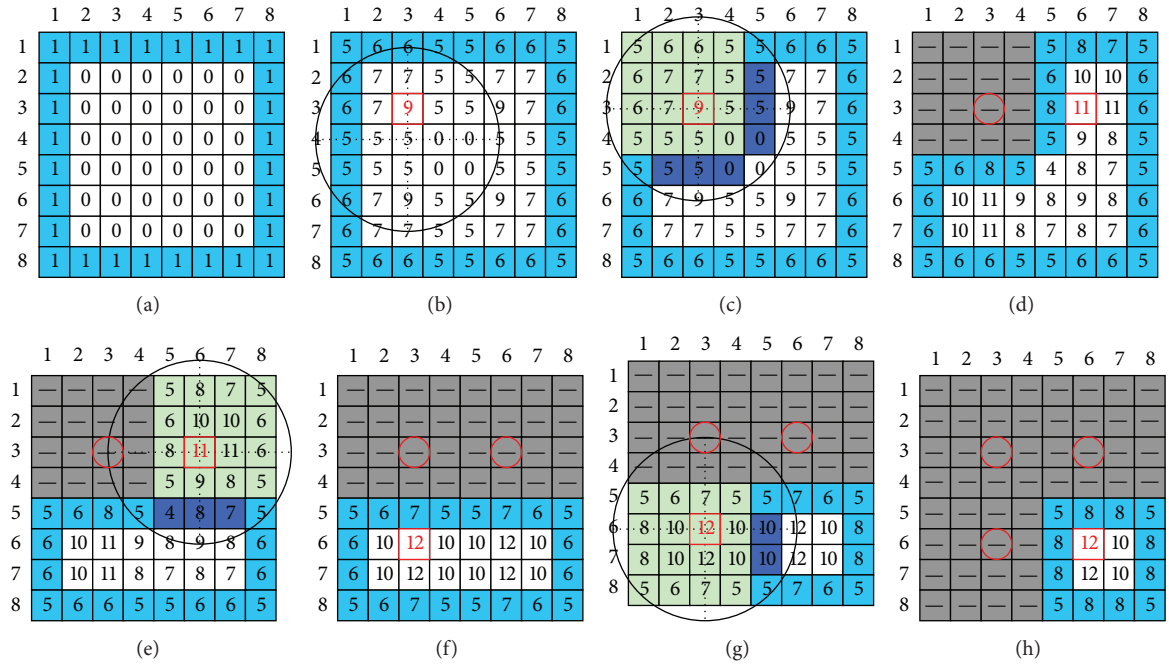



Figure 2: An example of the JSP algorithm.

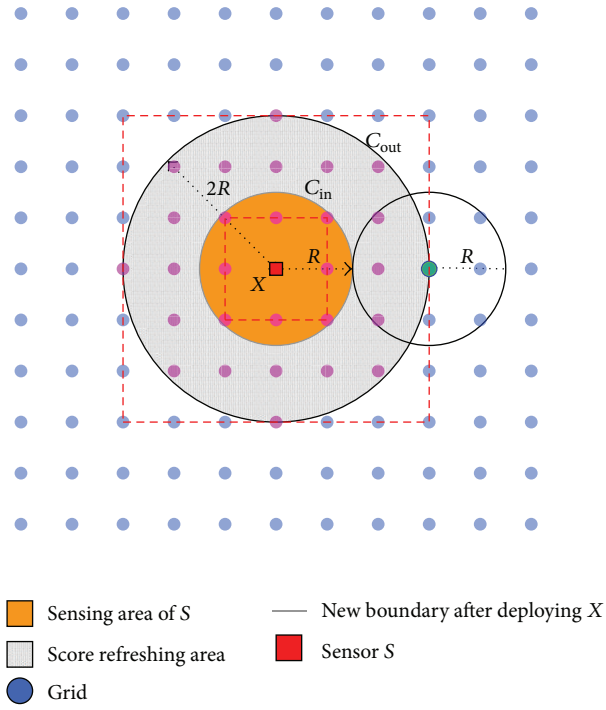



Figure 3: The rescoring area after a sensor node deployed.

As a sensor is placed at (3, 3), the grids covered by sensor is marked covered. The boundary grids covered by sensor is removed, and the grids located on the edge of the sensor coverage will be marked as new boundary grids as shown in Figure 2(c). Then, all uncovered grids will be re-scored as shown in Figure 2(d). The selected grid with the highest score is the (3, 6). After the next sensor is placed, covered grids are marked and new boundary grids are selected as shown in the Figure 2(e). This procedure is performed until all grids are covered. In this example, the final positions to place sensors will be (3, 3), (6, 3), (3, 6), and (6, 6).

*3.2. Improved JSP Algorithm.* In Algorithm 1, all grids in the set $V$ are scanned to update the score value after a sensor $S$ is deployed. However, only those grids near the deployed $S$ need to be rescored instead of all grids in $V$. As the example in Figure 3, the sensor $S$ is placed at the grid $X$. Those grids covered by $S$ set their scores to zero. The grids in the coverage edge of $S$ become the new boundary. The uncovered grid whose minimum distance to the boundary grids is less than $R$ should update its score. The grids to refresh their score will be the ones whose locations are within the circle $C_{out}$ centered at grid $X$ with radius $2R$ but not in the circle $C_{in}$ centered at grid $X$ with radius $R$. The area size is $3R^2\pi$. To simplify the computation, the area included by the rectangle center at $X$ bound to the circle $C_{out}$ but not included in the rectangle center at $X$ with edge length $\sqrt{2}R$ is scanned and thus, the scan area will be $(2R)^2 - ((\sqrt{2}/2)R)^2 = 3.5R^2$. Therefore, $T(n^2)$ can be represented as

$$\begin{aligned} T\left(n^2\right) &= T\left(0\right) + \left(3.5R^2\right)R^2 + \cdots + \left(3.5R^2\right)R^2 + n^2R^2 \\ &= \left(\frac{n^2}{\delta} - 1\right)\left(3.5R^2\right)R^2 + n^2R^2 \\ &= \left(\frac{n^2}{\delta}\right)\left(3.5R^2\right)R^2 - 3.5R^2 + n^2R^2 \in O\left(n^2R^4\right). \end{aligned} \tag{5}$$

```
(1) V: the set of uncovered grid points
(2) B: the set of the boundary grid points
(3) V*: the set of uncovered grid points within the circle centered at P*
(4) P*: the grid with maximal score
(5) //initial
(6)   For all P_ij in V    {
(7)      If P_ij is not a boundary point then
(8)      set T(P_ij) = 0
(9) Else
(10)     set T(P_ij) = 1, add P_ij into B
(11) }
(12)
(13) //scoring
(14) Computes the S(P_ij) of each P_ij in V according to (2)
(15) Use maximal heap tree to maintain sorting results of S(P_ij) in V
(16) Place the sensor at P*
(17) Do While (V is not empty) {
(18)         For all P_ij in V*       {
(19)         Case: |P*P_ij| < R
(20)               Set T(P_ij) = 0
(21)               Remove P_ij from V
(22)         Case: |P*P_ij| = R
(23)               Set T(P_ij) = 1
(24)               Add P_ij into B
(25)      }
(26)      Remove the P_ij from B if P_ij is not in V
(27)      Computes the S_ij of each P_ij in V* according to (2)
(28)      Re-heap maximal heap tree, Place the sensor at P*
(29) }
```

Algorithm 2: The enhanced JSP algorithm.

By limiting the area to update score, JSP can effectively reduce the time complexity from $O(n^4R^2)$ to $O(n^2R^4)$. The time complexity is $O(n^2)$ which is better than $O(n^2 \log n)$ of the DT-Score [16]. We will prove that JSP can use fewer numbers of deployed sensors than the DT-Score method in the simulation section. The enhanced JSP algorithm is given in Algorithm 2.

## 4. Simulation Results

This section shows the simulation results. The proposed JSP method is implemented using the C++ program. The MAX_AVG_COV method [13, 14], and the Delaunay triangulation method [16, 17]. The MAX and DT represent the MAX_AVG_COV method and the Delaunay triangulation method in the following figures. The MIN_AVG_COV method is not compared because its result is worse than that of the MAX_AVG_COV method. The evaluation methods include *the number of deployed sensors*, *effective coverage ratio (ECR)*, and distribution of *incremental coverage area (ICA)*.

*The number of deployed sensors* evaluates the efficiency of a deterministic deploying method when the size of region of interest is fixed. The method using fewer sensors to cover the region of interest is more efficient. *The effective coverage ratio* is the ratio that the maximal coverage area generated by all deployed sensors over the area of the region of interest. The method which generates less overlapping area between sensors will have small ECR. The ECR = 1 is the ideal case implying no overlapping on sensing area. In the real world, the circle-like sensing area of sensor must be overlapping with others. ECR = 1 can only be used as the reference value to evaluate the deploying methods. The ICA is the area which is covered for first time after $S$ *has* been placed. The results are represented as the percentage of the sensing area of a sensor. There are ten scales to evaluate this area size. They are 0%–10%, 10%–20%,. . ., and so on. The distribution of ICA can evaluate the deploying efficiency.

*4.1. Environment Setup.* In our simulation, the region of interest is 400 m × 400 m. The grid size is $1\,m^2$. Three scenarios are simulated as shown in Figure 4. The first scenario is an open-space. No obstacle is in the region of interest. The second scenario has a 200 m × 200 m rectangle obstacle in the center of the region of interest, and the third scenario has multiple obstacles irregularly placed in it. In each scenario, sensors with sensing radius 20 m, 30 m, 40 m, and 50 m are simulated. The sensing coverage of each sensor is assumed to be a perfect circle. The deploying procedure finishes when all grids in the simulation area are covered.

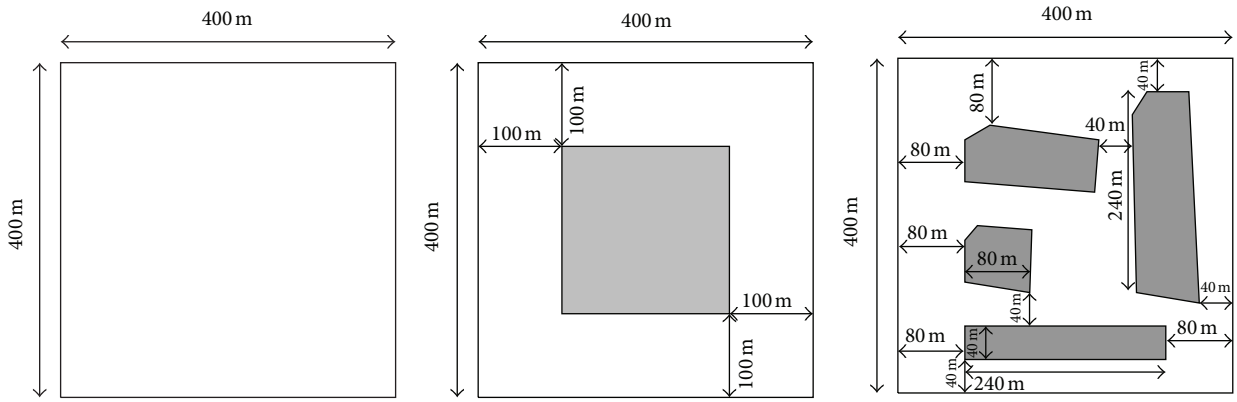
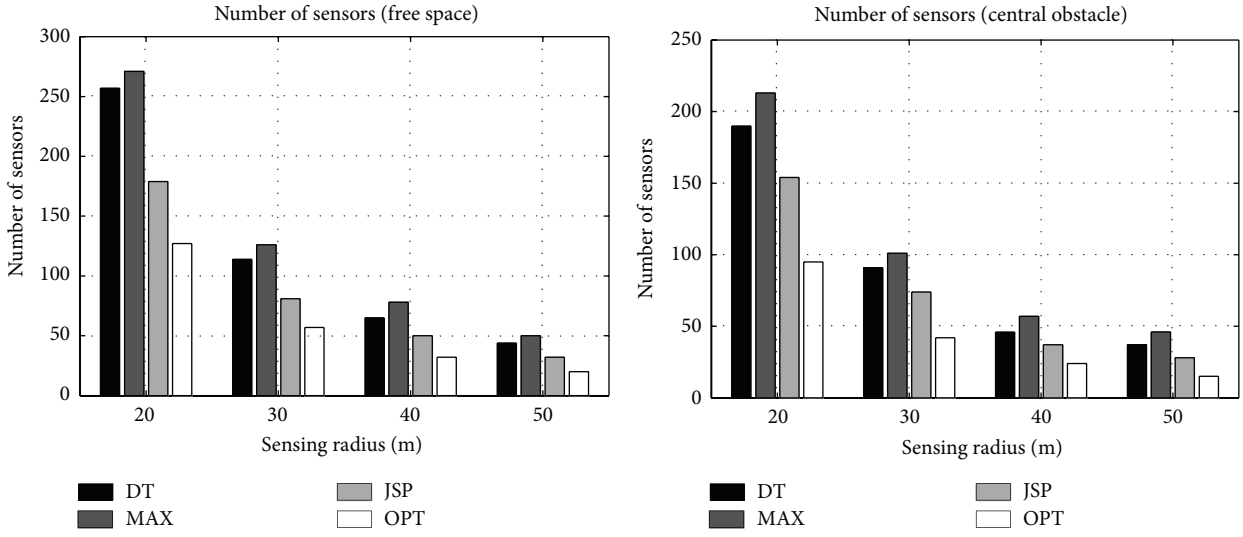



Figure 4: The simulation scenarios.



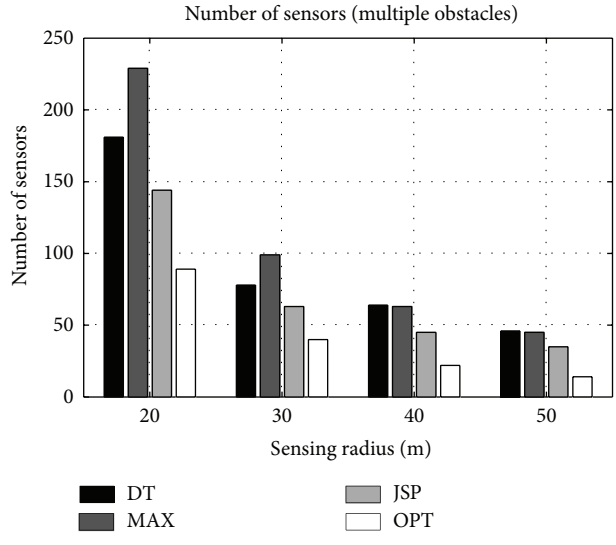
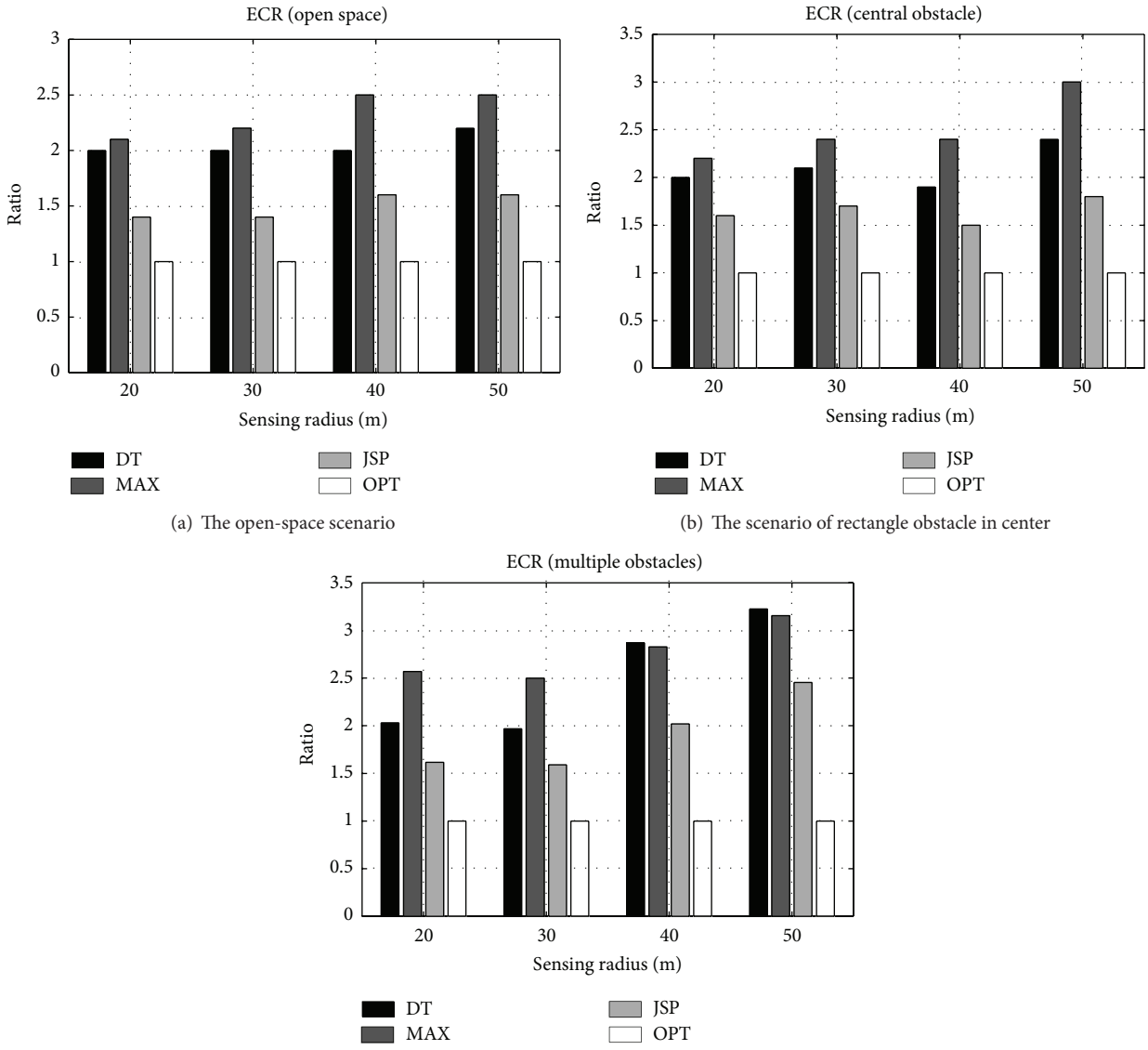




Figure 5: The number of deployed sensors in both scenarios.




(a) The open-space scenario

(b) The scenario of rectangle obstacle in center

(c) Multiple obstacles scenario

Figure 6: The effective coverage ratio.

*4.2. Numerical Results.* Figure 5 shows the number of sensors deployed to cover the whole space. The OPT represents the ideal case which simply divides the whole area size of the region of interest by the coverage area of a sensor. Because the sensing area of a sensor is a circle, the coverage areas of sensors must be overlapped. Thus, no method can use the same number of sensors to cover the whole region of interest as the OPT method. The OPT method is used as the lower bound of the number of deployed sensors.

In both simulated scenarios, the MAX method deploys most numbers of sensors to cover the whole space. This method introduces many small isolated regions when it greedily searches the position which can cover the largest uncovered space to place the next sensor. Thus, additional sensors are placed to cover those isolated regions. Increasing the sensing radius does not change the situation of the MAX method. The MAX method still has the worst result in any kind of sensor radius. More than 200 sensors are used to cover the whole space when sensing radius is 20 m in both scenarios.

The number of deployed sensors in the DT method is slightly better than the MAX method in both scenarios. The number of sensors of the DT method is less than the MAX method about 5% to 12% in the open-space scenario and about 10% to 20% in the obstacle scenario. Sensors in DT method are initially deployed along the contour of the region of interest. The DT method needs to add sensors one by one to cover the whole regions. The next location to add a sensor is the center of the circum-circle of the triangulation with the largest radius. Therefore, DT method is also similar to the MAX method that greedily places the next sensor to cover the maximum number of uncovered grids. However, placing

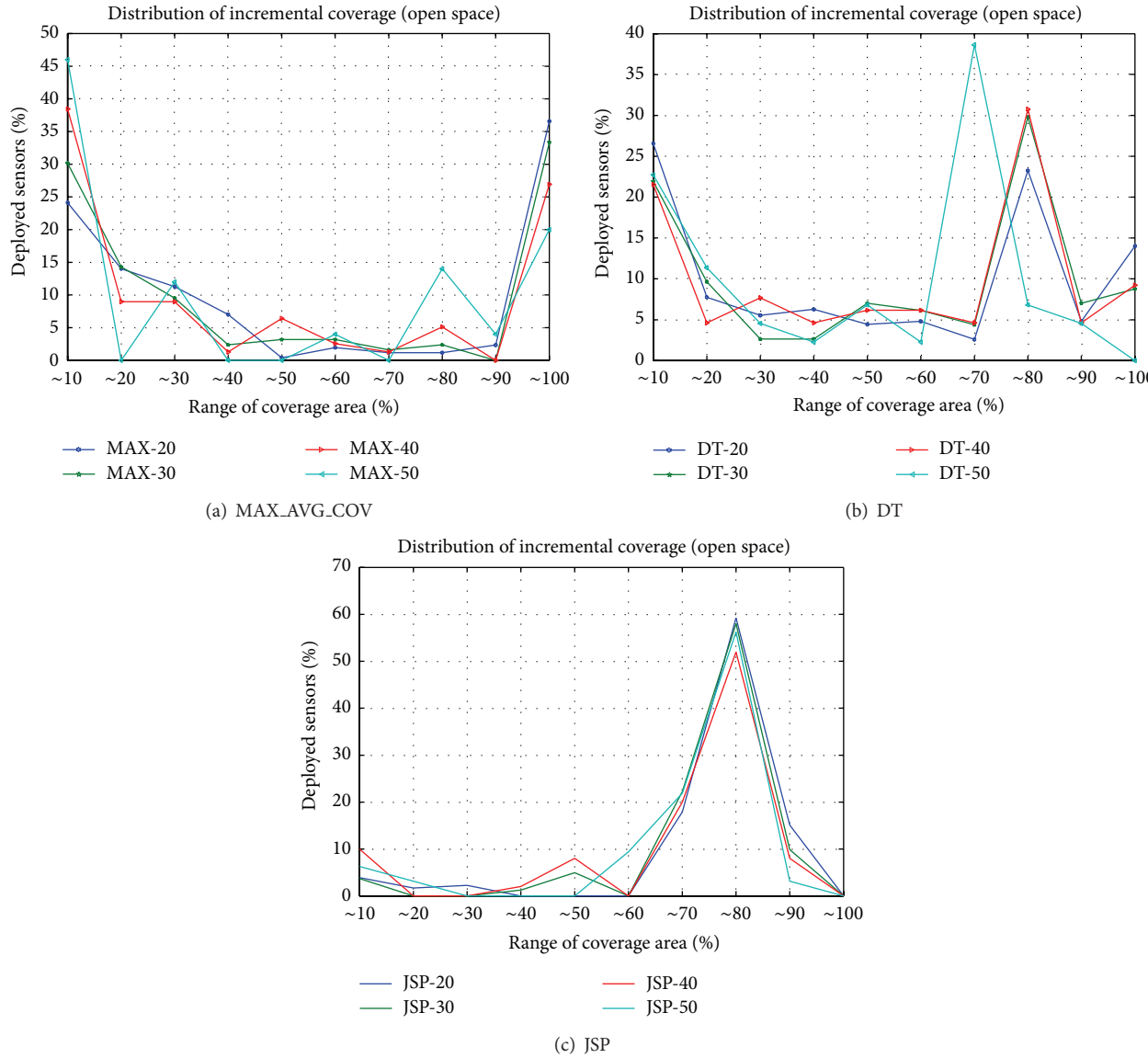



Figure 7: Distribution of ICA in open-space scenario.

the sensors along the boundaries of the obstacles prevents overlapping near the boundary of region and the obstacles. Thus, the results of DT method are better than those of the MAX method.

The JSP method exhibits better results than the MAX and DT methods in both scenarios. The number of deployed sensors of the JSP method is less than the MAX method about 34%–36% in the open-space scenario and about 28%–39% in the obstacle scenario. JSP also greedily includes the uncovered spaces, but it prevents dividing the uncovered space into isolated regions. In the open-space with sensing radius 20 m, JSP only needs 150 sensors to cover the whole space. The number of deployed sensors in the JSP method is about 1.5 times of the sensors used in the OPT method. In the obstacle scenario, the number of deployed sensors is still less than twice of that in OPT. The scenario with multiple obstacles is similar to the center obstacle scenario. More obstacles will also increase the initial deployed sensors of the DT method. More sensors are placed in the contour of the obstacles. As the sensing radius increases, the DT method still needs deployed sensors to enclose the obstacles. Therefore, the number of deployed sensors is close to that in the results of the MAX method.

Figure 6 shows the results of the ECR. The ECR of the OPT method is 1 which is the lower bound. The ECR of the MAX method is higher than 2.2 in both scenarios. In the obstacle scenario with sensing radius 50 m, the ECR of MAX is 3. This result implies that increasing the sensing radius generates more overlapping between sensors. When the space is divided into multiple isolated regions, large sensing radius will worsen the ECR. The ECR of the DT method is 2 in both scenarios. Placing sensors aside, the contour reduces the overlapping coverage near the boundary and obstacles. The ECR of JSP is less than 1.8. It can suppress the ECR

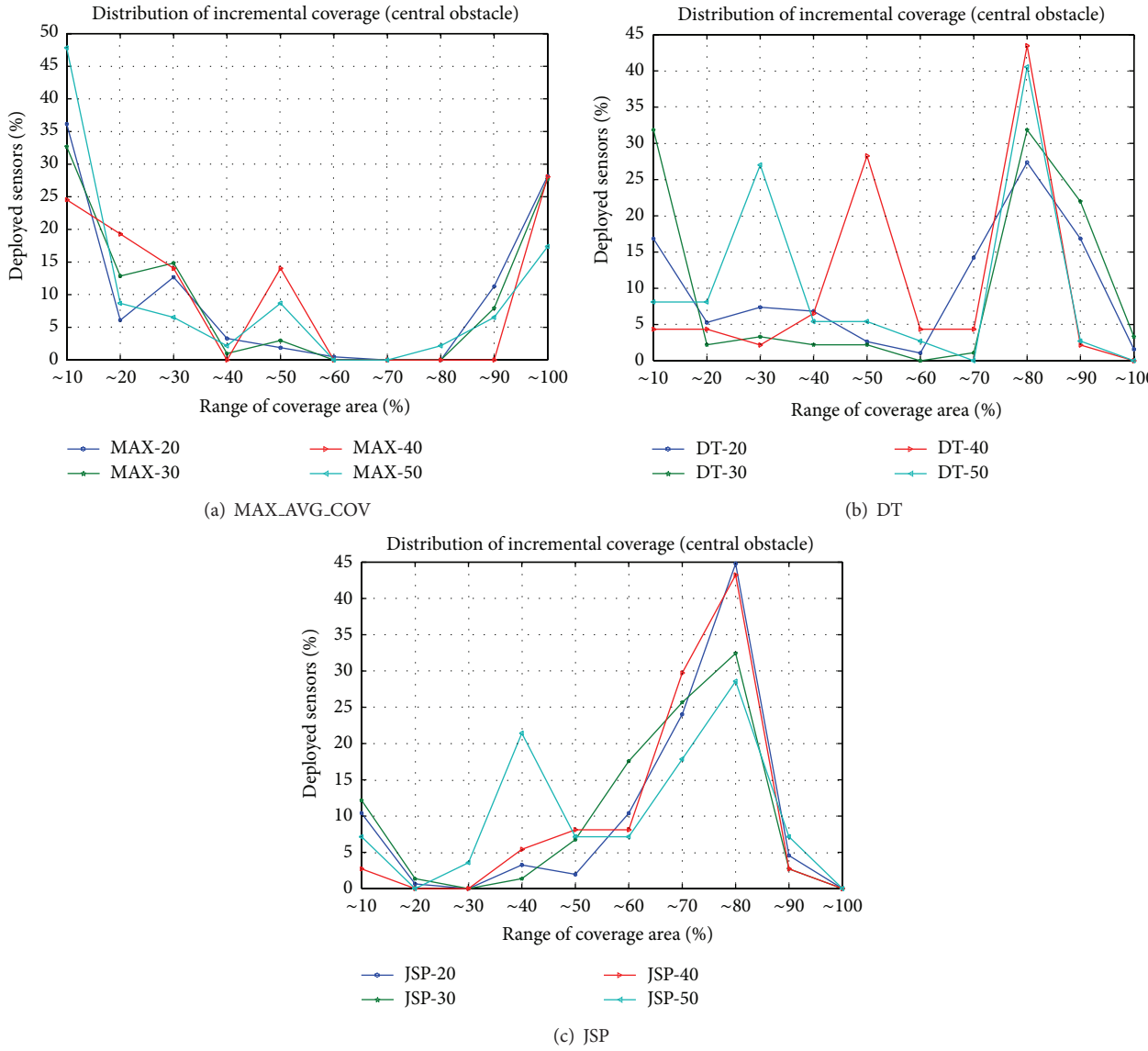



Figure 8: Distribution of ICA in central obstacle scenario.

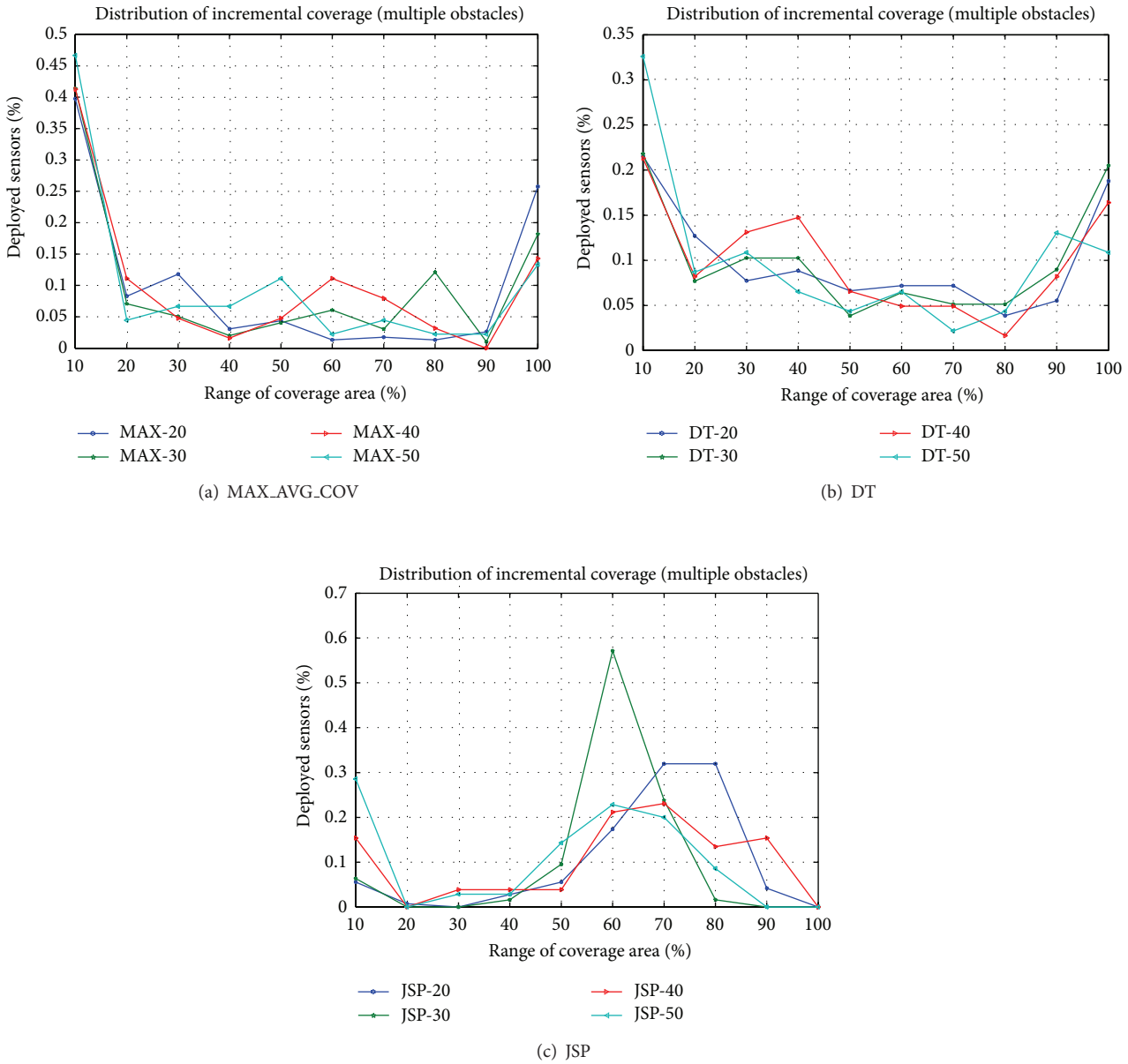
even increasing the sensing radius. In the scenario of multiple obstacles, the ECR of DT method in the cases of sensing radius 40 m and 50 m is worse than the MAX method. Large radius causes the initial deployed sensors which enclosed the obstacles to severely overlap their sensing areas with others. Therefore, the ECR is worse than the MAX method.

The distribution of ICA in the open-space scenario is shown in Figure 7. In the MAX method, the deployed sensors with ICA less than 20% sensing area are 40%–50% and the number of sensors with ICA more than 90% sensing area is 20%–40% as shown in Figure 7(a). The percentage of sensors with high ICA decreases when the length of sensing radius increases. It is related to the results in Figure 5. The MAX method deploys many sensors to cover the isolated regions.

The DT method improves the ICA but still many sensors are deployed to cover the small regions. The 25%–35% deployed sensors have their ICA less than 20% sensing area and 40%–50% sensors have their ICA more than 80% sensing area. Greedily placing the next sensor at the center of the circum-circle of the triangle with maximum radius introduces isolated regions as the MAX method. Deploying sensors near the contour of obstacles only moderates the problem suffered by the MAX method.

For the JSP, approximating 70% deployed sensors have their results on ICA more than 80%. The percentage of the deployed sensors with ICA smaller than 20% is less than 10% in all simulated sensing radii. The fluctuation of the ICA distribution is less than 10% when the sensing radius increases. The ICA with 70%–80% dominates the major portion of the deployed sensors. The results imply that JSP algorithm prevents deploying a sensor for covering a tiny uncovered space.

Figure 8 shows the ICA in the obstacle scenario. The distribution is similar to Figure 7. In all compared methods,




(a) MAX_AVG_COV

(b) DT

(c) JSP

Figure 9: Distribution of ICA in multiple obstacles scenario.

the distribution of ICA has explicit fluctuations when sensing radius is more than 40 m. Large sensing radius includes more area occupied by the obstacle. Thus, the distribution has great variation. However, the JSP and the MAX can still retain the distribution trend as shown in Figure 7. For the DT method, the obstacle limits the organization of the Delaunay triangulations. Therefore, the DT method exhibits different ICA distribution.

Figure 9 shows the ICA of the scenario with multiple obstacles. The MAX method and the JSP method still retain the distributions as the Figures 7 and 8. However, the distribution of the DT method has changed. Most of the sensors are used to cover a tiny area. The deploying efficiency of the DT method greatly decays in the scenario with multiple obstacles.

## 5. Conclusions

This paper proposed a Jigsaw-based sensor deploying (JSP) algorithm for wireless sensor network. Sensors are placed at the periphery of the region of interest. The JSP method prevents dividing the uncovered space into isolated regions and uses fewer numbers of sensors to cover the whole region of interest. The time complexity of the enhanced JSP algorithm is $O(n^2R^4)$ which is better than that in MAX_AVG_COV and not worse than the DT-Score method. The number of deployed sensors of the JSP method is less than the MAX_AVG_COV method about 34%–36% in the open-space scenario and about 28%–39% in the obstacle scenario. The effective coverage ratio of JSP is less than 1.8 instead of more than 2 in the DT-Score and the MAX_AVG_COV.

The distribution of incremental coverage area (ICA) is close to normal distribution. Sensors are not deployed to cover tiny regions.

## References


[1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, September 2002.

[2] J. D. Kenney, D. R. Poole, G. C. Willden et al., "Precise positioning with wireless sensor nodes: monitoring natural hazards in all terrains," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 722–727, October 2009.

[3] J. Zhang, W. Li, Z. Yin, S. Liu, and X. Guo, "Forest fire detection system based on wireless sensor network," in *Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA '09)*, pp. 520–523, May 2009.

[4] L. Qian, J. Fuller, and I. Chang, "Quickest detection of nuclear radiation using a sensor network," in *Proceedings of IEEE Conference on Technologies for Homeland Security (HST '12)*, pp. 648–653, November 2012.

[5] T. Gao, C. Pesto, L. Selavo et al., "Wireless medical sensor networks in emergency response: implementation and pilot results," in *Proceedings of IEEE International Conference on Technologies for Homeland Security (HST '08)*, pp. 187–192, May 2008.

[6] Y.-L. Tsou and S. Berber, "Design, development and testing of a wireless sensor network for medical applications," in *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC '11)*, pp. 826–830, July 2011.

[7] Y.-R. Tsai and Y.-J. Tsai, "Sub-optimal step-by-step node deployment algorithm for user localization in wireless sensor networks," in *Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC '08)*, pp. 114–121, June 2008.

[8] X. Wang, X. Wang, G. Xing, J. Chen, C.-X. Lin, and Y. Chen, "Towards optimal sensor placement for hot server detection in data centers," in *Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS '11)*, pp. 899–908, July 2011.

[9] K. Xu, H. Hassanein, G. Takahara, and Q. Wang, "Differential random deployment for sensing coverage in wireless sensor networks," in *Proceedings of the Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–5, December 2006.

[10] K. Xu, H. Hassanein, G. Takahara, and Q. Wang, "Relay node deployment strategies in heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 2, pp. 145–159, 2010.

[11] S. M. A. Salehizadeh, A. Dirafzoon, M. B. Menhaj, and A. Afshar, "Coverage in wireless sensor networks based on individual particle optimization," in *Proceedings of the International Conference on Networking, Sensing and Control (ICNSC '10)*, pp. 501–506, April 2010.

[12] S.-C. Huang, "Ion-6: a positionless self-deploying method for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 940920, 10 pages, 2012.

[13] S. S. Dhillon K and K. Chakrabarty, "Placement for effective coverage and surveillance in distributed sensor networks," in *Proceedings of IEEE International Conference on Wireless Communications and Networking (WCNC '03)*, vol. 3, pp. 1609–1614, March 2003.

[14] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," in *Proceedings of the 5th International Conference on Information Fusion*, vol. 2, pp. 1581–1587, July 2002.

[15] X. Xu and S. Sahni, "Approximation algorithms for sensor deployment," *IEEE Transactions on Computers*, vol. 56, no. 12, pp. 1681–1695, 2007.

[16] C.-H. Wu, K.-C. Lee, and Y.-C. Chung, "A Delaunay triangulation based method for wireless sensor network deployment," in *Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS '06)*, pp. 253–260, July 2006.

[17] J. Wu and J. Zhu, "Sensor node optimal placement algorithm based on coverage rates," in *Proceedings of the International Conference on Computer Science and Service System (CSSS '11)*, pp. 2547–2550, June 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 497264, 9 pages
http://dx.doi.org/10.1155/2013/497264

*Research Article*

# A Faster Convergence Artificial Bee Colony Algorithm in Sensor Deployment for Wireless Sensor Networks

**Xiangyu Yu,[1] Jiaxin Zhang,[1] Jiaru Fan,[1] and Tao Zhang[2]**

[1] *School of Electronic and Information Engineering, South China University of Technology, Guangzhou, Guangdong 510640, China*
[2] *School of Electronic and Communication Engineering, Guiyang University, Guiyang, Guizhou 550005, China*

Correspondence should be addressed to Xiangyu Yu; yuxy@scut.edu.cn

Received 27 June 2013; Revised 26 August 2013; Accepted 28 August 2013

Academic Editor: Chang Wu Yu



In wireless sensor networks (WSN), sensor deployment is one of the main topics for enhancing the sensor's coverage rate. In this paper, by modifying updating equation of onlooker bee and scout bee of original artificial bee colony (ABC) algorithm, a sensor deployment algorithm based on the modified ABC algorithm is proposed. Some new parameters such as forgetting and neighbor factor for accelerating the convergence speed and probability of mutant for maximizing the coverage rate are introduced. Simulation results showed that comparing with the deployment method based on original ABC and particle swarm optimization (PSO) algorithm, the proposed approach can achieve a better performance in coverage rate and convergence speed while needing a less total moving distance of sensors.

## 1. Introduction

Integrating with sensor technology, distributed information process, embedded technology, wireless communication and microelectronic technique, and so forth, wireless sensor networks (WSN) have become active fields of research. Because of their advantages in low energy consumption, distributed self-organization ability, WSN are extensively used in fields such as target tracking, environment monitoring and national defense, and underwater detecting [1, 2].

Due to its relation with energy saving, connectivity, and network reconfiguration, coverage is an important issue in WSN. It mainly addresses how to deploy the sensors to achieve sufficient coverage of the service area so that each position in the service area is monitored at least by one sensor. A good coverage is indispensable for the effectiveness of WSN. An efficient deployment of sensors will reduce the configuration and communication consumption of the network and improve the resource management; thus sensor deployment becomes a substantial work [3, 4].

Sensor deployment algorithms can be divided into static or dynamic ones. In some cases such as underwater, battlefield, and abominable environment where it is hard to arrange sensors manually, only dynamic deployment approaches can be applied. In dynamic deployment, sensors are initially located with random coordinates and can change their positions by using their knowledge of other's positions and other information collected. In order to increase the coverage rate of the whole network, many approaches have been proposed for dynamic node deployment, such as potential [5], virtual force [6–8], Voronoi diagram [9], and bionic intelligence algorithm [10–16].

In fact, the object of sensor deployment is to seek an optimal sensor placement, which has a resemblance on the bionic intelligence algorithm. Bionic intelligence algorithms are inspired by the social behavior of different animals or species and have become a research of interest in different domains for solving complex optimization problems. In order to ensure the effectiveness and the quality of the WSN, bionic intelligence and swarm intelligence algorithms are introduced in sensor deployment, for instance, the genetic algorithm (GA) [10], the particle swarm optimization (PSO) [11–13], the ant colony optimization (ACO), the glowworm (GSO) [14], and so forth. Among which, ABC-based deployment algorithm has been proved to be competitive with some conventional optimization algorithms [15, 16].

Based on the foraging behavior of honey bee swarms, ABC algorithm was proposed to optimize multivariable and multimodal continuous functions. It is found several applications in electrical engineering, mechanical and civil engineering areas, electronics, software and control engineering areas, image processing area, and so forth [17]. The original ABC algorithm, however, has low convergence speed and is easily trapped into the local optimum when solving complex multimodal problems. Therefore, a few modified versions were proposed by many researchers to improve it [18–23].

In this paper, a sensor deployment algorithm based on modified ABC algorithm called FNF, (forgetting and neighbor factor)-BL (backward learning) ABC algorithm is proposed. In order to have a better coverage and a faster convergence speed, the onlooker bee phase and the scout bee phase of the ABC algorithm have been modified. In onlooker bee phase, a modification called FNF, which introduced forgetting factor and neighbor factor into the original equation, is adopted. The scout bee phase is all new designed by changing the original method into the BL method.

The rest of this paper is organized as follows: original ABC algorithm and its application in sensor deployment algorithm are described in Section 2; our modifications of ABC algorithm are presented in Section 3; experiment results and analysis are shown in Section 4; some conclusions are drawn and our future works are discussed in Section 5.

## 2. ABC Algorithm and Its Application in Deployment

*2.1. ABC Algorithm.* In ABC algorithm, the colony of artificial bees consists of three groups of bees: employed bees, onlooker bees, and scout bees. The food searching of bees is collectively performed by these three kinds of bees. The employed bees search the food around the food sources in their memory and deliver the food information, such as the fitness value, to the onlooker bees. The onlooker bees select good food sources from this information and search the food around the selections for a better one. The scout bees are translated from a few employed bees, which abandon their food sources and search new ones. Each cycle of the searching consists of three phases, as describe below.

At initiation, food sources are randomly generated by:

$$x_{il} = x_l^{\min} + \left(x_l^{\max} - x_l^{\min}\right) \times \text{rand}\,(0,1)\,, \tag{1}$$

where $x_l^{\max}$ and $x_l^{\min}$ are the lower and upper bounds of the position and rand$(0, 1)$ is the random number in the interval $(0, 1)$.

Then at the employed bee phase (first phase), a neighbor of the current food source $k$ is randomly selected for the employed bee and then a new solution is produced, and its fitness value is calculated. For each employed bee, a candidate food source solution from the previous one is generated by:

$$V_{il} = x_{il} + \text{rand}\,(-1,1) \times \left(x_{il} - x_{kl}\right), \tag{2}$$

where $l \in \{1, 2, \ldots M\}$, $M$ is the number of variables (problem dimension) to be optimized, $i, k \in \{1, 2, \ldots J\}$, and $i \neq k$, $J$ is the number of employed bees (the number of food sources), and rand$(-1, 1)$ is a random number between $(-1, 1)$.

After comparing the fitness value of the new food source (solution) with the previous one, the employed bee memorizes the one which has higher fitness value. After all employed bees complete the searching procedure, they translate the fitness values of the food sources (solutions) and their positions information to the onlooker bees in the hive.

At the onlooker bee phase (second phase), for each onlooker bee, a food source is inherited depending on the probability value associated with that food source, $p_i$, calculated by:

$$p_i = \frac{\text{fitness}\left(\mathbf{x}_i\right)}{\sum_{i=1}^{S} \text{fitness}\left(\mathbf{x}_i\right)}. \tag{3}$$

After an onlooker bee reaches a food source, it looks for a new source within the neighborhood of the previous one and memorizes the food sources according to their fitness value. The update process used in the onlooker bee phase is the same as that in the employed bee phase, refering to (2). The main distinction between the employed bee phase and the onlooker phase is that every solution in the employed bee phase involves the update process, but only the selected solutions depending on the probability value have the opportunity to update in the onlooker bee phase.

The selection of the scout bee is controlled by a control parameter called "limit," which is a predetermined number of trials. If a solution representing a food source cannot be improved by the number of "limit," the source is considered to be exhausted and then the employed bee of this food source becomes a scout. The position of the abandoned food source is replaced with a random search by (1). So "limit" controls the selection of the scout bee and the qualities of solutions. This is the final phase of the ABC algorithm.

These three phases are repeated until the value of cycle is up to maximum cycle number (*MaxCycle*).

*2.2. ABC Algorithm in Deployment Problem.* The object of sensor deployment algorithm is to determine an optimum sensor distribution in an area of interest. So applying ABC algorithm into sensor deployment problem is possible. For each sensor in WSN there is a sensing range, which indicates the sensing ability of the sensor, and a communication range, which refers to the longest distance through which two sensors can exchange information with each other. These two ranges restraint the performance of a deployment algorithm. In this paper, there are three assumptions: first, sensing radius of the sensors indicating the sensing ability is same; second, all of the sensors are interconnected; and finally, all of the sensors can move. Table 1 shows the relationship between the parameters in ABC algorithm and the problem of sensor deployment. Notice that in this paper the coverage rate is set as the reciprocal of the fitness of food source fitness. So the optimum sensor distribution corresponds to the lowest fitness. Simulation results in [16] have verified that the ABC algorithm is more successful than the PSO algorithm for the dynamic deployment problem of WSN using a probabilistic detection model. Deployments found by ABC are better than

Table 1: Corresponding parameters in ABC algorithm and sensor deployment.

| ABC algorithm | Sensor deployment problem |
|---|---|
| Solution of a food source | Sensor distribution |
| $M$ dimensions in each solution | $M$ sensors' coordinate |
| Fitness of the solution | Coverage rate of interest area |
| Lowest fitness | Optimum sensor distribution |

those by PSO for all of the 30 independent runs which are started with the same initial deployment in that literature.

## 3. Modified ABC Algorithm and Its Application in Deployment

In sensor deployment, coverage and convergence are very important issues. In order to achieve the higher coverage rate and faster convergence speed, two modifications are introduced in our FNF-BL-ABC-based approach.

*3.1. Modification in Onlooker Bee Phase.* The original food source updating mechanism is calculated by (2). Because of the neighbor food source is randomly chosen, the fitness of the neighbor food source may be higher, which means the sensors distribution is worse than the current one. In addition, the updating equation remains unchanged as the iteration processes. Thus, this original updating mechanism cannot help bee to find the optimum food sources effectively, which results in low speed of achieving the optimum solution. In order to accelerate the whole procedure, two parameters named forgetting factor and neighbor factor [22] are introduced into (2):

$$v_{il} = \eta \times x_{il} + \tau \times \text{rand}\,(-1, 1) \times \left(x_{il} - x_{kl}\right), \tag{4}$$

where $\tau$ is the forgetting factor and $\eta$ is the neighbor factor.

Among these two parameters, the value of $\eta$ is related to the fitness of neighbor food source. The variation tendency of the value of neighbor factor is increased to prevent the global searching ability from being lost. The neighbor factor $\eta$ can strengthen or weaken the relation between present and neighbor food source. The forgetting factor $\tau$ stands for the intension of present food source memory when searching for the next food source. In order to fully use the neighbor information and find the global best, the forgetting factor is dynamically decreased as

$$\tau = \lambda \times w_{\tau}. \tag{5}$$

The value of neighbor factor $\eta$ is related to the fitness of neighbor food source as in

$$\eta = \lambda \times w_{\eta}. \tag{6}$$

In (5) and (6), when the fitness value of neighbor food source is lower than the present one, $\lambda > 1$; when the fitness value of neighbor food source is bigger than the present one, $\lambda < 1$. Consider

$$w_{\eta} = w_4 - \left(\frac{\text{maxcycle} - \text{iter}}{\text{maxcycle}}\right)^{\alpha} * \left(w_4 - w_3\right), \tag{7}$$

$$w_{\tau} = w_1 + \left(\frac{\text{maxcycle} - \text{iter}}{\text{maxcycle}}\right)^{\beta} * \left(w_2 - w_1\right). \tag{8}$$

In (7) and (8), $w_1$, $w_2$, $w_3$, and $w_4$ are constants. The interval of $\alpha$ and $\beta$ is $[0.8, 1]$ and $[1, 1.2]$. The parameter $w_{\eta}$ becomes larger from $w_3$ to $w_4$, and $w_{\tau}$ is reduced from $w_2$ as $w_1$ with the iteration goes.

*3.2. Modification in Scout Bee Phase.* In original ABC algorithm, the work of scout bee is to abandon the food source remaining unchanged by the "limit" time and randomly generating a food source for avoiding trapping into local optimum. The scout bee transforms to employed bee. The performance of deployment algorithm is affected by the parameter "limit." In order to reach a high coverage rate, the value of "limit" is important. However, it needs numerous experiments to find an appropriate value. Thus, a modification called backward learning [20] is introduced into the ABC algorithm. This approach can be described as shown in Pseudocode 1.

The following equation can calculate the value of probability of mutant (PM):

$$\text{PM} = 0.01 + 0.1 \times \left(2 - e^{\text{iter} * \ln 2/\text{maxcycle}}\right). \tag{9}$$

The backward learning strategy can be expressed as

$$v_{il} = x_{wl} + x_{bl} + \text{rand}\,(0, 1) \times x_{il}, \tag{10}$$

where $x_w$, and $x_b$ are the worst and best quality food source. Record $v_i$ if it is better than $x_i$.

## 4. Simulation Results

PSO and ABC are two most popular swarm intelligence algorithms; they both simulate the collective behavior of decentralized, self-organized systemss and have many idea in common, so in many research work the simulation results of ABC algorithm are compared with those of the PSO algorithm, which is also taken in this paper.

In this section, PSO algorithm and its parameters are first presented in detail, followed by explanations how the parameters in our proposed algorithm are chosen. Two sets of simulations are made to verify the effectiveness of the proposed approach. The first is an ideal case in a square room; the second is in a general room with obstacles. Three kinds of approaches, which are original ABC-based, FNF-BL-ABC-based, and PSO-based deployment, are compared. To guarantee the comparability between them, the experimental circumstances are identical consisting of the initial deployment and iteration time.

```
For i = 1 : n (n stands for the number of food source)
        If rand < probability of mutant (PM)
                backward learning.
        end
end
```

Pseudocode 1: Pseudocode of backward learning method.

### 4.1. Parameters Selection

*4.1.1. Parameters Selection for PSO Algorithm.* A standard version of PSO is applied in the sensor deployment in [24]. In PSO algorithm, several individuals are "evolved" by learning the experience from their own and companions through generation. Each individual is called "particle." Each particle also has a $l$-dimension space, which is similar to the food source in the ABC algorithm. Each particle is also randomly generated with (1).

The updating processing of the particle in PSO algorithm is divided into two phases.

In the first phase, the velocity is calculated as

$$v_{il} = v_{il} + c_1 * \text{rand1}\,() * \left(p_{il} - x_{il}\right) + c_2 * \text{rand2}\,() * \left(p_{gl} - x_{il}\right). \tag{11}$$

In the second phase, the position of the particle is updated as (12)

$$x_{il} = x_{il} + v_{il}. \tag{12}$$

There are some parameters that should be explained in these equations: rand( ), rand1( ), and rand2( ) are three numbers independently generated by the function "rand( )" in MATLAB. $p_{il}$ is the best position of the $i$th particle, and $p_{gl}$ is the best position of the particles in the neighbor of $i$th particle. $x_{il}$ is the $i$th particle's position and $v_{il}$ is its velocity. Based on [13, 24, 25], $c_1$ and $c_2$ are two positive numbers which can influence the performance of the algorithm and are calculated by

$$c_1 = c_{1\_\max} - \left(c_{1_{\max}} - c_{1_{\min}}\right) * \frac{t}{\text{gen}},$$
$$c_2 = c_{2\_\max} - \left(c_{2_{\max}} - c_{2_{\min}}\right) * \frac{t}{\text{gen}}, \tag{13}$$

where $c_{1\_\max}$, $c_{1\_\min}$, $c_{2\_\max}$, $c_{2\_\min}$ are parameters set to change the performance of the algorithm. $t$ is the iteration times, and *gen* is the maximum iteration times.

In this paper, 11 combinations of $c_{1\_\max}$, $c_{1\_\min}$, $c_{2\_\max}$, and $c_{2\_\min}$ have been tried to improve the performance in the optimizer processing. In this set of experiments, both $c_{1\_\min}$ and $c_{2\_\min}$ are invariable, equal to 1. The threshold of velocity (*max_velocity*) equals 6. The experiment results are shown in Table 2. The eighth experiment, in which $c_{1\_\max}$ and $c_{2\_\max}$ equal 3.4, has the best performance among the 11 combinations. Thus, in the comparison with FNF-BL-ABC and original ABC algorithm, the parameters in PSO are chosen as $c_{1\_\max} = 3.4$, $c_{1\_\min} = 1$, $c_{2\_\max} = 3.4$, $c_{2\_\min} = 1$, and the threshold of velocity (*max_velocity*) = 6.

Table 2: Experiment results with different values of parameters in PSO algorithm.

| Combination | | Coverage rate |
|---|---|---|
| $c_{1\_\max} = 2$ | $c_{2\_\max} = 2$ | 0.9035 |
| $c_{1\_\max} = 2.2$ | $c_{2\_\max} = 2.2$ | 0.8972 |
| $c_{1\_\max} = 2.4$ | $c_{2\_\max} = 2.4$ | 0.9059 |
| $c_{1\_\max} = 2.6$ | $c_{2\_\max} = 2.6$ | 0.9136 |
| $c_{1\_\max} = 2.8$ | $c_{2\_\max} = 2.8$ | 0.9136 |
| $c_{1\_\max} = 3$ | $c_{2\_\max} = 3$ | 0.9253 |
| $c_{1\_\max} = 3.2$ | $c_{2\_\max} = 3.2$ | 0.9369 |
| $c_{1\_\max} = 3.4$ | $c_{2\_\max} = 3.4$ | 0.9466 |
| $c_{1\_\max} = 3.6$ | $c_{2\_\max} = 3.6$ | 0.9460 |
| $c_{1\_\max} = 3.8$ | $c_{2\_\max} = 3.8$ | 0.9253 |
| $c_{1\_\max} = 4$ | $c_{2\_\max} = 4$ | 0.8910 |

*4.1.2. Parameter Selection for the Proposed Algorithm.* As for the proposed algorithm, there are three different sets of parameters:

(i) $w_i$ $(i = 1, \ldots, 4)$,

(ii) $\lambda$,

(iii) $\alpha$, $\beta$.

In this paper, we set $\alpha = 0.8$ and $\beta = 1.2$ for the reason that has been described in [21]. So a large number of tests have been made for the first two parameter sets. Among all the tests for $w_i$, the best 5 combinations are (1) $w_1 = w_3 = 0.2$ and $w_2 = w_4 = 1.2$; (2) $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 1.6$; (3) $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 2.0$; (4) $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 2.4$; (5) $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 2.8$. Among which, the simulation results of combinations 2 and 3 are better in convergence speed than the one $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 1.2$ given in [21], while combination 4 and 5 have almost the same performance as that in [21]. So in our simulations, we set $w_1 = w_3 = 0.2$ & $w_2 = w_4 = 1.6$, and this set of parameters is used for choosing $\lambda$. As for parameter $\lambda$, after numerous experiments, the best six sets are, from the best to the worst, (1) 1.4 or 0.6; (2) 1.3 or 0.7; (3) 1.2 or 0.8 (which is used in [21]); (4) 1.5 or 0.5; (5) 1.6 or 0.4; (6) 1.7 or 0.3 in (6) and (7), and we choose (1) in our simulation for a better performance in convergence speed.

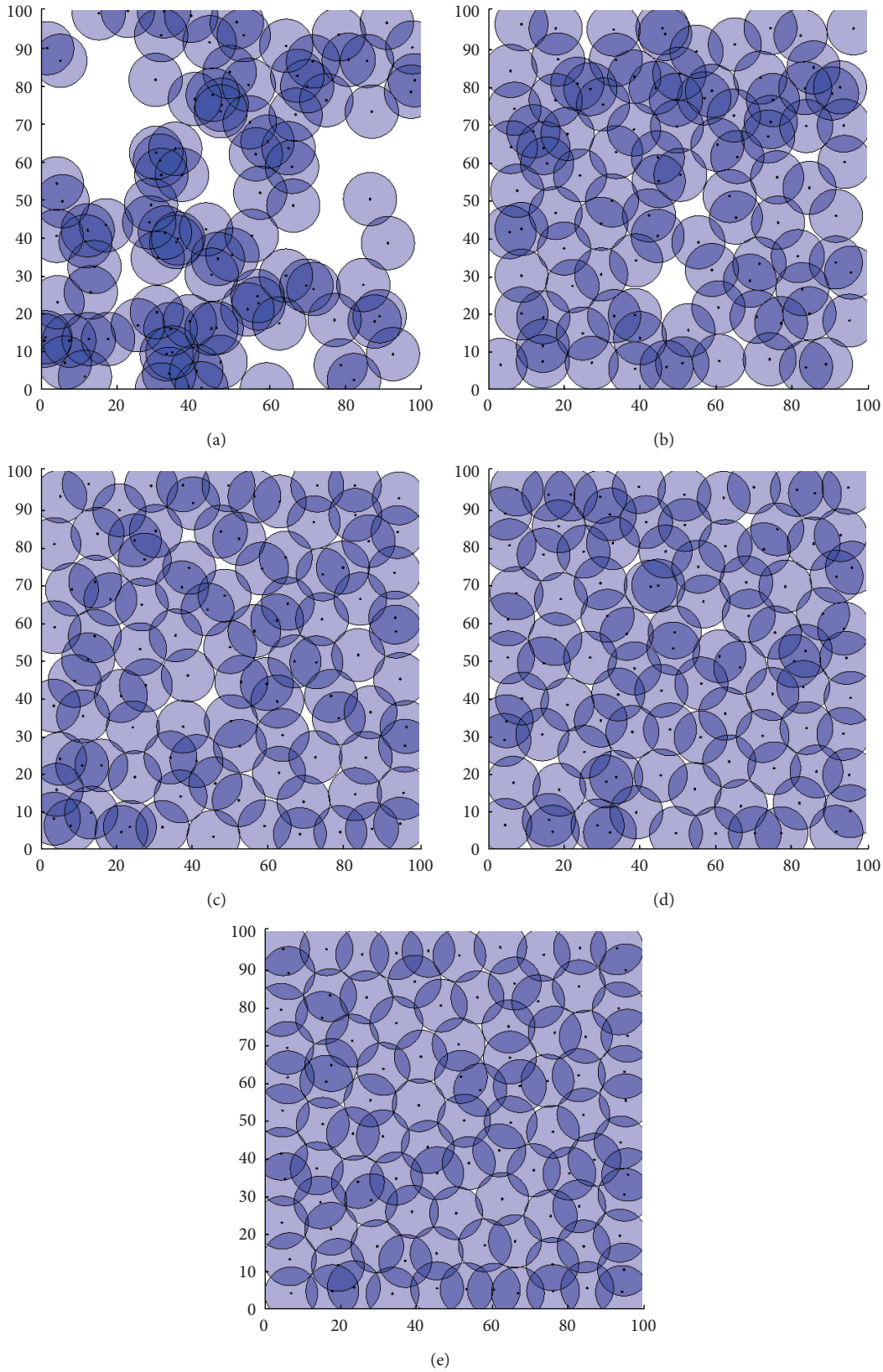



Figure 1: Sensor distribution: (a) initial, (b) 100 iterations, (c) 500 iterations, (d) 1000 iterations, and (e) 10000 iterations.

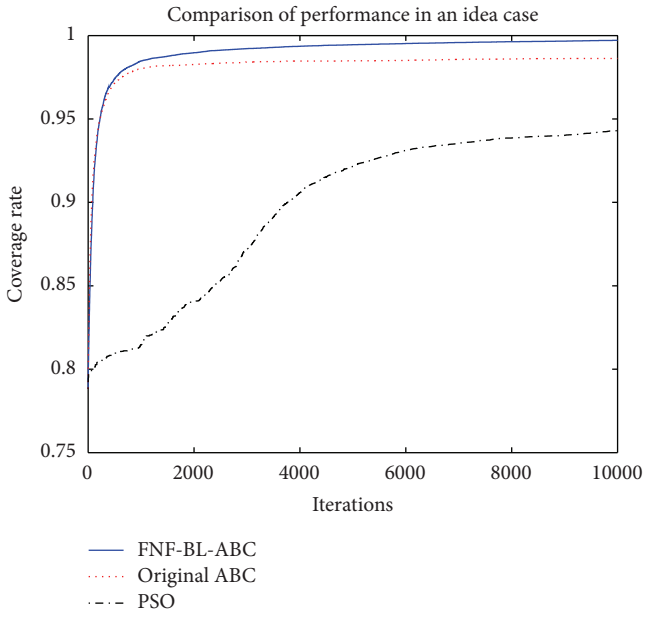



Figure 2: Coverage rate comparison between FNF-BL-ABC, original ABC, and PSO for an ideal room.

*4.2. Simulations in an Ideal Room.* In order to make the simulation more reliable, the parameters such as the number of sensors and their sensing radius are the same as those in [16]. There are 100 mobile sensors in our simulation. The sensing radius of each sensor $r$ is 7. The proportion of target area is $100 \times 100$. The max cycle is 10000. The colony size is 20. The run times are 20, and we calculate the average to make the result more reliable and a higher robustness. In the original ABC algorithm, there is a parameter called "limit" which needs to be set, however, since the scout bee phase is modified, we discard this parameter and the "limit" is dispensable.

Figure 1 shows the distribution of initial, 100 iterations, 500 iterations, 1000 iterations, and 10000 iterations. From these figures we can find that the proposed approach can cover the entire area effectively.

The performance of original ABC-based, FNF-BL-ABC-based and PSO-based deployment algorithm is compared in Figure 2. Each algorithm is based on the same initial deployment as in Figure 1.

As can be seen in Figure 2, at iteration 10000, the coverage rate is 99.71% for the proposed FNF-BL-ABC approach, which is larger than the 98.63% of the original ABC-based one and 94.31% of the PSO-based one, which showed that the proposed approach has better coverage rate than the other two. Meanwhile, we can also find that proposed one has faster convergence speed than the other two in Figure 2 and Table 3 and the effectiveness of the modifications is verified.

In (2), the first part is the information of the current food source. The second part is the different information between the current and neighbor food source, because the neighbor is selected randomly, which means the fitness of them is not considered. Moreover, the iteration number of times is also ignored by the food source updating process. These aspects will reduce the convergence speed. Equation (2) is replaced by (4). In (5) and (6), the values of $\tau$ and $\eta$ are associated with the comparison between the fitness of the neighbor and current food source. They are also affected by the iteration number of times in (8) and (9).

Table 3: Deployment results comparison between 3 algorithms at some iterations.

| Number of iterations | 100 | 500 | 2000 | 5000 | 10000 |
|---|---|---|---|---|---|
| FNF-BL-ABC | 0.9095 | 0.9740 | 0.9897 | 0.9945 | 0.9971 |
| Original ABC | 0.9184 | 0.9713 | 0.9826 | 0.9848 | 0.9863 |
| PSO | 0.7993 | 0.8097 | 0.8403 | 0.9215 | 0.9431 |

In the later stage of the whole algorithm, each food source has a better quality than before. However, some or all of them may be lost in the local optimum. In (9), the probability of mutant is increased with the process of the algorithm. The original method in (1) is altered by the backward learning method in (10). This backward learning method, which replaces the original solution by its relative solution, can acquire a better evaluation than the random solution. This conclusion has been testified in [26]. For one, the convergence speed can be increased immensely. For other, the local optimum can be somewhat avoided. Thus, the modified version of ABC algorithm can achieve a better performance in convergence.

In the practical use of WSN, power consumption must be considered, which is mainly in two ways: transmission/receive signal and movement. The energy consumption in moving is related to the total moving distance of all nodes. In our implement, we introduce the concept of "virtual move," which means that sensors do not move at each step, but move to the destination position after the final iteration only once. We record the total moving distance for each sensor from initial to final positions; for the original ABC-based algorithm, the total moving distance is 5216.1 while, for the proposed approach, the value is 3541.6. As for PSO-based algorithm, since it has worst coverage rate, we do not compare its moving distance here. Thus, we can make a conclusion that the energy consumption of the proposed approach is better than that of original ABC-based algorithm.

*4.3. Simulations in a Room with Obstacles.* In order to verify the effectiveness of the proposed approach in general case, there are two obstacle areas in the $100 \times 100$ room, each is $30 \times 20$, and there are 90 sensors to be placed in this room. The parameters for each sensor are the same as those in Section 4.1. Figure 3 shows the sensor distribution of initial and after 10000 iterations for both the proposed approach and the original ABC algorithm.

Because the performance of PSO algorithm is not good in Figure 2, only the comparison between FNF-BL version and original-ABC algorithm is provided here and shown in Figure 4.

From Figure 4 we can find that the proposed approach has higher coverage rate than original ABC. For this case with obstacles, the total moving distance of all sensors is 3817.8 for the proposed approach and 4870.4 for the original

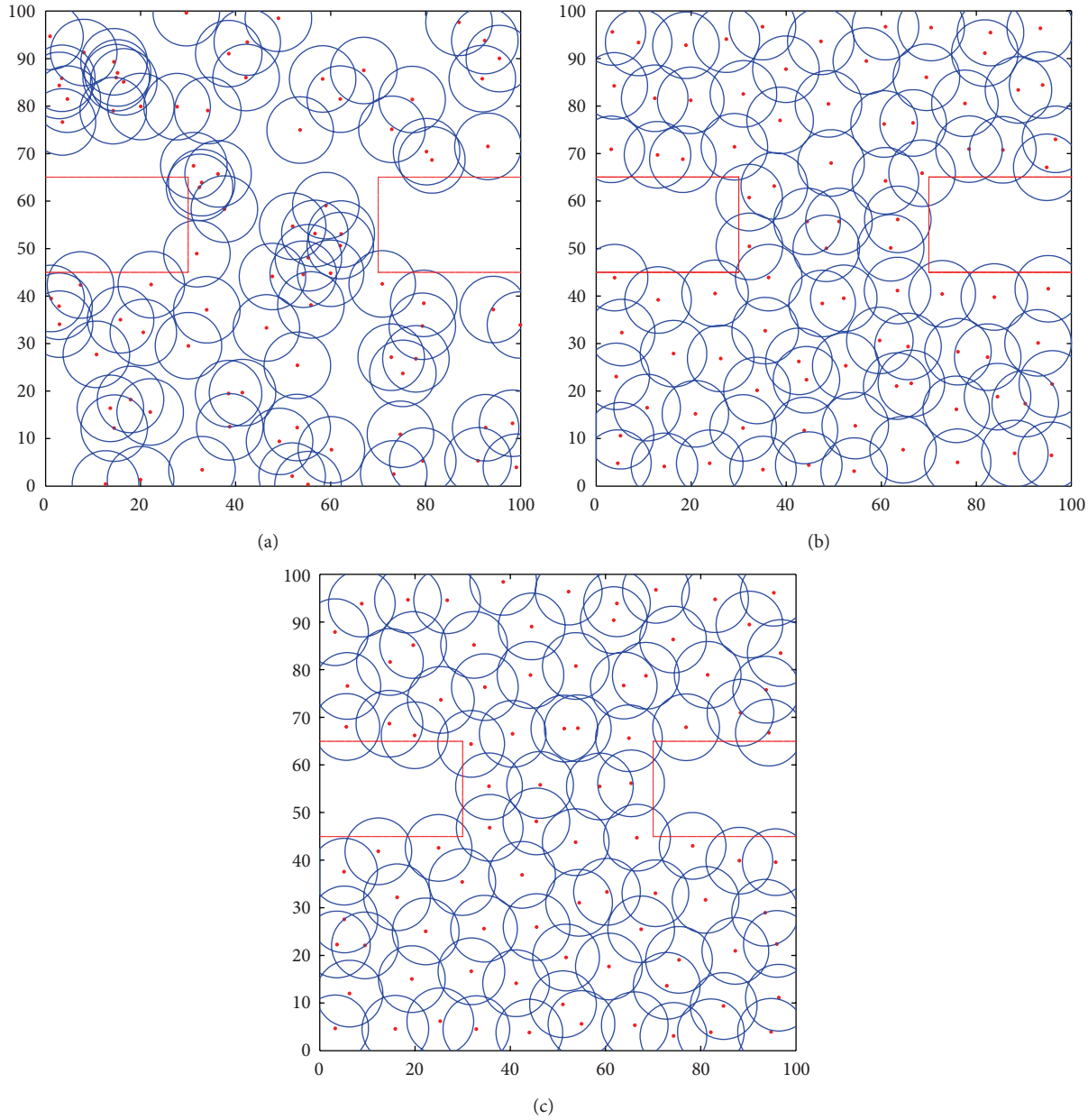



Figure 3: Sensor distribution: (a) initial, (b) original ABC algorithm after 10000 iterations, and (c) FNF-BL-ABC algorithm after 10000 iterations.

ABC algorithm. These again showed that, in the case with obstacles, the proposed approach has higher coverage rate and less energy consumption than original one.

## 5. Conclusion

In this paper, in order to optimize the deployment of wireless sensors, a modified ABC algorithm introducing forgetting and neighbor factor in the onlooker bee phase and backward learning in the scout bee phase is proposed. Simulation results showed that the proposed modified ABC algorithm has higher coverage rate and fast convergence along with less total moving distance for sensor deployment in an area of interest. It has revealed a better performance in solving this kind of problem. In our future work, virtual force-based deployment algorithm will be introduced before the proposed approach is applied to make the initial distribution of sensor less random to accelerate the processing and to make it more practical.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

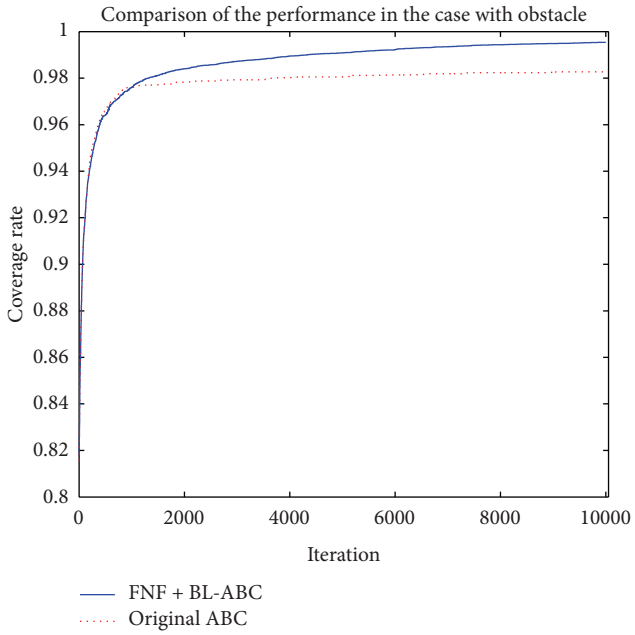



Figure 4: Coverage rate comparison between FNF-BL-ABC and original ABC for the room shown in Figure 3.

## Acknowledgments

This work was supported by the Foundation for Distinguished Young Talents in Higher Education Guangdong, China (Grant no. LYM10011), National Natural Science Foundation of China (Grant no. 61201178), Scientific Project of Guangdong (no. 2010B010600019), and United Foundation Project of Guiyang University Guizhou (Grant no. QianKeHeJ-LKG[2013]36).

## References

[1] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Network Perspective*, IEEE Press, New Jersey, NJ, USA, 2009.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[3] J. Chen, E. Shen, and Y. Sun, "The deployment algorithms in wireless sensor net works: a survey," *Information Technology Journal*, vol. 8, no. 3, pp. 293–301, 2009.

[4] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[5] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Conference on Distributed Autonomous Robotic Systems*, pp. 299–308, Fukuoka, Japan, 2002.

[6] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1293–1303, New York, NY, USA, April 2003.

[7] X. Yu, W. Huang, J. Lan, and X. Qian, "A novel virtual force approach for node deployment in wireless sensor network," in *Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '12)*, pp. 359–363, Hangzhou, China, 2012.

[8] X. Yu, W. Huang, J. Lan, and X. Qian, "A van der Waals force-like node deployment algorithm for wireless sensor network," in *Proceedings of the 8th International Conference on Mobile Ad-Hoc and Sensor Networks*, pp. 218–221, Chengdu, China, 2012.

[9] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[10] D. Jourdan and O. L. de Weck, "Layout optimization for a wireless sensor network using a multi-objective genetic algorithm," in *Proceedings of the 59th IEEE Vehicular Technology Conference (VTC '04)*, vol. 5, pp. 2466–2470, May 2004.

[11] X. Wang, S. Wang, and J. J. Ma, "An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment," *Sensors*, vol. 7, no. 3, pp. 354–370, 2007.

[12] N. Kukunuru, B. Thella, and R. Davuluri, "Sensor deployment using particle swarm optimization," *International Journal of Engineering Science and Technology*, vol. 2, no. 10, pp. 5395–5401, 2010.

[13] Z. Li and L. Lei, "Sensor node deployment in wireless sensor networks based on improved particle swarm optimization," in *Proceedings of IEEE International Conference on Applied Superconductivity and Electromagnetic Devices (ASEMD '09)*, pp. 215–217, Chengdu, China, September 2009.

[14] W. H. Liao, Y. Kao, and Y. S. Li, "A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12180–12188, 2011.

[15] C. Ozturk, D. Karaboga, and B. Gorkemli, "Artificial bee colony algorithm for dynamic deployment of wireless sensor networks," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 20, no. 2, pp. 255–262, 2012.

[16] C. Ozturk, D. Karaboga, and B. Gorkemli, "Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm," *Sensors*, vol. 11, no. 6, pp. 6056–6065, 2011.

[17] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, 2012.

[18] W. Gao and S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters*, vol. 111, no. 17, pp. 871–882, 2011.

[19] W. Gao, S. Liu, and L. Huang, "A global best artificial bee colony algorithm for global optimization," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2741–2753, 2012.

[20] A. Rajasekhar, A. Abraham, and M. Pant, "Levy mutated artificial bee colony algorithm for global optimization," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC '11)*, pp. 655–662, Anchorage, Alaska, USA, October 2011.

[21] H. Wang, "Improve artificial bee colony," *Computer Engineering and Design*, vol. 32, no. 11, pp. 3869–3872, 2011.

[22] X. Bi and Y. Wang, "Artificial bee colony algorithm with fast convergence," *Systems Engineering and Electronics*, vol. 33, no. 12, pp. 2755–2761, 2011.

[23] X. Yu and Z. Zhu, "A modified artificial bee colony algorithm with its applications in signal processing," *International Journal*

*of Computer Applications in Technology*, vol. 47, no. 2-3, pp. 297–303, 2013.

[24] Y. H. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of IEEE World Congress on Computational Intelligence (ICEC '98)*, Evolutionary Computation Proceedings, pp. 69–73, Anchorage, Alaska, USA, May 1998.

[25] R. V. Kulkarni and G. K. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: a brief survey," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 41, no. 2, pp. 262–267, 2011.

[26] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition versus randomness in soft computing techniques," *Applied Soft Computing*, vol. 8, no. 2, pp. 906–918, 2008.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 482727, 11 pages
http://dx.doi.org/10.1155/2013/482727

*Research Article*

# The Influence of Communication Range on Connectivity for Resilient Wireless Sensor Networks Using a Probabilistic Approach

**Yuanjiang Huang, José-Fernán Martínez, Juana Sendra, and Lourdes López**

*Centro de Investigación en Tecnologías Software y Sistemas Multimedia para la Sostenibilidad (CITSEM), Campus Sur Universidad Politécnica de Madrid (UPM), 28031 Madrid, Spain*

Correspondence should be addressed to Yuanjiang Huang; yuanjiang@diatel.upm.es

Received 12 June 2013; Revised 6 August 2013; Accepted 8 August 2013

Academic Editor: Shengming Jiang



Wireless sensor networks (WSNs) consist of thousands of nodes that need to communicate with each other. However, it is possible that some nodes are isolated from other nodes due to limited communication range. This paper focuses on the influence of communication range on the probability that all nodes are connected under two conditions, respectively: (1) all nodes have the same communication range, and (2) communication range of each node is a random variable. In the former case, this work proves that, for $0 < \varepsilon < e^{-1}$, if the probability of the network being connected is $0.36\varepsilon$, by means of increasing communication range by constant $C(\varepsilon)$, the probability of network being connected is at least $1 - \varepsilon$. Explicit function $C(\varepsilon)$ is given. It turns out that, once the network is connected, it also makes the WSNs resilient against nodes failure. In the latter case, this paper proposes that the network connection probability is modeled as Cox process. The change of network connection probability with respect to distribution parameters and resilience performance is presented. Finally, a method to decide the distribution parameters of node communication range in order to satisfy a given network connection probability is developed.

## 1. Introduction

Wireless sensor networks (WSNs) [1, 2] are a promising technology nowadays. The use of WSNs in numerous applications, such as forest monitoring, disaster management, space exploration, factory automation, secure installation, border protection, and battlefield surveillance, is emerging. WSNs technology is the basis of future network "Internet of Things" (IoT) [3], which offers a vision where anyone can interact with any addressable nodes (things or objects)—such as RFID tags, sensors, and mobile phones—anywhere and anytime. "Anywhere" suggests that any object is reachable from any location. From the network topology point of view, every node in WSNs should be able to, directly or through limited number of intermediate nodes, connect to any other nodes. This kind of network is called "connected network." If the network is still connected after removing at most $k-1$ nodes, it is called $k$-connected network, where $k = 1, 2, 3, \ldots$. A $k$-connected network guarantees that at least $k$ different paths are available for transmitting signals from one node to any other nodes.

However, $k$-connected network is not always possible. In WSNs, sensor nodes are usually deployed in the areas of interest either randomly or according to a predefined distribution. In this case, it is likely that some nodes are isolated from other nodes. Therefore, the network connection is characterized by probability. On the other hand, the resilient problem, which indicates fault-tolerance capability in the presence of node failure, is also important in the probabilistic network. Our concern in this paper is the probability that the WSNs are a connected network and network resilience against the node failures.

Most of earlier studies focus on the model where each node in a network is the same and, for example, has the same communication range. However, WSNs nodes are usually heterogeneous. The communication range of the WSNs node

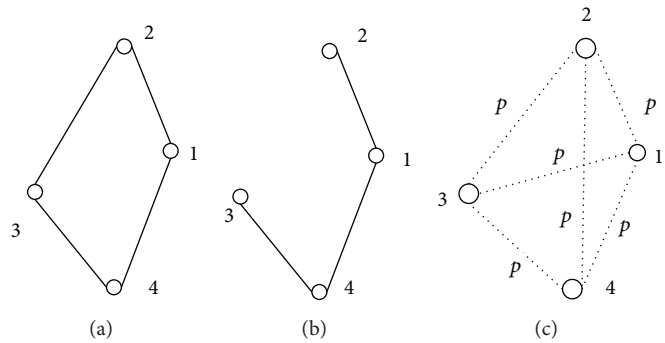



Figure 1: Different methods to connect nodes: (a) node connects 2 nearest neighbors; (b) node connects other nodes within its communication range; (c) node connects other nodes with same probability $P$.

may vary from one node to another, and even communication range of the same node may change over time. For instance, in a wireless network, the transmission power required for a node to reach another node is proportional to $R^{\alpha}$, where $R$ is the transmission radius and $\alpha$ is the loss constant depending on the wireless medium of which typical value is between 2 and 4 and may vary from devices to devices [4]. According to various wireless communication technologies, communication range may vary from tens to thousands meters, such as IEEE 802.11 (25–600 m), Bluetooth (10–100 m), ZigBee (10–75 m), HomeRF (50 m), UWB (10 m), and WiMAX (1–50 km). Depending on how long nodes work, residual energy of battery powered devices decreases over time, so a node may try to shorten communication range in order to save energy. Environments where nodes are deployed, for example, indoor or outdoor, with or without obstacle, result in communication range quite different due to the interference, shadowing, fading, and pass loss [5].

This work concentrates on WSNs connection probability for both heterogenous and homogenous networks in terms of communication range. Assuming WSNs nodes are randomly and uniformly distributed, two problems are addressed in this paper: given a network where all nodes have the same communication range, how does the connection probability change as communication range increases? In the case that communication range is a random variable, what is the network connection probability?

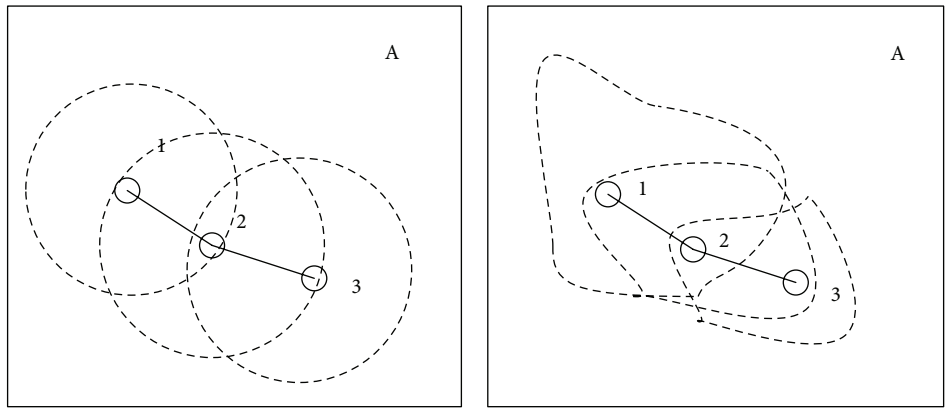
Through analysis, this work finds that for $0 < \varepsilon < e^{-1}$ and the number of nodes in the network is big enough and if the original network connection probability is $0.36\varepsilon$, through increasing the communication range by constant $C(\varepsilon)$, the probability of a network being connected increases from $0.36\varepsilon$ to $1 - \varepsilon$. Explicit function $C(\varepsilon)$ is given in this paper. It turns out that, when a network is connected, it is also almost sure $\log(n) + b$-connected (where $n$ is the total number of nodes deployed and $b$ is a constant greater than 1), which is important for the WSNs resilient against the node failure. Afterwards, the connection probability problem with random communication range, which is often the real case in the WSNs, is studied. The model is reformulated as Cox process, and the connection probability is analyzed by simulation. A method for determining the distribution function parameters for a given connection probability is developed.

Our main contributions are as follows: first, this paper employs an effective and novel approach to obtain analytical results for homogenous WSNs connectivity, some of which have been validated by previous studies; second, we propose that the Cox process can be used to model heterogenous WSNs and the simulations are performed to reveal the relations between the network connection probability and its distribution parameters.

The rest of this paper is organized as follows. Section 2 introduces the basic concepts of network model and the problem to be addressed. In Section 3, derivation and verification in case that the network nodes have the same communication range are presented. In Section 4, communication range is modeled as a random variable. A brief introduction of related works is provided in Section 5 while Section 6 concludes our work.

## 2. Network Model and Problem Statement

Usually, there are three methods to create links between nodes, as presented in Figure 1. One is *k-nearest neighbor model*. In this model, the network is formed by each node connecting to $k$-nearest neighbors; for example, in Figure 1(a), each node has 2 neighbors. The second is *disc model*. Node is modeled as a disk with communication radius $r$. The node $s$ is linked to node $u$ if the Euclidean distance between $s$ and $u$ is less than $r$; for example, in Figure 1(b), node 3 cannot connect to node 2 and node 1 because they are out of communication range of node 3. The last one is *Erdös-Rényi random graph* that connects any two nodes by the same probability which is inappropriate in the WSNs; for example, in Figure 1(c), each node connects other nodes with the same probability $P$. The *k-nearest neighbor model* can be achieved by changing communication range of each node until the number of neighbors reaches $k$. *Disc model*, on the other hand, connects those nodes that fall into its communication range. *k-nearest neighbor model* and *disc model* are different. *k-nearest neighbor model* makes sure that there is no isolated node, but *disc model* is characterized by the probability that a network does not have isolated nodes. *Disc model* is more plausible in the WSNs in the case that obtaining $k$ neighbors is not always feasible. For instance, in wireless environment, some nodes may be unable to connect




(a) (b)

Figure 2: (a) Disk communication model and (b) irregular communication model.

to a required number of neighbors due to the communication range limitation.

The notations and basic network definitions that will be used throughout the paper are now introduced. Additional terminologies are referred to [6]:

$n$: total number of nodes deployed in target field, and $n \gg 1$,

$A$: area of node deployed,

$\rho$: node density, defined as $n/A$,

$t$: expected number of neighbors of node.

Note that in this paper "log" means the logarithm to nature base $e$. Next, main definitions are introduced.

*Definition 1.* Node's communication range is defined as the area where other nodes can receive its signal.

For a disk, the communication range is the circle with radius $r$. However, communication range is not necessary modeled as a disk. The communication range of radio is highly probabilistic and irregular [7, 8]. Figures 2(a) and 2(b) illustrate the ideal disk communication and irregular communication model, respectively. More importantly, the communication range of each node may not be the same. Note that the analysis in this section is a disk, but it can also apply to the irregular communication model.

*Definition 2.* $S_{n,r}$ denotes a network following *disc model*. More specifically, the network is formed by $n$ nodes randomly and uniformly deployed in area $A$. The node is modeled as a disk with radius $r$.

This paper focuses on the probability of network $S_{n,r}$ being $k$-connected. A $k$-connected network implies that there are still $k-1$ alternative path(s) if one path failed, therefore a higher $k$ indicates that the network is more resilient against failures. In this paper, $k$ is used to evaluate the WSNs resilience. This property depends on many factors, such as communication range, node density $\rho$, node processing capability, node energy, and deployment environment. This paper is only interested in the impact of communication range on the connection probability. The problem can be stated as follows.

"Given WSNs $S_{n,r}$ with fixed node density $\rho$, in the cases in which node communication range is the same and different, how network connection probability and resilience performance change as node communication ranges vary?"

## 3. Homogenous Node Deployment in WSNs

This section considers that, in the network $S_{n,r}$, each node has the same communication range. First, the mathematical model that will be used is presented. Based on this model, theoretical results are proved and validated by an example and simulations. In Section 4, the situation where communication range of each node is a random variable will be discussed.

*3.1. Network Connection Probability Analysis.* For uniformly distributed nodes with density $\rho$, the number of nodes in the area $\pi r^2$ has a *Poisson* distribution [9]; therefore the probability of a node having $N$ neighbor nodes is

$$P(N) = \frac{\left(\rho\pi r^2\right)^N}{N!} e^{-\rho\pi r^2}. \tag{1}$$

Number of node neighbor is also called the node's degree. The minimal degree of all nodes is called the network degree. If the network has $n$ nodes, the probability of network $S_{n,r}$ is $k$-connected given by following well-known formula [9]:

$$P(k) = \left(1 - \sum_{N=0}^{k-1} \frac{\left(\rho\pi r^2\right)^N}{N!} e^{-\rho\pi r^2}\right)^n. \tag{2}$$

Let

$$t = \pi r^2. \tag{3}$$

Note that $t$ indicates the communication range of a node, but, if $\rho = 1$, $t$ actually is the expected number of neighbors a node has.

Without loss of generality, assume that $\rho = 1$. For a real network the density of node $\rho = 1$ indicates that the average number of nodes in unit area is one. However, whether $\rho$ is equal to 1 is irrelevant in this model, because if $\rho$ is not 1, say $\rho'$, then letting $r' = r/\sqrt{\rho'}$ the results will be the same. In order to simplify denotation, define

$$W(t,k) = \sum_{N=0}^{k-1} \frac{(t)^N}{N!} e^{-t}. \tag{4}$$

So, (2) can be rewritten as

$$P(t,k) = (1 - W(t,k))^n, \tag{5}$$

and its first derivation with respect to $t$ is

$$\begin{aligned} \frac{dP(t,k)}{dt} &= -n(1 - W(t,k))^{n-1} \frac{dW(t,k)}{dt} \\ &= n\left( \frac{e^t - \sum_{N=0}^{k-1} \left(t^N/N!\right)}{e^t} \right)^{n-1} \left( e^{-t} \frac{t^{k-1}}{(k-1)!} \right). \end{aligned} \tag{6}$$

The function $P(t,k)$ can be written as $P(t) = (1 - e^{-t})^n$ when $k = 1$. In this section, the properties of $P(t)$ are analyzed, namely, 1-connected network. Two points are found out where $P(t)$ almost starts and stops growing in order to show that the connection probability increases from near 0 to reach 1.

**Proposition 3.** *Letting $P(t) = (1 - e^{-t})^n$ and $n \gg 1$, then following statements hold:*

(1) *for every $0 < \varepsilon < (1 - (1/n))^{(n-1)}$, there exists $0 < t_1 < t_2$ such that $P'(t_1) = P'(t_2) = \varepsilon$;*

(2) *$P(t)$ has a flex point at $(\log(n), (1 - (1/n))^n)$.*

*Proof.* (1) First, (5) is a monotonically increasing function for any $k \geq 1$ (note that $e^t = \sum_{N=0}^{+\infty} (t^N/N!)$ is the Taylor expansion of $e^t$, and $t \geq 0$); see Figure 3(a).

Now consider the first and second derivative functions of $P(t)$:

$$\begin{aligned} P'(t) &= n\left(1 - e^{-t}\right)^{n-1} e^{-t}, \\ P''(t) &= ne^{-t}\left(1 - e^{-t}\right)^{n-2} \left(ne^{-t} - 1\right). \end{aligned} \tag{7}$$

It is evident that $P''(t)$ vanishes at $t = 0$ and $t = \log(n)$. Furthermore, $P''(t) > 0$ for $t \in (0, \log(n))$ and $P''(t) < 0$ for $t \in (\log(n), \infty)$. Therefore, $P'(t)$ is an increasing function in the interval $(0, \log(n))$ and a decreasing function in $(\log(n), \infty)$. Hence, $P'(t)$ reaches the maximum value at $t = \log(n)$ (see Figure 3(a)). On the other hand, since $\lim_{t \to \infty} P'(t) = 0$, by applying *Bolzano* Theorem, for every $0 < \varepsilon < P'(\log(n)) = (1 - (1/n))^{(n-1)}$, there exists $0 < t_1 < t_2$, such that $P'(t_1) = P'(t_2) = \varepsilon$ (Figure 3(b)).

(2) It is derived from the proof of statement above. □

*Remark 4.* Note that $0 < \varepsilon < (1 - (1/n))^{(n-1)} < 1$ since $n \gg 1$. In Figure 3(b), $\varepsilon = (1 - (1/500))^{(500-1)} \approx 0.368$.

The proof of the previous proposition can be applied to obtain the following result.

**Theorem 5.** *Let $P(t) = (1 - e^{-t})^n$, $n \gg 1$, $b > 1$, and $\varepsilon = e^{-b}$. Then there exists a constant number of neighbors, $C(\varepsilon) = \log((1 - \log\varepsilon)/\varepsilon)$, for which the network becomes connected with probability increasing from $P(t_1 = \log(n/(b+1)))$ to $P(t_2 = \log(n) + b)$.*

*Proof.* First, it can be observed that $\varepsilon = e^{-b}$ satisfies the hypothesis in Proposition 3(1) since $\varepsilon = e^{-b} < e^{-1}$; therefore $0 < \varepsilon < (1 - (1/n))^{(n-1)}$. Let $x = e^{-t}$ and define $f(x)$ as

$$f(x) = n(1-x)^{n-1}x - \varepsilon. \tag{8}$$

Then, the goal is to find the roots of (8). For this purpose, consider the derivative function:

$$\begin{aligned} f'(x) &= n(1-x)^{n-1} - n(n-1)(1-x)^{n-2}x \\ &= n(1-x)^{n-2}(1 - nx). \end{aligned} \tag{9}$$

Note that $f(x)$ is the function $P'(t) - \varepsilon$ under the change of variable $x = e^{-t}$, and by applying the proof of Proposition 3 there exist only two roots $x_1$ and $x_2$ of $f(x)$ in the interval $(0, 1)$. *Newton* method can be used to find out the approximation of roots $x_1$ and $x_2$. However, its accuracy depends on the initial value, which should be close enough to the real root. Letting $x_0$ be the initial value, according to (8) and (9), yields

$$\begin{aligned} x &\approx x_0 - \frac{f(x_0)}{f'(x_0)} \\ &= x_0 - \frac{n(1-x_0)^{n-1}x_0 - \varepsilon}{n(1-x_0)^{n-2}(1 - nx_0)} \\ &= x_0 - \frac{n(1-x_0)x_0 - \left(\varepsilon/(1-x_0)^{n-2}\right)}{n(1 - nx_0)} \\ &= x_0 - \frac{(1-x_0)x_0}{1 - nx_0} + \frac{\varepsilon/(1-x_0)^{n-2}}{n(1 - nx_0)}. \end{aligned} \tag{10}$$

Additionally, the inequality $0 < x_2 < (1/n) < x_1 < 1$ holds from the proof of Proposition 3; then *Newton* method can be applied. Let $x_0 = 0$ as the initial value to approximate $x_2$ and $x_0 = (b/n)$ (where $b > 1$) as the initial value to find $x_1$:

$$\begin{aligned} x_2 &= \frac{\varepsilon}{n} = \frac{e^{-b}}{n}, \\ x_1 &= \frac{b}{n} - \frac{(1 - (b/n))(b/n)}{1-b} + \frac{\varepsilon/(1-(b/n))^{n-2}}{n(1-b)} \\ &= \frac{1}{n}\left( \frac{\left(b^2/n\right) - b^2 + \left(\varepsilon/(1-(b/n))^{n-2}\right)}{1-b} \right). \end{aligned} \tag{11}$$

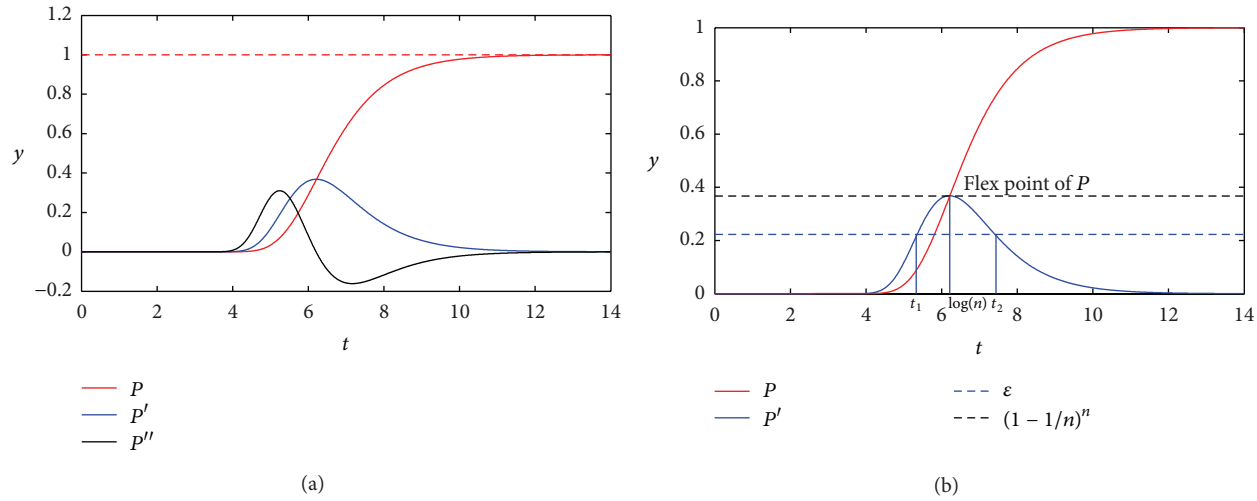



FIGURE 3: Network connection probability function $y = P(t)$ and corresponding $P'(t)$ and $P''(t)$ (a); values $t_1$ and $t_2$ for a given $\varepsilon$ and $n = 500$ (b).

Taking into account that we have $(1 - (b/n))^{n-2} \to e^{-b}$ when $n \to +\infty$ and $\varepsilon = e^{-b}$, therefore

$$\frac{\left(b^2/n\right) - b^2 + \left(\varepsilon/(1-(b/n))^{n-2}\right)}{1-b} \longrightarrow 1 + b. \tag{12}$$

Note that $b = -\log(\varepsilon)$. Letting $t_2 = -\log x_2$, $t_1 = -\log x_1$, and defining $C(\varepsilon) = t_2 - t_1$, according to (12), we obtain

$$\begin{aligned} &C(\varepsilon) \\ &= \log\left(\frac{x_1}{x_2}\right) \\ &= \log\left(\frac{(1/n)\left(\left(\left(b^2/n\right) - b^2 + \left(\varepsilon/(1-(b/n))^{n-2}\right)\right)/(1-b)\right)}{\varepsilon/n}\right) \\ &\longrightarrow \log\left(\frac{1-\log(\varepsilon)}{\varepsilon}\right). \end{aligned} \tag{13}$$

Finally, taking into account that $t_2 = \log(n) + b$, we have $t_1 = t_2 - C(\varepsilon) = \log(n/(b+1))$. Since $P(t)$ is an increasing function, we conclude that the network becomes connected with probability increasing from $P(t_1 = \log(n/(b+1)))$ to $P(t_2 = \log(n) + b)$. □

**Theorem 6.** *Letting $\varepsilon = e^{-b}$ and $b > 1$, if $\pi r^2 = \log n + b$, then network connection probability $P(S_{n,r}$ connected$)$ is at least $1 - \varepsilon$ when $n \to +\infty$.*

*Proof.* Consider

$$P\left(S_{n,r} \text{ connected}\right) = \left(1 - \frac{e^{-b}}{n}\right)^n \geq 1 - n\frac{e^{-b}}{n} = 1 - \varepsilon. \tag{14}$$

□

*Remark 7.* This theorem shows that, as $b \to +\infty$, network connection probability tends to 1 and leads to the network that has degree $\log n + b$. The author in [10] proves that if a network does not have any links at the beginning, and later links are added to connect nodes, the resulting network becomes $k$-connected as soon as network degree is $k$. Therefore, this theorem shows that once network becomes connected, it turns out to be $\log n + b$-connected with high probability. This conclusion is consistent with the result in [11]: by increasing $k$ network becomes $s$-connected very shortly after it becomes connected, for $s = O(\log n)$. $\log n + b$-connected network makes WSNs more resilient against node failure because there are $\log n + b$ distinct paths from one node to any other nodes.

**Theorem 8.** *Letting $\varepsilon = e^{-b}$ and $b > 1$, if $\pi r^2 = \log(n/(b+1))$, then the network connection probability $P(S_{n,r}$ connected$)$ is about $0.36\varepsilon$ when $n \to +\infty$.*

*Proof.* Consider

$$\begin{aligned} P\left(S_{n,r} \text{ connected}\right) &= \left(1 - \frac{b+1}{n}\right)^n \\ &\longrightarrow e^{-b-1} = 0.36e^{-b} = 0.36\varepsilon. \end{aligned} \tag{15}$$

□

**Theorem 9.** *Letting $\varepsilon = e^{-b}$ and $b > 1$, if $\pi r^2 = \log n + b$, then the network connection probability is $e^{-e^{-b}}$, when $n \to +\infty$.*

*Proof.* Consider

$$P\left(S_{n,r} \text{ connected}\right) = \left(1 - \frac{e^{-b}}{n}\right)^n \longrightarrow e^{-e^{-b}}. \tag{16}$$

□

*Remark 10.* This conclusion is the same as [12] and has similar form in the Erdös-Rényi random graph [13].

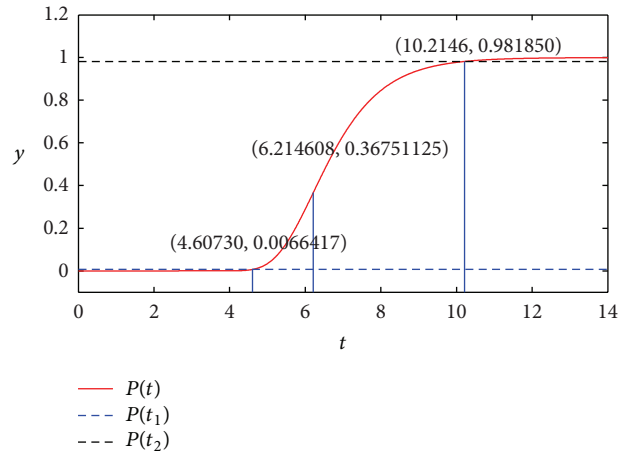



Figure 4: Network connection probability $y = P(t)$ increases from $P(t_1)$ to $P(t_2)$ for $n = 500$.

**Corollary 11.** *Letting* $0 < \varepsilon < e^{-1}$, *if the probability of network being connected is* $0.36\varepsilon$ *when communication range is* $\pi r^2$, *then, by increasing node communication range by constant* $C(\varepsilon) = \log((1 - \log\varepsilon)/\varepsilon)$, *namely,* $\pi r^2 + C(\varepsilon)$, *the probability of network being connected is at least* $1 - \varepsilon$.

*Proof.* It is obvious from the previous Theorems 5, 6, and 8. □

This section addresses one question. If a node current communication range is known, then the connection probability can be calculated by using (2). If the network connection probability is very low, maybe one wants to increase the node communication range to obtain a higher network connection probability. Equation (2) can be used again to calculate the required communication range, but surprisingly the corollary proved in this section shows that the incremental of communication range to obtain a high connection probability is a constant for any size of network.

*3.2. Validation Results.* This section validates the previous results by an example and simulations. In the example, 500 nodes with equal communication range are deployed in the field with $\sqrt{500} \cdot \sqrt{500}\, m^2$.

*Example 12.* The function $P(t) = (1-e^{-t})^n$ for $n = 500$ and $b = 4$ is studied. According to Theorem 5, there exists a constant number of neighbors $C(\varepsilon) = \log((4 + 1)/e^{-4}) = 5.60944$, for which the network becomes connected with probability increasing from $P(t_1) = 0.65705\%$ to $P(t_2) = 98.18507\%$ (as depicted in Figure 4).

First, it is observed that

$$\varepsilon = e^{-4} = 0.0183 < \left(1 - \frac{1}{500}\right)^{(500-1)} = 0.3682 \qquad (17)$$

which satisfies the hypothesis in Proposition 3(1). In our approach, the Newton's method is used to approximate the roots of

$$f(x) = 500(1 - x)^{500-1}x - 0.0183, \qquad (18)$$

obtaining

$$x_1 = 9.98 \cdot 10^{-3}, \qquad x_2 = 0.04 \cdot 10^{-3}. \qquad (19)$$

Observe that $0 < x_2 < 1/500 = 0.002 < x_1$. Hence,

$$t_2 = -\log(x_2) = 10.21461, \qquad t_1 = -\log(x_1) = 4.60517, \qquad (20)$$

which indicates $P(t_1 = 4.60517) = 0.65705\%$, $P(t_2 = 10.21461) = 98.18507\%$, and $C(\varepsilon) = t_2 - t_1 = 5.60944$.

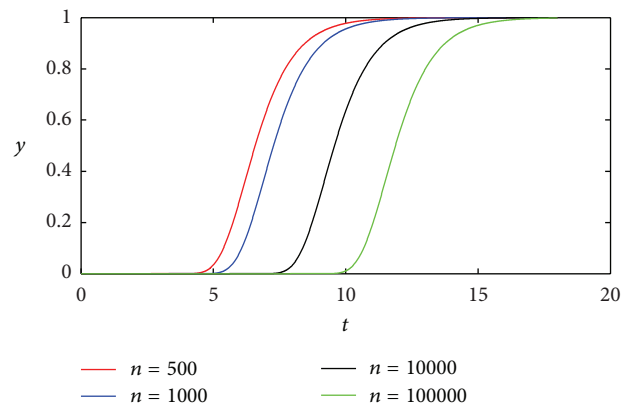



Figure 5: Network connection probability $y = P(t)$ when $n = 500, 1000, 10000$, and $100000$.

Table 1: Connection probability with different sizes of network.

| $n$ | $t_1$ | $t_2$ | $t_2 - t_1$ | $P(t_1)$ | $P(t_2)$ |
|---|---|---|---|---|---|
| 500 | 4.60517 | 10.21461 | 5.60944 | 0.65705% | 98.18507% |
| 1000 | 5.29832 | 10.90776 | 5.60944 | 0.66540% | 98.18509% |
| 10000 | 7.60090 | 13.21034 | 5.60944 | 0.67295% | 98.18511% |
| 100000 | 9.90349 | 15.51293 | 5.60944 | 0.67371% | 98.18511% |

Figure 5 shows connection probability $y = P(t)$ when $n = 500, 1000, 10000, 100000$. Table 1 demonstrates the values of $x_1$, $x_2$, $t_1$, $t_2$, and $C(\varepsilon)$ and corresponding values of $P(t_1)$ and $P(t_2)$, for $n = 500, 1000, 10000, 100000$. For any $n$ in the table, the obtained value $t_2 - t_1 \approx C(\varepsilon) = 5.60944$. Of course, in a real network, the number of neighbors is integer, so 6 neighbors are needed. This example implies that, regardless of network size (number of nodes should be big enough), if the network connection probability is 0.66%, by increasing the communication range until each node obtains 6 more neighbors (namely, increasing communication range by 6 $m^2$), the network connection probability reaches at least 98.17%. Meanwhile, the network will be at least 10-connected.

In order to validate Theorems 6 and 8, this paper calculates the error between theoretical results and approximation values with different , $n$ and $b$, as shown in Figure 6. The error of Theorem 6 is defined as $(1 - (e^{-b}/n))^n - (1 - e^{-b})$, and the error of Theorem 8 is defined as $(1 - ((b + 1)/n))^n - 0.36e^{-b}$. The errors for both theorems are very small, which indicate that both have a good approximation.

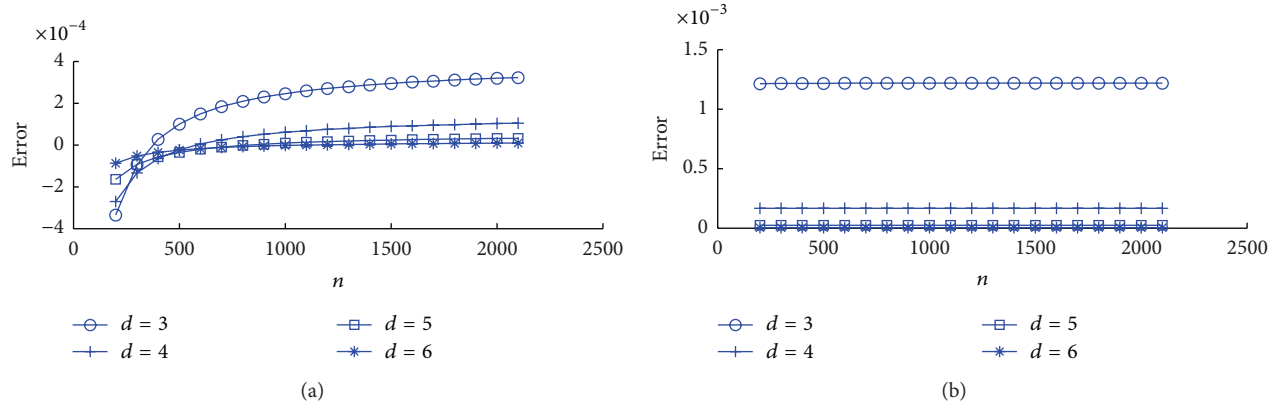



FIGURE 6: (a) The errors of Theorem 6; (b) the errors of Theorem 8.

## 4. Heterogenous Node Deployment in WSNs

In the last section, the obtained asymptotic results were based on the assumption that each node has the same communication range which is often not the case in practice. This section presents the connection probability when node communication range follows a normal distribution, that is, $t \sim N(\mu, \sigma^2)$.

Formally, network model is reformulated as follows: $n$ nodes are randomly and uniformly deployed in area $A$ with density $\rho = 1$. Communication range of node $i$, denoted as $t_i$, is i.i.d random variable and has normal distribution $t_i \sim N(\mu, \sigma^2)$. Hence, the number of neighbors of the node, denoted as $k_t$, is the Poisson random variable condition on parameter $t$, where $t \sim N(\mu, \sigma^2)$. This model is analog to the so-called Cox process in which random variable is Poisson process where density itself is a stochastic process. Cox process is widely used in economics, for example, [14].

*4.1. Connection Probability for Random Communication Range.* $E[V]$ denotes the expected value of a random variable $V$; therefore the expected neighbors of node are

$$E[k_t] = E[E[k_t \mid t]] = E[t] = \mu. \tag{21}$$

In what follows, connection probability itself is researched. For $k \geq 1$, probability of node $i$ having at least $k$ neighbors is given by

$$P_{i,k} = g_i(t_i) = 1 - \sum_{N=0}^{k-1} \frac{t_i^N}{N!} e^{-t_i}. \tag{22}$$

For $k = 1$, $P_{i,1} = 1 - (1/e^{t_i})$ is the probability that node $i$ is not isolated. $e^{t_i}$ has log normal distribution. Therefore, the expected value $E[P_{i,1}]$ and variance $\mathrm{Var}[P_{i,1}]$ can be obtained via standard method:

$$\begin{aligned} E[P_{i,1}] &= 1 - e^{-\mu + (1/2)\sigma^2}, \\ \mathrm{Var}[P_{i,1}] &= \left(e^{\sigma^2} - 1\right)\left(E[P_{i,1}]\right)^2. \end{aligned} \tag{23}$$

For $k > 1$ neighbors, the distribution of $P_{i,k}$ does not have a closed-form expression.

If $n$ is big enough, the probability of network being $k$-connected is

$$P_k = \prod_{i=1}^{n} P_{i,k}. \tag{24}$$

Since parameter $t$ is a random variable, $P_k$ is a random variable as well. Letting $P_{\min} = \min\{P_{1,k}, P_{2,k}, \ldots, P_{n,k}\}$, because $0 \leq P_{i,k} \leq 1$, so

$$P_{\min}^n \leq P_k \leq P_{\min}. \tag{25}$$

Therefore the obstruction of connection probability of entire network is the node which has the minimal communication range.

$P_k$ is affected by several parameters: $k$, $n$, $\mu$, and $\sigma$. Theorem 6 is used to decide $\mu$. According to Theorem 6, the probability of the network being connected is at least 99.33% when $b = 5$. Let $\rho = 1$ and take $\log n + 5$ as average $\mu$ of communication range; for instance, if $n = 500$, then $\mu = 11.2$. In other words, 500 nodes with node communication range following normal distribution $t \sim N(11.2, \sigma^2)$ are deployed.

Our major concerns are the parameter $\sigma$ which indicats communication range difference and $k$ which shows the resilience capability. In order to study the changes of connection probability $P_k$ as parameters vary, the following simulations are performed: (1) cumulative distribution function (CDF) of $P_k$ is calculated after 500 runs with various $\sigma$ and $k$, as shown in Figures 7 and 8; (2) given $\mu$ and $\sigma$, what is the probability of network being $k$-connected as the number of nodes deployed grows? This is done by computing average of $P_{n,k}$ after 500 runs for a given number of nodes, as illustrated in Figures 9 and 10; (3) how to choose the parameters in order to get the required connection probability. This is discussed in Section 4.3.

Figures 7 and 8 show the CDF of $P_{n,k}$ when $\sigma$ and $k$ change. The network probability is sensitive to standard deviation. As mentioned earlier, a single node that has small communication range can cause the whole network connection probability to be low. For instance, in Figure 8 when $\sigma = 3$ and $k = 2$, the probability of network being connected is almost sure less than 40%.

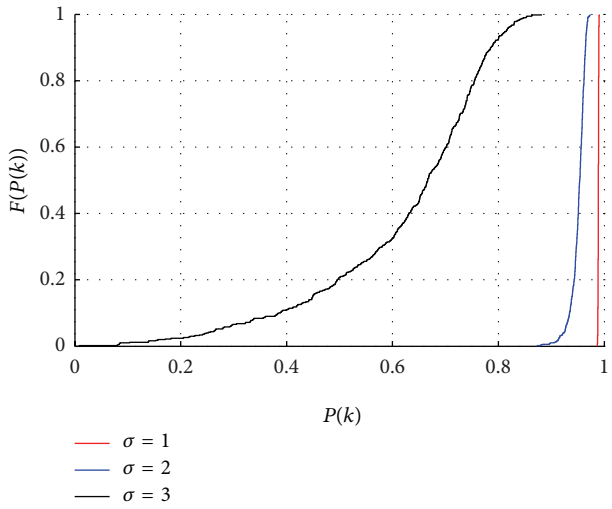



Figure 7: Cumulative distribution function (CDF) of $P_k$ when $\sigma = 1, 2$, and 3, $n = 500$, and $k = 1$.

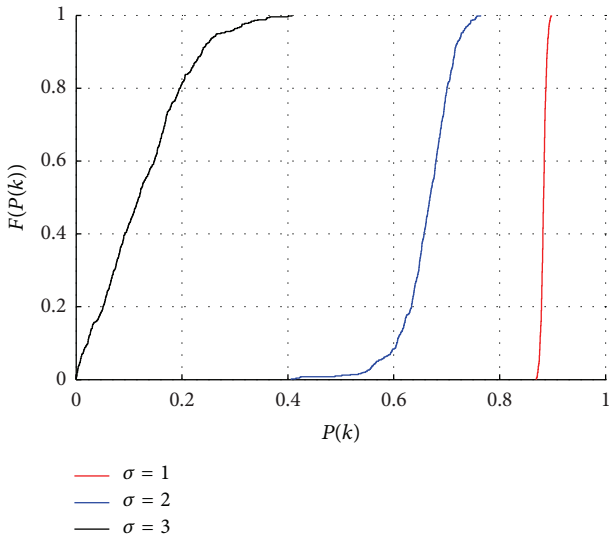



Figure 8: Cumulative distribution function (CDF) of $P_k$ when $\sigma = 1, 2$, and 3, $n = 500$, and $k = 2$.

Figure 9 illustrates the connection probability as $N$ nodes were deployed in network when $\sigma = 2$ and $k = 1, 2, 3, 4$. Figure 10 shows the changes when $\sigma = 1, 2, 3$ and $k = 1$. Both figures show that the average of $P_k$ is the decreasing function of $k$, $N$, and $\sigma$. Network connection probability as network size growing is predictable. For instance, Figure 10 shows that the network average connection probability for $\sigma = 3$ is about 73% when the network has 250 nodes, but the probability falls to 55% if the network size is doubled. Figure 9 shows how the resilience performance decreases when network size grows or the probability decreases if higher resilience performance is required. For example, for networks which have 200 nodes, the probability that this network can tolerate 1, 2, and 3 (i.e., $k = 2, 3, 4$) nodes failure are about 83%, 50%, and 10%, respectively.

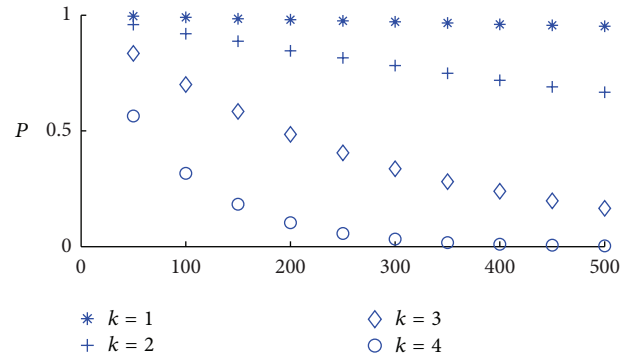



Figure 9: $k$-connected probability $P$ when $\sigma = 2$ and $k = 1, 2, 3$, and 4.

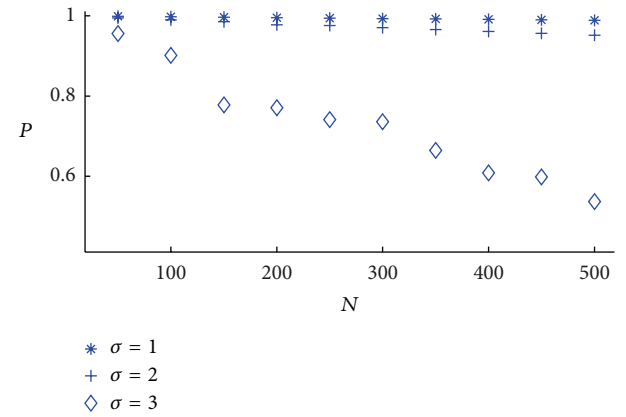



Figure 10: Connection probability $P$ when $\sigma = 1, 2$, and 3 and $k = 1$.

*4.2. Choose Distribution Parameter.* The simulations in Section 4.2 show $P_k$ with different parameters. In this section, it is addressed which distribution parameter(s) can maintain the given $P_k$. This is helpful to choose appropriate parameters when network simulator is used to simulate real networks. According to (6), (22) is a monotonically increasing function of $t_i \in [0, +\infty)$, and its inverse function is written as

$$t_{i,k} = g_i^{-1}(p_{i,k}). \tag{26}$$

Letting $p_{i,k}^{(0)}$ be an instance of $p_{i,k}$, thus $t_{i,k}^{(0)} = g_i^{-1}(p_{i,k}^{(0)})$. The probability $p_{i,k}$ being greater than $p_{i,k}^{(0)}$ is given by

$$\int_{t_{i,k}^{(0)}}^{+\infty} f_T(t)\, dt, \tag{27}$$

where $f_T(t)$ is the probability density function of $t$. If the probability of a network required to keep network $k$-connected is at least $P_0$, the corresponding probability for each node is at least

$$p_{\min} \geq P_0^{1/n}. \tag{28}$$

With formula (26)–(28), the required density function parameter of communication range for given $P_0$ can be calculated.

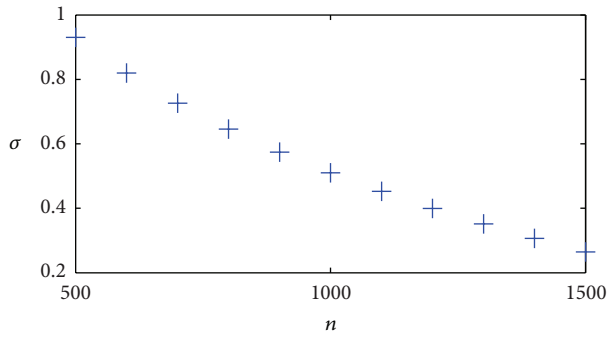



Figure 11: Required $\sigma$ for 90% connection probability when $\mu = 10$.

For example, 500 nodes are deployed in $\sqrt{500} \cdot \sqrt{500}\,\mathrm{m}^2$; communication area is $t \sim N(\mu, \sigma^2)$ with mean $\mu = 10$. If the desired probability of the whole network being connected, that is, $k$-connected, is at least 90%. Standard deviation of this distribution is evaluated. For $k = 1$, according to (26) and (28), corresponding minimal range is $t_{i,1}^{(0)} = 8.46$. In order to make probability of $t_{i,1}$ greater than $t_{i,1}^{(0)}$ is high, for example, at least 95%, according to (27), $t_{i,1} = \mu - 1.65\sigma$. Therefore corresponding standard deviation $\sigma$ should be no more than 0.93. This is useful in the case of using network simulator to choose appropriate parameters to design high probability connected networks. Figure 11 shows the required $\sigma$ in order to make the network connection probability at least 90% when the number of nodes are different. Note that the node density is always 1.

## 5. Related Works

Extensive studies have been done on the connection problem of networks. Many of them focus on how many neighbors or network density is needed so that a network connects with high probability, such as [15]; some construct network to satisfy connectivity [16, 17]; some works try to develop algorithms to preserve network connectivity or coverage, for example, [18–20], while some other works study other aspects of network connectivity, such as [21] which evaluates the quality of connectivity by measuring the reliability of link; it shows that the largest eigenvalue of the probabilistic connectivity matrix can serve as a good measure of the quality of network connectivity. When all the nodes of a region fail, [22] measures the number of connected components. This paper studies the connection probability when the network nodes are randomly deployed.

When nodes are randomly deployed, asymptotic upper and lower bounds of connection probability for both *k-nearest neighbor* and *disk model* have been studied [12]. For *k-nearest neighbor*, [23] concludes that, as $n \to \infty$, if each node is connected to less than $0.074 \log n$ neighbors, the network is disconnected with probability one, while, if neighbors are more than $5.1774 \log n$, the network is connected with probability one. Reference [24] finds that if $k \leq 0.3043 \log n$, the network is not connected with high probability and if $k \geq 0.5139 \log n$, then network is connected with high probability as $n \to \infty$. But for the directed network the upper and lower bounds are $0.7209 \log n$ and $0.9967 \log n$, respectively. Reference [25] improves the upper bound to be $0.4125 \log n$. For *disk model* [26] states that 6 to 10 average numbers of neighbors almost make sure that network will be fully connected no matter how many nodes there are totally in the network. In [27], if communication range $\pi r^2 = \log n + b$, then the network connection probability tends to be $e^{-e^{-b}}$. Compared with [26], Table 1 in this paper shows that, when $n = 10000$, at least 13 neighbors are needed in order to make sure that network is connected with high probability. Besides, a result (Theorem 9) presented in our paper is the same as [27] but uses a totally different approach.

Reference [11] shows that, in *k-nearest neighbor model* by increasing $k$, network becomes $s$-connected very shortly after it becomes connected, where $s = O(\log n)$. Reference [28] proves one conjecture in [24] that, in *k-nearest neighbor model* for every $0 < \varepsilon < 1$ and $n$ sufficiently large, there exists $C = C(\varepsilon)$ such that, if the network has $k$-connected probability $\varepsilon$, then $(k + C)$-connected probability is bigger than $1 - \varepsilon$. This paper improves the results in [11], obtaining an explicit expression for *disk model*, that is, $s = \log n + b$, where $b > 1$. The corollary in this paper proves that the result for *disc model* has a similar form presented in [28].

Nodes having the same communication range usually are not true in reality. In order to make the model more accurate, [8, 29] utilize irregular radio to model real nodes. The connectivity for heterogenous networks has been well studied; for example, [16, 30] investigate the relay node placement problem such that network is the $k$-connected. The authors in [31] assumes that node communication radius $r_i$ of node $i$ is i.i.d. random variable with normal probability density $r_i \sim N(\mu, \sigma^2)$. Reference [32] adopts the model that Poisson intensity is given by a normal distribution; then it obtains the asymptotic bound of range that all nodes in this area are connected to the origin. Reference [33] considers nodes are placed according to a shot-noise Cox process rather than uniform deployment. This paper employs the stochastic methods to characterize heterogenous network. In this paper the density is maintained constant, but the node communication range is normal distribution.

## 6. Conclusion and Future Works

When deploying many WSNs nodes, one of the key problems is whether all nodes in the network are connected to other nodes. Isolated nodes will be useless for applications. This paper presents the results on how the network connection probability changes as the communication range varies in randomly and uniformly distributed homogenous and heterogenous WSNs. In case of network with all nodes having the same communication range, through theory derivation and validation, this paper proves that, regardless of network size, the network connection probability increases from $0.36\varepsilon$ to $1 - \varepsilon$ by increasing constant communication range of each node. As the example shows in Section 3.2, regardless of network size, if the network connection probability is 0.66%, by increasing the communication range until each

node obtains 6 more neighbors, the network connection probability reaches at least 98.17%. On the other hand, this paper shows that, once network is connected, it also becomes $\log n + b$-connected with high probability, which makes the network resilient against node failures because there are $\log n + b$ alternative paths between any two distinct nodes.

In case each node communication range is i.i.d random variable which has normal distribution, this paper analyzes the connection probability by simulation. This paper shows that network connection probability is determined by the distribution parameters and the network size, especially sensitive to standard deviation $\sigma$. The reason is that the network connection probability is dependent on the node that has minimal communication range. It implies that it needs to take care of the node which has minimal communication range because it is the bottleneck of the whole network. The network will become disconnected if they fail. With the same configuration, the resilience capability decreases when network size grows. Besides, given the required connection probability, this paper develops one method to decide the distribution parameter of communication range. This method can be used to choose appropriate distribution parameter of communication range for network simulators or real deployments.

In some circumstances, a full connected network is impractical and not necessary. One would be more interested in the giant connected component which contains most nodes of entire network are connected. More specifically, the relation between the giant connected component and the communication range distribution is what is wanted to be learnt. It is a percolation problem with random communication range. Percolation occurs when a node belongs to infinite component with none-zero possibility. The critical intensity $\lambda_c$ is defined as the minimum intensity in which percolation occurs. For *disk model*, the bound for critical intensity is known (e.g., [34]) but for variable radius is unknown. Therefore, studying the percolation problem with i.i.d communication range (or radius) will be our future work. On the other hand, the degree of the node obeys Poisson distribution in this paper. It has been found that many networks, such as the World Wide Web, the Internet, airplanes connection networks, some biological systems, and international ownership network, have power-law degree distribution with an exponent that ranges between 2 and 3 [35]. Our future work will center on connection probability with a more accurate model.

## References


[1] A. A. Aziz, Y. A. Sekercioglu, P. Fitzpatrick, and M. Ivanovich, "A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 1, pp. 121–144, 2013.

[2] C. Zhu, C. Zheng, L. Shu, and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619–632, 2012.

[3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[4] I. Saha, L. K. Sambasivan, S. K. Ghosh, and R. K. Patro, "Distributed fault-tolerant topology control in wireless multihop networks," *Wireless Networks*, vol. 16, no. 6, pp. 1511–1524, 2010.

[5] P. C. Pinto and M. Z. Win, "Communication in a poisson field of interferers—part I: interference distribution and error probability," *IEEE Transactions on Wireless Communications*, vol. 9, no. 7, pp. 2176–2186, 2010.

[6] Adrian Bondy and U. S. R. Murty, *Graph Theory*, Springer, Berlin, Germany, 2008.

[7] P. K. Chong and D. Kim, "Surface-level path loss modeling for sensor networks in flat and irregular terrain," *ACM Transactions on Sensor Networks*, vol. 9, no. 2, pp. 1–32, 2013.

[8] K. Zhu, "Ad hoc network deployment accommodating short and uncertain transmission range," in *Proceedings of the 9th ACM International Symposium on Mobility Management and Wireless Access (MobiWac '11)*, pp. 101–108, New York, NY, USA, November 2011.

[9] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proceedings of the 3rd ACM International Symposium On Mobile Ad Hoc Networking & Computing (MOBIHOC '02)*, pp. 80–91, Lausanne, Switzerland, June 2002.

[10] M. D. Penrose, "On k-connectivity for a geometric random graph," *Random Structures and Algorithms*, vol. 15, no. 2, pp. 145–164, 1999.

[11] P. Balister, B. Bollobás, A. Sarkar, and M. Walters, "Highly connected random geometric graphs," *Discrete Applied Mathematics*, vol. 157, no. 2, pp. 309–320, 2009.

[12] P. Balister, A. Sarkar, and B. Bollobs, "Percolation, connectivity, coverage and colouring of random geometric graphs," in *Handbook of Large-Scale Random Networks*, pp. 117–142, Springer, Berlin, Germany, 2008.

[13] R. Van Der Hofstad, "Random graphs and complex networks," http://www.win.tue.nl/~rhofstad/NotesRGCN2010.pdf.

[14] S. Delattre, C. Y. Robert, and M. Rosenbaum, "Estimating the efficient price from the order flow: a Brownian Cox process approach," *Stochastic Processes and Their Applications*, vol. 123, no. 7, pp. 2603–2619, 2013.

[15] H. Cai, X. Jia, and M. Sha, "Critical sensor density for partial connectivity in large area wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 7, no. 4, article 35, 2011.

[16] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 5, pp. 643–656, 2010.

[17] J. L. Bredin, E. D. Demaine, M. Hajiaghayi, and D. Rus, "Deploying sensor networks with guaranteed capacity and fault tolerance," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 309–319, Urbana-Champaign, III, USA, May 2005.

[18] J. Yu, S. Ren, S. Wan, D. Yu, and G. Wang, "A stochastic coverage scheduling algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 746501, 11 pages, 2012.

[19] Z. Mi, Y. Yang, and H. Ding, "Self-organized connectivity control and optimization subjected to dispersion of mobile ad hoc sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 672436, 15 pages, 2012.

[20] M. Bishop, M. Carvalho, R. Ford, and L. M. Mayron, "Resilience is more than availability," in *Proceedings of the New Security Paradigms Workshop (NSPW '11)*, pp. 95–104, Marin County, Calif, USA, September 2011.

[21] S. Dasgupta and G. Mao, "On the quality of wireless network connectivity," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '12)*, vol. 500, no. 505, pp. 3–7, December 2012.

[22] S. Banerjee, S. Shirazipourazad, P. Ghosh, and A. Sen, "Beyond connectivity—new metrics to evaluate robustness of networks," in *Proceedings of the IEEE 12th International Conference on High Performance Switching and Routing (HPSR '11)*, pp. 171–177, July 2011.

[23] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[24] P. Balister, B. Bollobás, A. Sarkar, and M. Walters, "Connectivity of random k-nearest-neighbour graphs," *Advances in Applied Probability*, vol. 37, no. 1, pp. 1–24, 2005.

[25] M. Walters, "Small components in k-nearest neighbour graphs," *Discrete Applied Mathematics*, vol. 160, no. 1314, pp. 2037–2047, 2012.

[26] J. Ni and S. A. G. Chandler, "Connectivity properties of a random radio network," *IEE Proceedings*, vol. 141, no. 4, pp. 289–296, 1994.

[27] M. D. Penrose, *Random Geometric Graphs*, Oxford University Press, Oxford, UK, 2003.

[28] V. Falgas-Ravry and M. Walters, "Sharpness in the k-nearest neighbors ranmdom geometric graph model," *Advances in Applied Probability*, vol. 44, no. 3, pp. 617–634, 2012.

[29] H. Chenji and R. Stoleru, "Mobile sensor network localization in harsh environments," in *Proceedings of the 6th IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 244–257, Springer, Santa Barbara, Calif, USA, 2010.

[30] D. R. Dandekar and P. R. Deshmukh, "Relay node placement for multi-path connectivity in heterogeneous wireless sensor networks," *Procedia Technology*, vol. 4, pp. 732–736, 2012.

[31] Z.-H. Guan, L. Ding, and Z.-M. Kong, "Multi-radius geographical spatial networks: statistical characteristics and application to wireless sensor networks," *Physica A*, vol. 389, no. 1, pp. 198–204, 2010.

[32] P. Balister, B. Bollobas, A. Sarkar, and M. Walters, "Connectivity of a Gaussian network," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 3, no. 3, pp. 204–213, 2008.

[33] G. Alfano, M. Garetto, E. Leonardi, and V. Martina, "Capacity scaling of wireless networks with inhomogeneous node density: lower bounds," *IEEE/ACM Transactions on Networking*, vol. 18, no. 5, pp. 1624–1636, 2010.

[34] S. C. Ng, G. Mao, and B. D. O. Anderson, "Analytical bounds on the critical density for percolation in wireless multi-hop networks," in *Proceedings of the 54th Annual IEEE Global Telecommunications Conference Energizing Global Communications (GLOBECOM '11)*, pp. 1–6, Houston, Tex, USA, December 2011.

[35] S. Vitali, J. B. Glattfelder, and S. Battiston, "The network of Global corporate control," *PLoS One*, vol. 6, no. 10, Article ID e25995, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 279483, 13 pages
http://dx.doi.org/10.1155/2013/279483

*Research Article*

# Incremental Localization Algorithm Based on Multivariate Analysis

**Xiaoyong Yan,[1,2] Huanyan Qian,[2] and Jiguang Chen[2]**

[1] *College of Information Technology, Jingling Institute of Technology, Nanjing 211169, China*
[2] *School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*

Correspondence should be addressed to Huanyan Qian; hyqian@mail.njust.edu.cn

Received 13 May 2013; Accepted 1 July 2013

Academic Editor: Qin Xin



In view of the traditional increment localization method, only the heteroscedasticity caused by the error accumulation is considered unilaterally a kind of incremental localization algorithm based on multivariate analysis is proposed. The algorithm combines the feasible weighted least squares (FWLS) in the multianalysis with the canonical correlation regression (CCR) and utilizes the FWLS to solve the heteroscedasticity caused by the error accumulation in the process of incremental localization; in the process of estimation, the CCR is used to solve the topology problems between the original beacon nodes and new beacon nodes. The simulation results show that the method can not only effectively restrain the influence caused by the accumulative errors but also can adapt to the different node topological shapes, so as to improve the positioning accuracy of the nodes.

## 1. Introduction

In many application problems related with sensor network, location information of nodes is of great importance to the monitoring activity of the whole network, which plays a critical role in many applications [1]. Monitoring data without nodes' location information is often of no use. 80% of information provided by sensor nodes to users related with the monitored area is connected with location [2]. In the application of wireless sensor network, nodes' location information can be acquired by adding GPS to nodes. However, this is only applicable with outdoor and open-sided circumstances. Besides, GPS is large in volume and high in cost and energy consumption. Moreover, GPS also needs stable base installations. These facts have made it difficult to realize the requirements of sensor network, which are "low price, low cost and low energy consumption" [3]. As for this, in the deploying area, only some of the nodes can be installed with GPS. For the rest nodes, their location information can only be calculated via a certain algorithm. However, in monitoring area, some unknown nodes not within communication radius of beacon nodes cannot be estimated out though ordinary localization algorithm concurrently at once for the reasons that communication radiuses of sensor nodes are limited by energy, randomness of nodes distribution, barrier between nodes, and so on, which will cause deficient monitoring area coverage; as a result, the quality of sensor network service decreases rapidly and is not able to effectively monitor the deployment area. The most frequently used solution is to increase the coverage of nodes through mobile beacon nodes [4], but the path, difficult reachability, and relatively larger power consumption of mobile nodes and others limit the application of this method. Incremental estimation algorithm [5, 6] is another method to enhance nodes coverage that has some advantages, such as, it does not need to consider path problems, and is not limited by actions of mobile nodes, and moreover consumes much less power compared with mobile nodes. Incremental algorithm will estimate unknown nodes near beacon nodes at first; these unknown nodes will act as new beacon nodes once their locations are determined, then locations of the rest unknown nodes will be estimated by newly added beacon nodes together with original beacon nodes and so on in a similar way, locations of all nodes were estimated out.

Incremental localization is a kind of low energy consumption positioning method which can effectively solve the coverage rate of the monitoring area; through outward extension in turn, each node is localized successively. Due to the successive localization, the previous estimation error will be bound to affect the following estimated accuracy. Such error accumulation inevitably leads to the inconsistency of the variance between the previous error term and the following localized error term; such phenomenon is also known as heteroscedasticity [7, 8]. In the process of location estimation, the heteroscedasticity appears, then the traditional ordinary least squares (OLS) is used to estimate the location of unknown nodes; the estimated value of the obtained node coordinate may not be the efficient estimator, even not the asymptotically efficient estimator. In order to correct the adverse effect caused by the heteroscedasticity, Meesookho et al. [9] proposed the weighted least squares (WLS); they used the reciprocal of error variance as the weighting of weight to restrain the error propagation. WLS is considered to be the improved OLS method; similar to OLS, the residual sum of squares is solved firstly by WLS and followed by the minimum value. However, in the process of finding the residual sum of squares, unlike OLS, WLS considers the influence of heteroscedasticity. In view of considering the heteroscedasticity based on location estimation by WLS, the location accuracy is improved through corresponding different weights with different data. Subsequently, Xiong et al. [10] proposed a kind of incremental node localization method with the optimal weighted least squares on the basis of WLS; this method is based on the obtained optimal weighted least squares when the error variance matrix is estimated as the minimal. Ji and Liu [5] proposed another strategic improved incremental localization approach (IILA) with the hypothesis that previous localization accuracy is greater than next one during incremental localization, this means that the estimation of location nearer to original beacon nodes is more accurate; on the basis of this assumption, with estimated distance of previous location as a constraint condition, the localization problem is converted into trust region sequences that can be solved by sequential quadratic programming (SQP) method. However, IILA did not consider sensor network as a kind of multihop network that there are many paths to certain node, and neglected complexity of deployment environment but only assumed errors during localization process that definitely increase with increasing hop counts; that is, heteroscedasticity of localization process is only monotonically increasing. In complex monitoring environment, variation tendency of heteroscedasticity is difficult to predict, is not necessarily monotonically increasing, but also is possibly decreasing or concurrently increasing and decreasing. For example, as shown in Figure 1, in monitoring area, there are many paths from node A to unknown node D, such as A → B → C → D and A → E → D due to environment complexity of monitoring area, barrier or interference sources exist between node A and node E, and accuracy of measurement from node A to Node E is far less than that of other nodes; as a result, it is not appropriate to estimate ED distance if AE distance acts as the constraint condition.

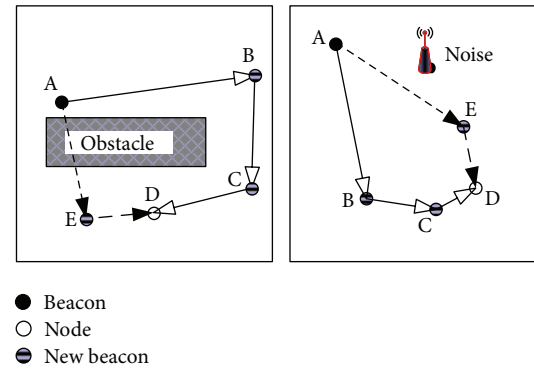



Figure 1: Localization in complex environment.

In addition, there is a problem that error of locations through estimation has directivity, for example, in Figure 1, errors between node A and node E could along direction of $\overrightarrow{AE}$ also could along $\overleftarrow{AE}$; similarly, errors between node E and node D also have directivity; if the direction of errors between node A and node E is in opposite direction of errors between node E and node D, errors between node E and node D is possibly less than those between node A and node E therefore, the assumption of IILA would not be valid any more.

In pervious incremental algorithms, most incremental localization algorithms are used to adjust heteroscedasticity during localization process and assume that heteroscedasticity is only monotonically increasing but fail to take deployment environment and networking features of sensor network into account. Sensor network is a kind of multihope network with relatively worse deployment environment, and incremental pattern of its heteroscedasticity is complicated and diversified for incremental localization algorithm. Furthermore, as same as concurrent localization algorithm, the accuracy of incremental localization is influenced by topology quality of beacon nodes also, and multicollinearity problem [11, 12] caused by topological relations among beacon nodes is not considered by previous incremental algorithms. For these reasons, we will propose a feasible incremental localization algorithm, (Location Estimation-FWLS-CCR) LE-FWLS-CCR, which uses less beacon nodes and considers multi-hops features of sensor network, error accumulation, heteroscedasticity and multicollinearity, and other problems. This algorithm will resolve heteroscedasticity problem through feasible weighted least squares (FWLS) [13] iterative computation mode. The iterative process is more proper for multi-hops features, while in estimation process, we adopts Canonical Correlation Regression (CCR) [12, 14–16] multivariate analysis to solve errors exist in newly added beacon nodes as well as topological relations between newly added and original beacon nodes.

## 2. Relevant Knowledge Review

*2.1. Feasible Weighed Least Squares.* In a concurrent localization process, distance-coordinates formula is generally transformed into a form of $\mathbf{Ax} = \mathbf{b} + \xi$ [11], in which $\xi$

is distance-measuring error term. The hypothesis to deduce locations of unknown nodes by distance-coordinates formula is variance of distance-measuring errors, $\mathrm{Var}(\xi) = \sigma^2\mathbf{I}_n$ ($\sigma^2$ is a constant, $\mathbf{I}_n$ is unit matrix), and in estimation algorithm, constant variance of errors is also known as homoscedasticity [7, 8]. While in location estimation by incremental algorithm, distance-measuring error is not inevitable, so variance of distance-measuring errors ($\mathrm{Var}(\xi) = \sigma^2\Omega \neq \sigma^2\mathbf{I}_n$) in step-by-step localization process would not be a constant, and this manifestation is called as heteroscedasticity. Because incremental localization process is complicated, there are abundant heteroscedasticity problems in it. Due to existence of heteroscedasticity, the results of typical location estimation models are not accurate and effective.

In the presence of heteroscedasticity, the positions of localization data in location estimation are different; smaller variance of error term of data means higher confidence level of residuals, while bigger variance of error term means lower confidence level of residuals. Therefore, as to the estimation of location in coordinates under the circumstance that heteroscedasticity exists, it usually uses weighted least squares method to discriminate different residuals [17], namely, to pay adequate attention to data terms with relatively smaller residuals and so assign larger weights on them and smaller weights on data term with larger residual in order to adjust the effect of various data items on estimation computation, and therefore to obtain an effective estimation value in localization process.

So, variance of error term of distance-coordinates formula exhibits heteroscedasticity which is expressed as

$$\mathrm{Var}(\xi) = \sigma^2\Omega, \tag{1}$$

among which $\sigma^2$ represents a constant; $\Omega$ denotes the $n$-order symmetric positive definite matrices. It is easy to understand that a $n$-order invertible matrix must exist, so as to make the following formula true.

$$\Omega = \mathbf{D}\mathbf{D}^T \Longrightarrow \mathbf{D}^{-1}\Omega\left(\mathbf{D}^T\right)^{-1} = \mathbf{I}_n. \tag{2}$$

$\mathbf{D}^{-1}$ is multiplied at both sides of the equation $\mathbf{A}\mathbf{x} = \mathbf{b} + \xi$

$$\mathbf{D}^{-1}\mathbf{A}\mathbf{x} + \mathbf{D}^{-1}\xi = \mathbf{D}^{-1}\mathbf{b}. \tag{3}$$

Assume, $\mathbf{b}^* = \mathbf{D}^{-1}\mathbf{b}$, $\mathbf{A}^* = \mathbf{D}^{-1}\mathbf{A}$, $\xi^* = \mathbf{D}^{-1}\xi$, then (3) can be converted into

$$\mathbf{b}^* + \xi^* = \mathbf{A}^*\mathbf{x} \tag{4}$$

Then the variance of the error term is

$$\begin{aligned}
\mathrm{Var}(\xi^*) &= E\left[\xi^*(\xi^*)^T\right] = E\left[\mathbf{D}^{-1}\xi\left(\mathbf{D}^{-1}\xi\right)^T\right] \\
&= E\left[\mathbf{D}^{-1}\xi\xi^T\left(\mathbf{D}^{-1}\right)^T\right] \\
&= \mathbf{D}^{-1}E\left[\xi\xi^T\right]\left(\mathbf{D}^{-1}\right)^T \\
&= \mathbf{D}^{-1}\sigma^2\Omega\left(\mathbf{D}^{-1}\right)^T \\
&= \sigma^2\mathbf{D}^{-1}\Omega\left(\mathbf{D}^{-1}\right)^T \\
&= \sigma^2\mathbf{I}_n.
\end{aligned} \tag{5}$$

Then, the heteroscedasticity of the error term is eliminated, and it is easy to learn that $E(\xi^*) = 0$. Obviously, the error term $\xi^*$ in (5) meets the assumption of the least squares model; therefore, there is the loss equation $S(\mathbf{x})$:

$$\begin{aligned}
S(\mathbf{x}) &= \left(\xi^*\right)^T\xi^* \\
&= \left(\mathbf{b}^* - \mathbf{A}^*\mathbf{x}\right)^T\left(\mathbf{b}^* - \mathbf{A}^*\mathbf{x}\right) \\
&= (\mathbf{b} - \mathbf{A}\mathbf{x})^T\Omega^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}).
\end{aligned} \tag{6}$$

In order to obtain the optimal solution, we must make

$$\min(\mathbf{b} - \mathbf{A}\mathbf{x})^T\Omega^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}). \tag{7}$$

It is assumed that $\widehat{\mathbf{x}}_{\mathrm{WLS}}$ is the minimized optimal solution. Therefore, $\widehat{\mathbf{x}}_{\mathrm{WLS}}$ meets the minimal least squares equation solution as follows:

$$\left(\mathbf{A}^T\Omega^{-1}\mathbf{A}\right)\widehat{\mathbf{x}}_{\mathrm{WLS}} = \mathbf{A}^T\Omega^{-1}\mathbf{b}. \tag{8}$$

Obviously, if the row vector of $\mathbf{A}$ is linearly independent, then the row vector of $\mathbf{A}^*$ is linearly independent. So, $(\mathbf{A}^*)^T\mathbf{A}^* = (\mathbf{A}^*)^T\Omega^{-1}\mathbf{A}^*$ is reversible; thus, the optimal solution of (8) is expressed as

$$\widehat{\mathbf{x}}_{\mathrm{WLS}} = \left(\mathbf{A}^T\Omega^{-1}\mathbf{A}\right)^{-1}\mathbf{A}^T\Omega^{-1}\mathbf{b}. \tag{9}$$

Through Schwarz inequality [17, 18], it is proved that when the matrix $\Omega$ is the reciprocal of the variance matrix of the range error under the condition that the ratio of range error to the distance is independent Gaussian random variables, the error variance by WLS is minimal. But in reality, the variance of the error term is unknown; therefore, if WLS is solved, the weight needs to be taken according to the actual situation.

FWLS is a feasible method which is able to overcome the problem that cannot be implemented by WLS due to the unavailable weight. FWLS uses residuals attained at each computation as weight matrix; therefore, real weight values can be acquired in computation process, and procedure of FWLS algorithm is shown in Algorithm 1.

It can be noted that the FWLS algorithm is in marching iteration; the derivation of the optimal estimated value $\widehat{\mathbf{x}}_i$ in

(1) Firstly, it is essential to estimate the model through OLS method and obtain the estimated value $\widehat{x}$; then substitute it into the equation, and obtain the residual error $\widehat{u}_0 = b - A\widehat{x}$ accordingly.
(2) Utilize the square of the residual term as the $\mathbf{\Omega}$ matrix, for example, $\widehat{\Omega}_i = \mathrm{diag}(\widehat{u}_{i,0}^2, \widehat{u}_{i,1}^2, \ldots, \widehat{u}_{i,n-1}^2)$
(3) Obtain the next-order estimated value and residual value by WLS
$\widehat{x}_{i+1} = (A^T\widehat{\Omega}_i^{-1}A)^{-1}A^T\widehat{\Omega}_i^{-1}b$
$\widehat{u}_{i+1} = b - A\widehat{x}_{i+1}$
(4) Go back to Step 2, until the number of iterative times meet the number of times according to the algorithm requirements.

Algorithm 1: FWLS algorithm.

each step is based on the assumption of nonexistent multicollinearity problem in $\mathbf{A}^T\widehat{\Omega}_i^{-1}\mathbf{A}$; unfortunately, by virtue of FWLS, it is feasible to eliminate the interference of heteroscedasticity, but the multicollinearity interference cannot be sure to eliminate. Therefore, in the process of iteration, it is essential to make corresponding strategies to avoid the algorithm insolubility caused by the multicollinearity.

*2.2. Canonical Correlation Regression.* In concurrent localization algorithm, beacon nodes have a very large influence on final location estimation and possibly cause significant errors when beacon nodes relations are collinear or approximately collinear [11]. Principal component analysis (PCA) in multivariate analysis will remove partial information through recombination of coordinate information of beacon nodes in order to reduce noise and effects of multicollinearity.

As to incremental localization algorithm, it can use PCA [19] method in the first estimation of coordinates of unknown nodes to avoid problems caused by multicollinearity in location estimation, but in incremental algorithm, measuring errors cannot be absolutely avoided or eliminated, which indicates that there would be always errors in locations of newly added beacon nodes in coordinates; therefore, it requires processing error information in output message. PCA only processes input variables; for incremental localization algorithm, its output variables act as locations of newly added beacon nodes, and errors contained in them shall be preprocessed to a certain extent.

Canonical correlation analysis (CCA) is a kind feasible and powerful multivariate analysis method especially appropriate for processing and analysis of two correlated data. At the same time, it is a kind of descending dimension method similar to PCA and is also able to remove some noise information that contains collinear information through recombination of data like PCA. CCA pays more attention to data processing and analysis of correlated data; for this reason, it is more proper for regression algorithm and has higher regression accuracy than PCA.

For the equation $\mathbf{Ax} = \mathbf{b} + \xi$, the solution procedure of CCA is as follows

Suppose that there are two groups of data, $\mathbf{A}$ and $\mathbf{b}$, which have been processed with centralization, $\mathbf{A} \in \mathbb{R}^p$, $\mathbf{b} \in \mathbb{R}^q$, CCA is mainly used to seek linear combination of $\mathbf{A}$ and $\mathbf{b}$, $\mathbf{w}_\mathbf{A}^T\mathbf{A}$ and $\mathbf{w}_\mathbf{b}^T\mathbf{b}$, respectively, making them correlate with each other maximally, that is to say, to find the maximum solution of following equation:

$$\begin{aligned}\rho &= \frac{E\left[\mathbf{w}_\mathbf{A}^T\mathbf{A}\mathbf{b}^T\mathbf{w}_\mathbf{b}\right]}{\sqrt{E\left[\mathbf{w}_\mathbf{A}^T\mathbf{A}\mathbf{A}^T\mathbf{w}_\mathbf{A}\right]E\left[\mathbf{w}_\mathbf{b}^T\mathbf{b}\mathbf{b}^T\mathbf{w}_\mathbf{b}\right]}} \\ &= \frac{\mathbf{w}_\mathbf{A}^T\mathbf{C}_{\mathbf{Ab}}\mathbf{w}_\mathbf{b}}{\sqrt{\mathbf{w}_\mathbf{A}^T\mathbf{C}_{\mathbf{AA}}\mathbf{w}_\mathbf{A}\mathbf{w}_\mathbf{b}^T\mathbf{C}_{\mathbf{bb}}\mathbf{w}_\mathbf{b}}},\end{aligned} \tag{10}$$

in which $\mathbf{C}_{\mathbf{AA}} \in \mathbb{R}^{p\times p}$, $\mathbf{C}_{\mathbf{bb}} \in \mathbb{R}^{p\times p}$ is within-set covariance matrix of variable $\mathbf{A}$ and $\mathbf{b}$, respectively, $\mathbf{C}_{\mathbf{Ab}} \in \mathbb{R}^{p\times q}$ means between-set covariance matrix, and moreover $\mathbf{C}_{\mathbf{Ab}} = \mathbf{C}_{\mathbf{Ab}}^T \in \mathbb{R}^{p\times q}$.

The correlation function $\rho$ is independent of scales of $\mathbf{w}_\mathbf{A}$ and $\mathbf{w}_\mathbf{b}$; by constraining respective within-set covariance $\mathbf{C}_{\mathbf{AA}}$ and $\mathbf{C}_{\mathbf{bb}}$ of $\mathbf{A}$ and $\mathbf{b}$, CCA can be formulated as solution of optimization problem of the following equation:

$$\begin{aligned}&\max_{\mathbf{w}_\mathbf{A},\mathbf{w}_\mathbf{b}} \mathbf{w}_\mathbf{A}^T\mathbf{C}_{\mathbf{Ab}}\mathbf{w}_\mathbf{b} \\ &\text{s.t.} \quad \mathbf{w}_\mathbf{A}^T\mathbf{C}_{\mathbf{AA}}\mathbf{w}_\mathbf{A} = 1, \quad \mathbf{w}_\mathbf{b}^T\mathbf{C}_{\mathbf{bb}}\mathbf{w}_\mathbf{b} = 1.\end{aligned} \tag{11}$$

To solve this optimization problem of (11), we can build a Lagrange equation to obtain the optimal solution; that is,

$$\begin{aligned}L\left(\mathbf{w}_\mathbf{A}, \mathbf{w}_\mathbf{b}, \lambda_1, \lambda_2\right) &= \mathbf{w}_\mathbf{A}^T\mathbf{C}_{\mathbf{Ab}}\mathbf{w}_\mathbf{b} \\ &\quad + \frac{1}{2}\lambda_1\left(1 - \mathbf{w}_\mathbf{A}^T\mathbf{C}_{\mathbf{AA}}\mathbf{w}_\mathbf{A}\right) \\ &\quad + \frac{1}{2}\lambda_2\left(1 - \mathbf{w}_\mathbf{b}^T\mathbf{C}_{\mathbf{bb}}\mathbf{w}_\mathbf{b}\right).\end{aligned} \tag{12}$$

Differentiating (12) with $\mathbf{w}_\mathbf{A}$ and $\mathbf{w}_\mathbf{b}$, partial derivatives are as follows:

$$\frac{\partial L}{\partial \mathbf{w}_\mathbf{A}} = \mathbf{C}_{\mathbf{Ab}}\mathbf{w}_\mathbf{b} - \lambda_1\mathbf{C}_{\mathbf{AA}}\mathbf{w}_\mathbf{A} \tag{13a}$$

$$\frac{\partial L}{\partial \mathbf{w}_b} = \mathbf{C}_{\mathbf{bA}}\mathbf{w}_\mathbf{A} - \lambda_2\mathbf{C}_{\mathbf{bb}}\mathbf{w}_\mathbf{b}. \tag{13b}$$

To obtain the optimal solution, let (13a), (13b), (14a),and (14b) equal to zero, then

$$\mathbf{C_{Ab}w_b} = \lambda_1 \mathbf{C_{AA}w_A}, \tag{14a}$$

$$\mathbf{C_{bA}w_A} = \lambda_2 \mathbf{C_{bb}w_b}. \tag{14b}$$

Multiply both sides of formula (14a) and (14b) by $\mathbf{w_A}$ and $\mathbf{w_b}$ from left, respectively, easily obtain $\lambda_1 = \lambda_2$, and take $\lambda_1 = \lambda_2 = \lambda$, then the above formula can be simplified into

$$\mathbf{C_{Ab}w_b} = \lambda \mathbf{C_{AA}w_A}, \tag{15a}$$

$$\mathbf{C_{bA}w_A} = \lambda \mathbf{C_{bb}w_b}. \tag{15b}$$

Given $\mathbf{C_{bb}}$ is reversible, from (15b), obtain $\mathbf{w_b} = (1/\lambda)\mathbf{C_{bb}^{-1}C_{bA}w_A}$, and then substitute it into (15a), and reorganize them into

$$\mathbf{C_{Ab}C_{bb}^{-1}C_{bA}w_A} = \lambda^2 \mathbf{C_{AA}w_A}\ , \tag{16a}$$

$$\mathbf{C_{bA}C_{AA}^{-1}C_{Ab}w_b} = \lambda^2 \mathbf{C_{bb}w_b}. \tag{16b}$$

Here, the solution of CCA was translated into generalized eigenvalue-eigenvector problem of two matrixes whose scales are $p \times p$, and $q \times q$ respectively. And then CCA problem is equally described as generalized eigenvalue problem of formula (17):

$$\begin{pmatrix} & \mathbf{Ab}^T \\ \mathbf{bA}^T & \end{pmatrix} \begin{pmatrix} \mathbf{w_A} \\ \mathbf{w_b} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{AA}^T & \\ & \mathbf{bb}^T \end{pmatrix} \begin{pmatrix} \mathbf{w_A} \\ \mathbf{w_b} \end{pmatrix}. \tag{17}$$

Formula (17) can be abridged into $\mathbf{Xw} = \lambda \mathbf{Yw}$, in which $\mathbf{X}$, $\mathbf{Y}$, respectively, correspondes to left and right matrix of previous formula, $\mathbf{w} = [\mathbf{w_A}^T, \mathbf{w_b}^T]^T$ therefore, $\mathbf{w_A}$ and $\mathbf{w_b}$ are eigenvectors of $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}(\mathbf{b}^T\mathbf{b})^{-1}\mathbf{b}^T\mathbf{A}$ and $(\mathbf{b}^T\mathbf{b})^{-1}\mathbf{b}^T\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$, respectively.

The literature [12, 15] proposed a regression method based on canonical correlation analysis, namely, canonical correlation regression (CCR). CCR combines least squares with canonical correlation analysis with the purposes of optimizating solution of regression coefficient under most relevant significance. CCA-based regression method, to some extent, avoids interferences of multicollinearity of samples through utilization of components that have been features-extracted for regression; in addition, CCA considers correlation between output and input invariables and so can be regarded as an advanced regression between two multivariates, an extension of multiple linear regression (MLR), and known as a "multi-to-multi" regression method. Regression coefficient of canonical correlation regression can be computed out by following equation:

$$\widehat{\mathbf{x}}_{\mathrm{CCR}} = \left(\mathbf{W}_k \mathbf{W}_k^T\right) \mathbf{x}^T \mathbf{b}, \tag{18}$$

in which, $\mathbf{W}_k = [\mathbf{w}_\mathbf{A}^1, \mathbf{w}_\mathbf{A}^2, \ldots, \mathbf{w}_\mathbf{A}^k]$ is composed of first $k$ of maximum eigenvector.

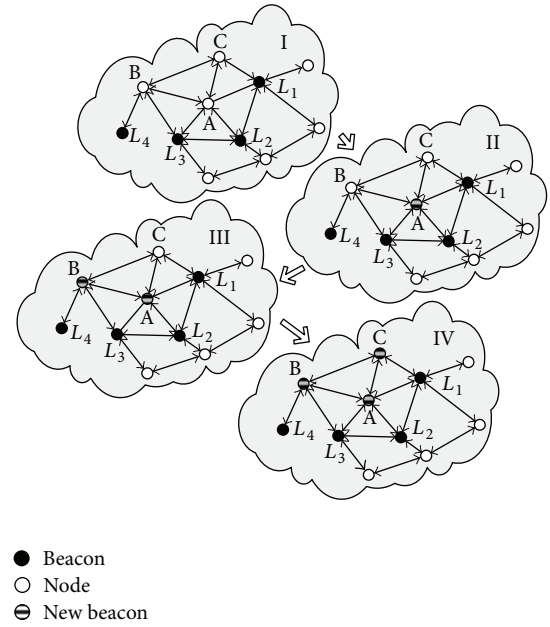



Figure 2: Example and phases of LE FWLS-CCR.

## 3. Methodology

*3.1. The LE-FWLS-CCR Algorithm.* The existence of multicollinearity usually brings seriously adverse effects on model estimation, testing, and prediction. During localization estimation, multicollinearity not only exists in concurrent localization but also exists in incremental location estimation process. For this reason, we add canonical correlation regression method into FWSL algorithm to acquire optimal prediction direction through correlation analysis of input and output variables and dimension reduction processing then use FWLS method to resolve problems caused by heteroscedasticity. Because the procedure of FWLS-CCR localization algorithm is similar to that of FWLS algorithm, its solution is carried out through iteration, and the algorithm process is shown in Algorithm 2.

Node localization process based on FWLS-CCR is shown in Figure 2. Assume that the monitoring area is deployed with several sensor nodes and $L_1$, $L_2$, $L_3$, and $L_4$ are original beacon nodes, which are set to be zero-level beacon nodes. The nodes of A, B, and C are the nodes to be localized. Node A is directly connected with the three original beacon nodes of $L_1$, $L_2$, and $L_3$. Node B is connected with $L_3$ and $L_4$, while node C is only connected with $L_1$.

Obviously, the coordinates of node A can be calculated and estimated according to the beacon nodes of $L_1$, $L_2$, and $L_3$. According to CCR incremental localization algorithm, node A may be updated to a new beacon node after adopting CCR algorithm to calculate its estimated coordinates, and the node may be set as the first level beacon node. By calculating the residual, matrix $\Omega$ can be obtained. Assuming that $\mathbf{b}^* = \mathbf{D}^{-1}\mathbf{b}$, $\mathbf{A}^* = \mathbf{D}^{-1}\mathbf{A}$, the location calculation equation is transformed into the form of $\mathbf{b}^* + \xi^* = \mathbf{A}^*\mathbf{x}$. taking original beacon nodes $L_3$, $L_4$, and the newly added beacon node A

Input: Beacon nodes coordinates:
Distance from beacon nodes to unknown nodes $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}, m \geq 3$

Output: Estimated coordinates of unknown nodes: $\{\widehat{\mathbf{x}}_{m+1}, \widehat{\mathbf{x}}_{m+2}, \ldots, \widehat{\mathbf{x}}_n\}$

(1) Beacon nodes deliver their location information outwards through controllable flooding, if unknown nodes acquire more than 3 beacon nodes, it firstly uses CCR to estimate locations of unknown nodes, and will stop if there is no rest unknown nodes, otherwise, will carry out next step.
(2) Uses estimated location to estimate residual vector by the estimation formula: $\widehat{\mathbf{u}}_0 = \mathbf{b} - \mathbf{A}\widehat{\mathbf{x}}_{\mathrm{CCR}}$;
(3) Constructs covariance matrix through FWLS method residual vectors, $\widehat{\mathbf{\Omega}}_i = \mathrm{diag}\left(\widehat{u}_{i,0}^2, \widehat{u}_{i,1}^2, \ldots, \widehat{u}_{i,n-1}^2\right)$;
(4) Uses newly-constructed covariance matrix to rewrite location distance equation, $\mathbf{Ax} = \mathbf{b} + \xi$;
(5) According to new equation, uses CCR method to estimate locations of secondary beacon nodes.
(6) If there still are some nodes which locations have not be estimate out in deployment area, skip to Step 2.
(7) The algorithm will finish if there is no node to be estimated in deployment area, and it will output coordinates of unknown nodes.

Algorithm 2: LE-FWLS-CCR algorithm.

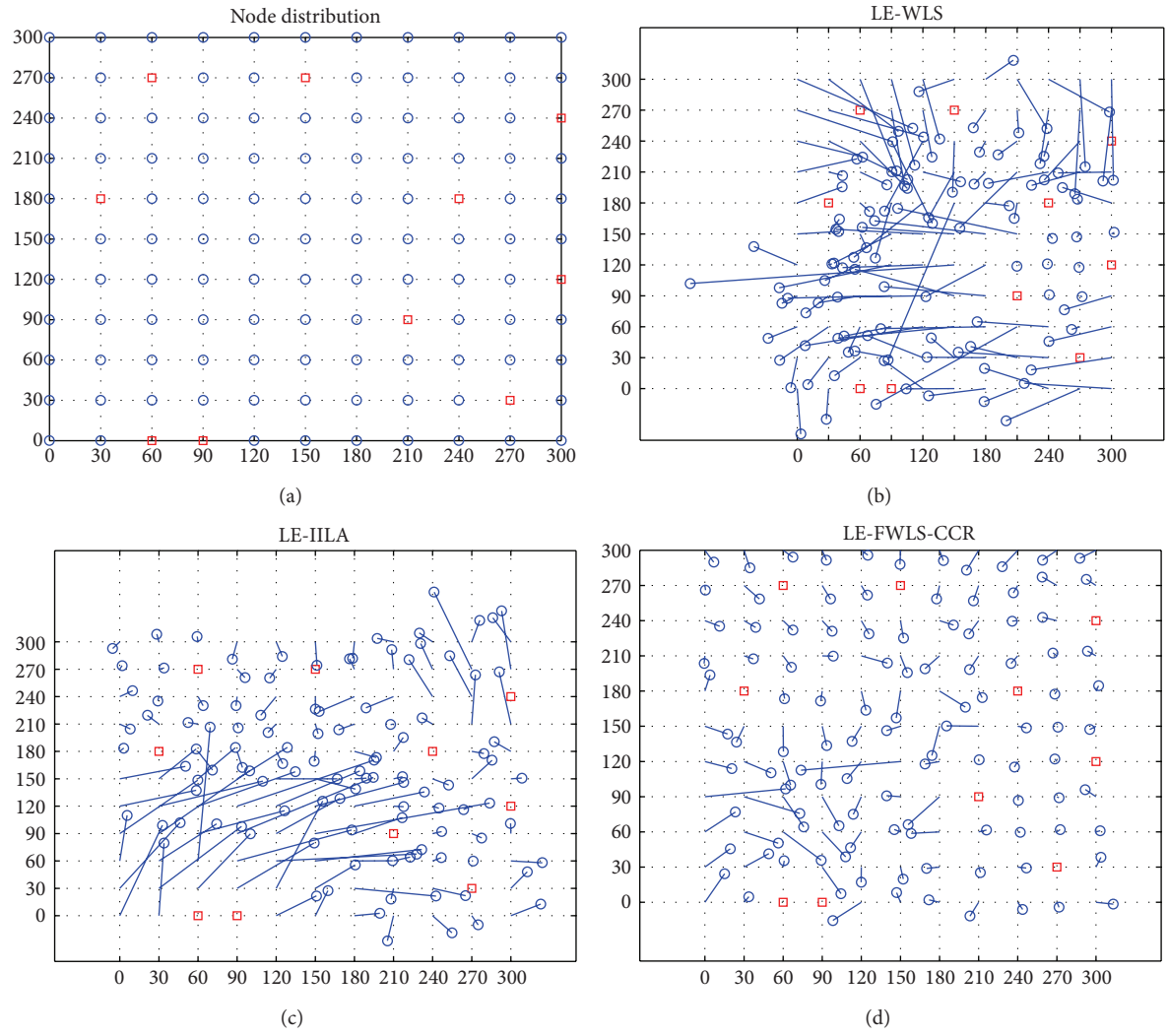



Figure 3: Localization results of regular deployment in square area.

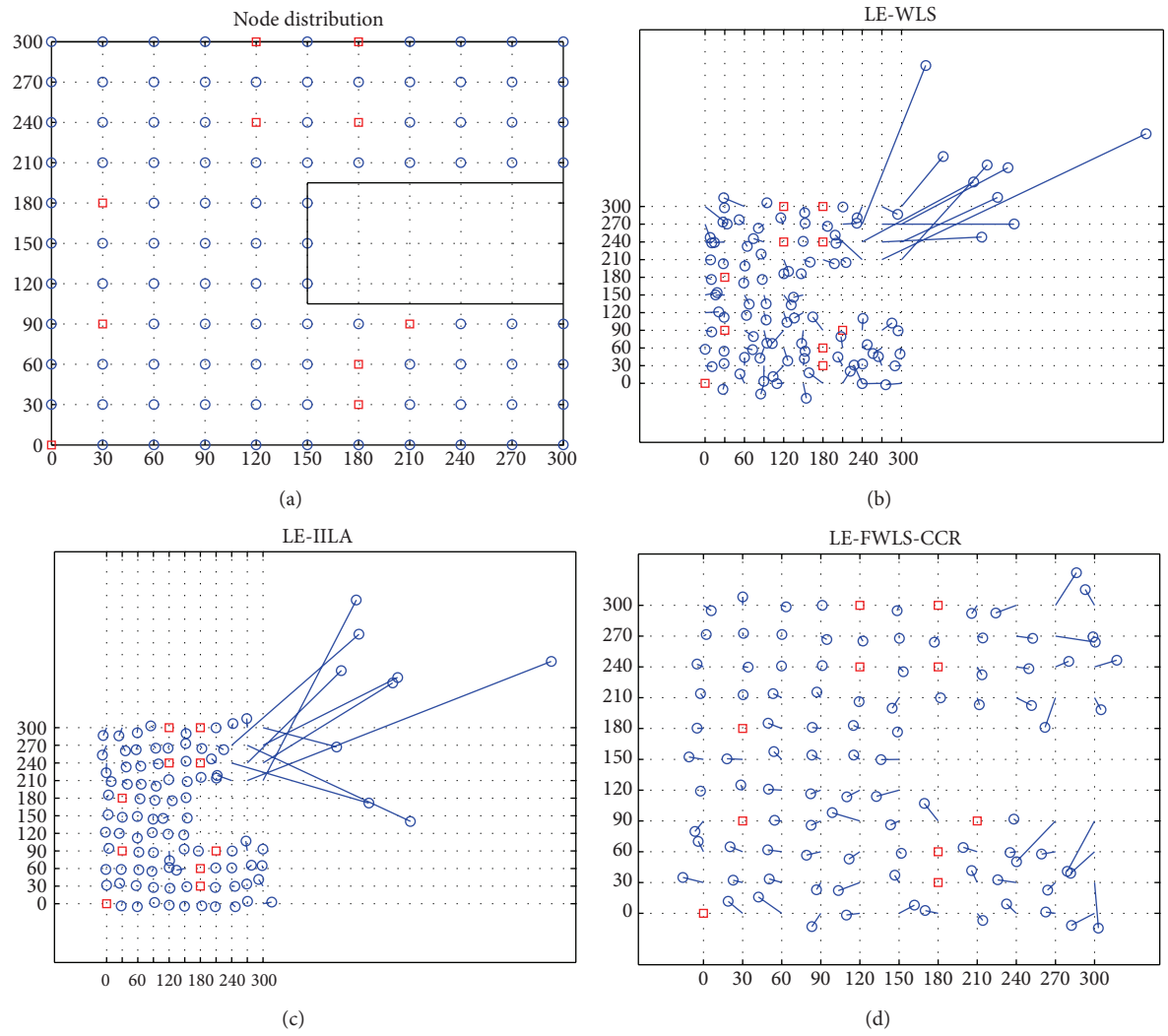



Figure 4: Localization results of regular deployment in C-shape area.

as the referential nodes, and then readopt CCR algorithm to figure out the estimated coordinates of node B. After the calculation, node B is added as a new beacon node, with second level. Similarly, the location of node C can be estimated based on beacon $L_1$, node A, and node B. Node C is defined to be a third level beacon node.

In incremental localization approach, localization of nodes is implemented in batches. Owing to distance measurement error, there is a certain difference between the estimated value and the practical value of first level beacon node. As for second level beacon nodes, their estimated values are as well influenced by measurement error and the intrinsic error of first level beacon nodes. In addition, it is shown in Figure 2 that, when estimating node C, the positions of referential nodes $L_1$, A, and B are founded to be approximately collinear, which may lead to even larger error when estimating node C. On this basis, FWLS-CCR method is applied to utilize CCR to reduce the dimensionality of input and output data after the data is analyzed. The objective is to partially eliminate error and multicollinearity. Meanwhile, attainable residuals are adopted as the weight value, making the algorithm even feasible and applicable to practical environment. Thus, compared with previous algorithms, localization algorithm based on FWLS-CCR is more feasible, with higher adaptability.

*3.2. Time Complexity Analysis.* In this Section, we compare the time complexities for localization as required by LE-FWLS-CCR and other popular incremental location estimation algorithms, namely, WLS-based location estimation (LE-WLS) proposed in the literature [10] as well as SQP-based location estimation (LE-IILA) proposed in the literature [5]. These algorithms will also be compared experimentally in Section 4.

LE-FWLS-CCR: basically, the complexity of our algorithm is dominated by two parts: canonical correlation regression and feasible weighted least squares. The complexity of computing CCR is mainly determined by the core algorithm CCA, whose complexity is $O(n^3 \log n)$. FWLS method is OLS improvement, which uses residuals attained at

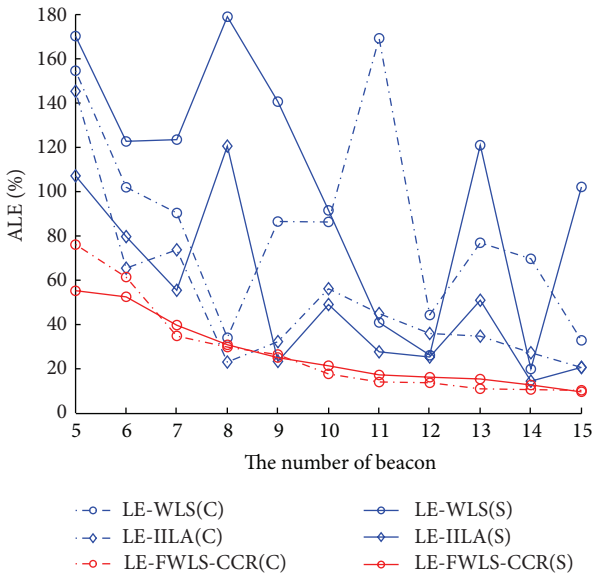



Figure 5: Average localization error of regular deployment.

each computation as weight matrix; therefore, complexity of FWLS is $O(n^4)$. Thus, the total complexity of the FWLS-CCR is $O(n^4)$.

LE-WLS: WLS-based location estimation method using the WLS method as the core algorithm, and the complexity of WLS is $O(n^3)$; thus, the computer complexity of LE-WLS method is $O(n^3)$.

LE-IILA: the complexity of SQP method is the main reason in LE-IILA algorithm. If the limited number of iterations, the computational complexity of SQP is $O(k^2+kn)$, where $k$ is the number of iterations. If $n \gg k$, the complexity of computer is $O(kn)$, otherwise the complexity of computer is $O(k^2)$.

The literature [20] has proved that properly increasing the calculation amount of algorithm will not affect the performance of sensor network. For this reason, it is worthy to improve the localization accuracy based on FWLS-CCR by sacrificing partial calculation volume.

## 4. Simulation and Experiments

This section will analyze and evaluate LE-FWLS-CCR localization algorithm on Matlab platform. In simulation experiment, it is supposed that nodes are deployed in a two-dimensional monitoring area and adopt transformation of RSSI signals to distance for matrix of distance among nodes. In order to compare impartiality of experimental results, this section adopts signal model proposed in the literature [21] to simulate signal strength among nodes; that is,

$$P_{ij} \sim N\left(\overline{P}_{ij}, \sigma_{dB}^2\right), \quad \overline{P}_{ij} = P_0 - 10n_p \lg\left(\frac{d_{ij}}{d_0}\right), \tag{19}$$

among which $P_{ij}$ represents the transmitted signal power which is received by node $i$ from the node $j$, and the unit is dBm; $P_0$ represents the received signal power corresponding to the point of the reference range $d_0$; $d_0$ represents the reference range; $n_p$ represents the attenuation coefficient of the wireless transmission and is related to the environment; $\overline{P}_{ij}$ represents the received signal power corresponding to the point of the reference range $d_0$ (dBm); $\sigma_{dB}^2$ represents the shadow variance. $n_p$ uses fitting data from real collection in the literature [20]; as for $\sigma_{dB}^2$, let $\sigma_{dB}^2/n_p = 1.2$ in this experiment.

Due to higher coverage of incremental algorithm, the experiment in this section mainly examines the accuracy of localization of nodes with ALE as evaluation basis, and the definition of ALE is as follows:

$$\text{ALE} = \frac{\sum_{i=1}^{n} \sqrt{(\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2}}{n \times R} \times 100\%. \tag{20}$$

In the formula, $(\hat{x}_i, \hat{y}_i)$ represents the estimated coordinate location of the $i$th node, $(x_i, y_i)$ represents the actual coordinate location of the $i$th node $n$ represents the number of the unknown nodes, and $R$ represents the communication radius. It can be seen from the above formula that ALE refers to the ratio of the average error of the Euclidean distance from the estimation location of all nodes to the real location in the area to the communication radius. ALE can reflect the stability of the localization algorithm and the positioning accuracy; when the communication radius of the node is given, if the average localization error of the node is smaller, then the positioning accuracy of the algorithm is higher and vice versa.

This experiment also compares the method proposed in this paper, the localization algorithm based on FWLS-CCR (LE-FWLS-CCR), with WLS-based location estimation (LE-WLS) proposed in the literature [10], as well as IILA-based location estimation (LE-IILA) proposed in the literature [5]. In addition, the experiment carries out comparison by use of data collected from actual scenes provided in the literature [20].

*4.1. Simulation Experiments Based on Distance-Measuring Model.* The experiments based on distance-measuring model have set four experimental scenes: random deployment nodes in square area, regular deployment nodes in square area, random deployment nodes in C-shape area and regular deployment nodes in C-shape area, in which C-shape area, is formed because of a bigger barrier, mainly used to evaluate localization performance with lager barrier, that is, in case of non-line-of-sight. In order to decrease the effect of single one experiment, each group of experiments will be repeated for 50 times in each scene, finally the average indicators of the 50 experiments will be reported. The experiments will examine accuracy of final localization results of unknown nodes with incremental quantity of beacon nodes. In these experiments, the valid communication radius of nodes is supposed to be 60 m.

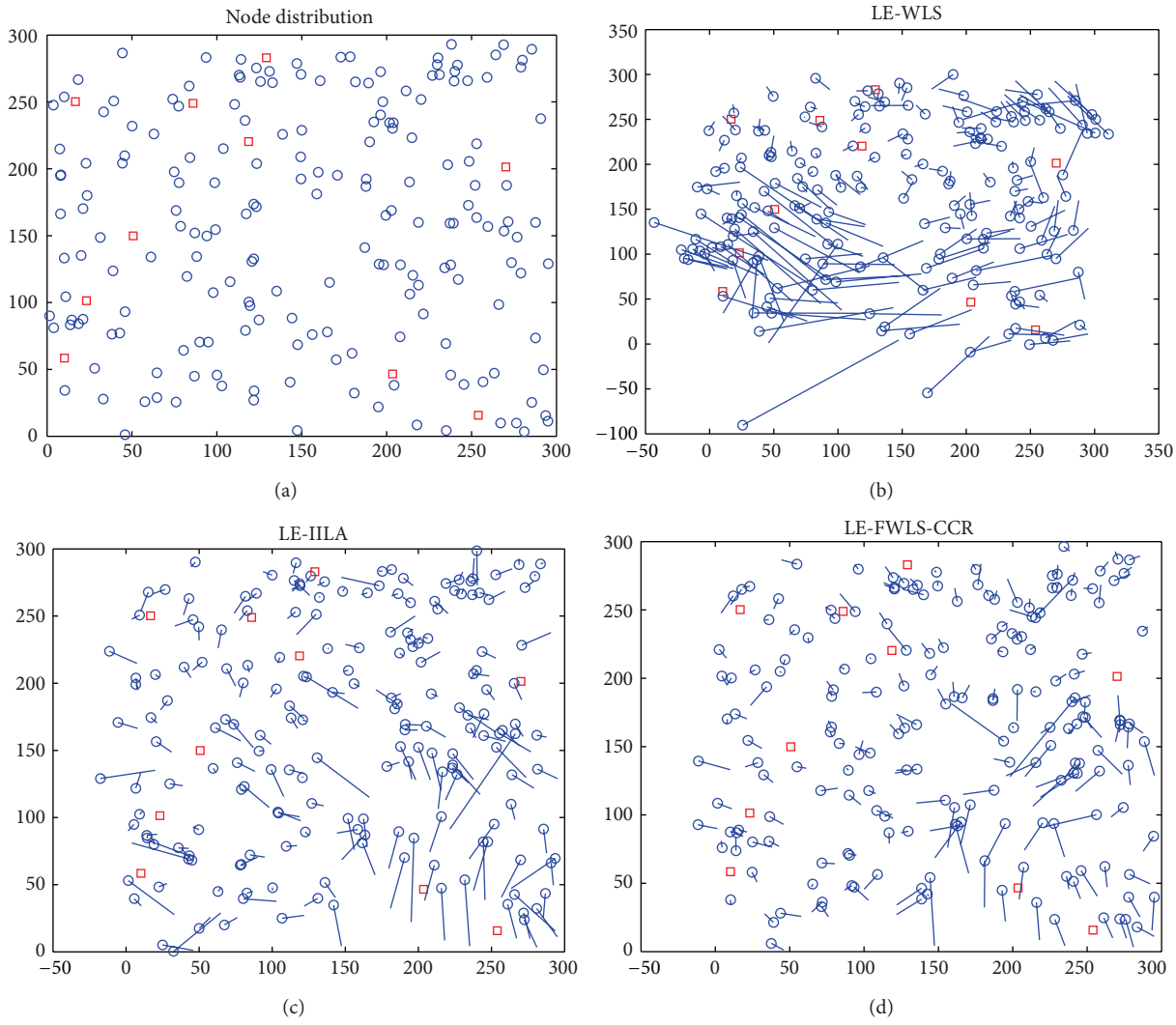



Figure 6: Localization results of random deployment in square area.

*4.1.1. Rules Deployment.* Regular deployment of nodes in monitoring area principally aims to explore effects of collineation of beacon nodes on localization accuracy, while regular deployment in C-shape area is used to observe effects of non-line-of-sight caused by barriers in monitoring area on localization accuracy.

In this group of experiments, regular deployment of nodes is within a 300 m × 300 m area, whose side length of grids is 30 m, there are 121 nodes in total without barrier, but a 150 m×90 m barrier was placed in C-shape area, and then the quantity of nodes becomes 106. In these experiments, select 5–15 nodes as beacon nodes, provided that their location information is known. Figure 3 is the final localization result of nodes of certain deployment in square area under the circumstance that there are 10 beacon nodes, in which circle denotes unknown node, box denotes beacon node, and the straight line connects actual coordinates of unknown node with its estimated coordinates. Figure 3(a) shows deployment of nodes; Figure 3(b) shows localization results based on weighted least squares with weights being reciprocals of variance of error term that is optimal theoretically, and in this figure, ALE = 70%; Figure 3(c) shows LE-IILA method proposed by Ji, and in this figure, ALE = 43.2%; Figure 3(d) shows LE-FWLS-CRR proposed in this paper, and in this figure, ALE = 18.2%.

The circumstance that there is a barrier in regular deployment area was described in Figure 4(b) to Figure 4(d), and this experiment mainly is used to study effects of non-line-of-sight on localization results. Figure 4(a) shows deployment of nodes; Figure 4(b) shows localization results of LE-WLS, ALE = 46.8%; Figure 4(c) shows LE-IILA method proposed by Ji, ALE = 42.2%; Figure 4(d) shows LE-FWLS-CRR proposed in this paper, ALE = 16.1%.

It can be seen from Figures 3 and 4 that LE-WLS and LE-IILA only considered heteroscedasticity but didn't take multicollinearity into account; therefore, only nodes in partially incremental area obtained satisfactory results in experiments, and localization errors in some area are still large. LE-FWLS-CCR method proposed in this paper comprehensively considered heteroscedasticity, error escalation as well as

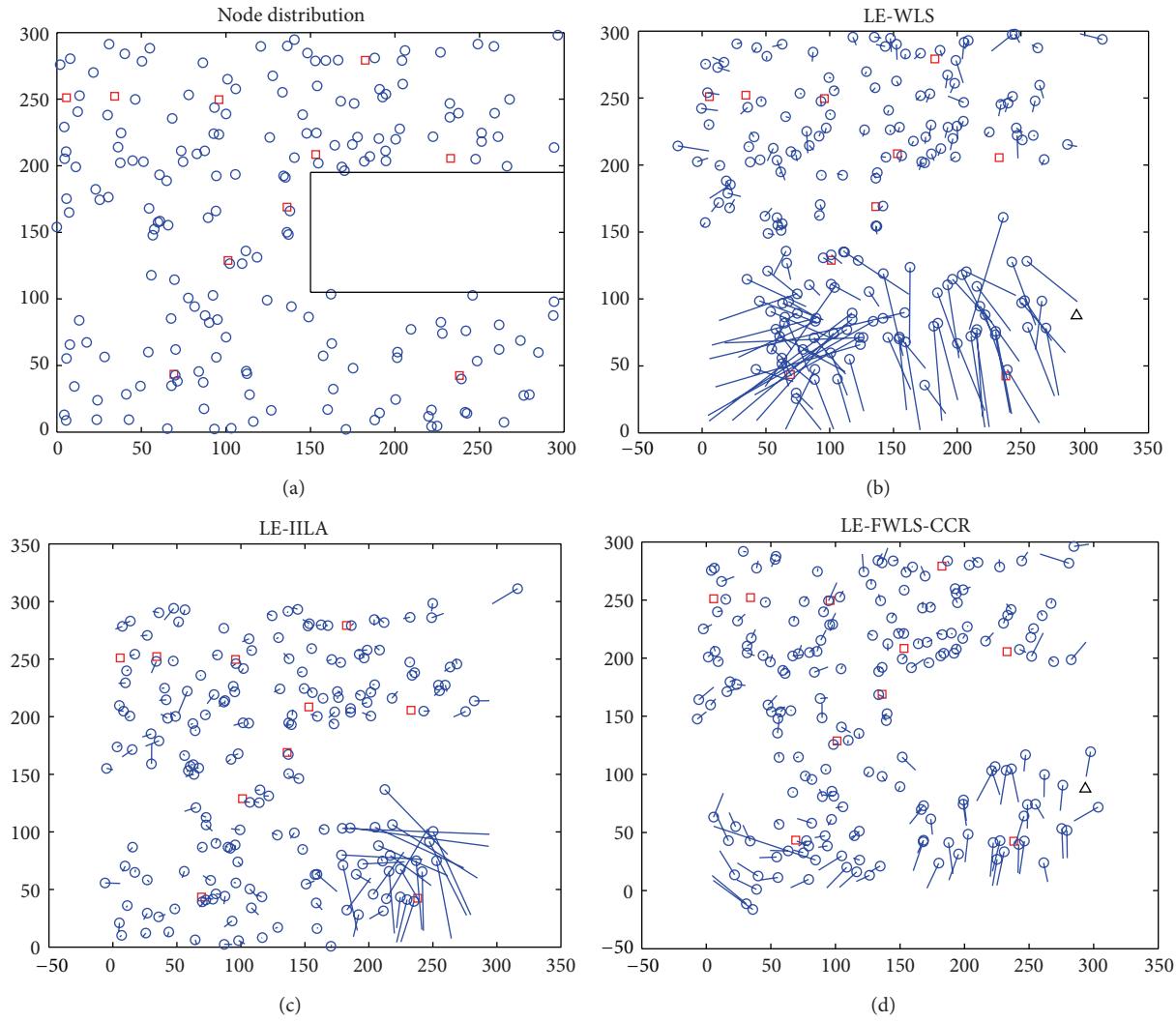



Figure 7: Localization results of random deployment in C-shape area.

multicollinearity, and other problems, and so localization results are more effective than those of LE-WLS and LE-IILA methods.

Because incremental localization method is used to locate nodes through gradually increment, as shown in Figure 3, barrier did not cause localization coverage reduction. While because of "extrusion" of barrier, the ratio of beacon nodes in such deployment area is higher than the same size of deployment area without barrier under the circumstance that the quantity of beacon nodes is equivalent. Ratio of beacon nodes increases; hence, localization accuracy in most of the area is relatively higher as shown in Figure 4. As shown in Figures 3 and 4, LE-WLS and LE-IILA similarly did not consider multicollinearity between original and newly added beacon nodes; as a result, localization errors of nodes in some areas are large and then affect the whole performance of localization.

Figure 5 describes curve of average ALE of repeated experiments by three localization methods varying with the quantity of beacon nodes in regular deployment scenes. It is easily to find ALE of LE-WLS fluctuating as the strongest, and accuracy is the lowest; LE-IILA is in second place, while LE-FWLS-CCR proposed in this paper is most accurate. This is because original and newly added beacon nodes are being most possibly collinear, furthermore, LE-WLS and LE-IILA algorithms did not consider this and noise was not removed completely, especially LE-WLS method that only considered heteroscedasticity but did not take noise escalation into account; consequently ALE curve waves greatly, sometimes; ALE is approximate to 180%; LE-IILA method only ideally considered noise escalation but did not take multicollinearity into account and so obtain better localization results than LE-WLS method, but the localization results are still not stable, and sometimes ALE is greater than 100%; therefore, the effectiveness of localization by LE-IILA is hard to meet actual needs; the method proposed in this paper considered multiple factors that affect accuracy in incremental localization process and obtained fairly stable localization results and significantly higher accuracy than other incremental localization methods.

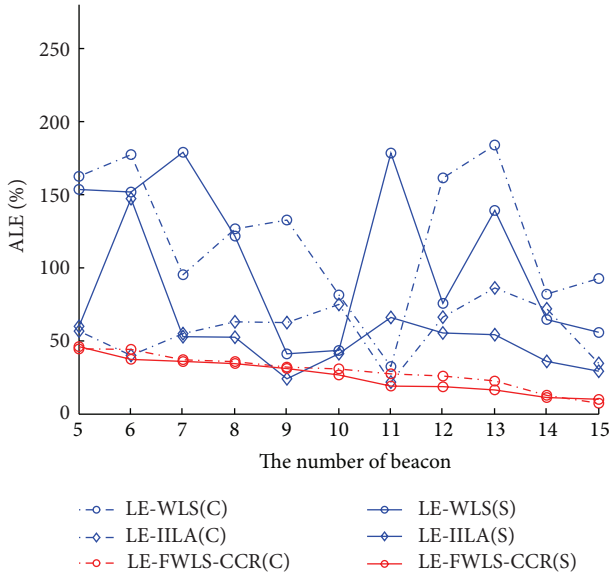



Figure 8: Average localization error of random deployment in square area.

*4.1.2. Random Deployment.* Random deployment is more close to actual situation. The experiments in this scene are used mainly is to discuss whether this algorithm is proper for various actual situations or not. In the same way, experiments about random deployment are classified into two groups, in C-shape area with barrier and in square area without barrier. In this group of experiments, there are 200 nodes randomly deployed in a 300 m × 300 m monitoring area, similar with regular deployment, to compare LE-FWLS-CCR with LE-WLS and LE-IILA in order to evaluate changes of ALE with quantity of beacon nodes by two algorithms. In the scene with barrier, a 150 m × 90 m object was placed in deployment area to artificially lead to ineffective communication of nodes in this area. In these experiments, select 5–15 nodes as beacon nodes provided that their location of information is known.

Figure 6 shows final localization result of nodes of certain deployment in square area under the circumstance that there are 10 beacon nodes. Figure 6(b) shows localization results of LE-WLS method, with weights being reciprocals of variance of error term that is optimal theoretically, ALE = 41%; Figure 6(c) shows LE-IILA method proposed by Ji, ALE = 18.5%; Figure 6(d) shows LE-FWLS-CRR proposed in this paper, ALE = 15.2%. From this figure, localization results of unknown nodes surrounding the area crowded with original beacon nodes are better, but with increasing incremental series, effectiveness of localization by LE-WLS is worse; LE-IILA is better than LE-WLS because the former considered error escalation, but in some areas, estimation results of unknown nodes are still far from actual values, for the reason that it did not consider multicollinearity between newly added beacon nodes and original beacon nodes. The results of LE-FWLS-CRR proposed in this paper are still stable and have higher accuracy than the previous two methods.

Figure 7 shows localization result of certain random deployment in C-shape area with barrier. Figure 7(a) shows nodes distribution diagram of certain localization. Figures (b),(c), and (d) show localization result of LE-WLS, LE-IILA, and LE-FWLS-CCR, respectively, and final ALE of various localization in Figure 7 is 38.6%, 15.6%, and 13.1%, respectively. The figure shows very obvious trace of incremental localization, that is, anticlockwise step-by-step location estimation. Due to existence of barrier, same quantity of beacon nodes accounts for a higher proportion in unit area than random deployment in square area; therefore, if the incremental series is lower, three methods' estimation of accuracy for unknown nodes is high, but with incremental series increasing, advantages and disadvantages of three methods will appear. LE-WLS did not consider error control and multicollinearity, so localization error for higher series is large; although LE-IILA considered error escalation to a certain extent, it did not consider multicollinearity, the localization results were improved but are still poor in some areas. The algorithm proposed in this paper obtained similar localization results with the previous three scenes, and its localization accuracy is still stable and excellent.

Figure 8 describes curve of average ALE of repeated experiments by three localization methods varying with quantity of beacon nodes in random deployment scenes. Curves of LE-WLS and LE-IILA methods still wave ups and downs, because randomness of random deployment is far greater than that of regular deployment, and as a result, maximum ALE of LE-WLS and LE-IILA methods is even approximate to 180%; however, the method proposed in this paper still obtains stable results. Owing to full considerations on adverse factors in localization process, the characteristics of random deployment did not reduce localization accuracy greatly but improved it.

*4.2. Simulation Experiment Based on Actually Measured Data.* This paper uses actually measured data set provided by Neal Patwari of Utah State University. The experiment was arranged in a standard office area that is a 12 m × 14 m rectangle. There are 44 nodes (in which 4 nodes act as beacon nodes) deployed in it; the communications among nodes adopt direct sequence spread spectrum (DS-SS), and the center frequency of deployment nodes is 2.4 GHz. This paper uses these data and enlarges effective communication radius of nodes to compare LE-WLS, LE-IILA, and LE-FWLS-CCR methods proposed in this paper. Table 1 shows that localization results of LE-FWLS-CCR for four different communication radiuses are better than those of other two localization algorithms. The details are shown in Table 1

Figure 9 shows the localization results of three algorithms under the circumstance that communication radius is 7 m, in which Figure 9(a) shows nodes deployment; Figure 9(b) is localization result of LE-WLS method, ALE = 81.8%; Figure 9(c) is localization result of LE-IILA method, ALE = 39.6%; Figure 9(d) shows localization result of LE-FWLS-CCR method proposed in this paper, ALE = 19.6%. From Figure 9, it can be seen that only localization results of unknown nodes near original beacon nodes are relatively satisfactory in Figure 9(b); in Figure 9(c), localization results of unknown nodes far from original beacon nodes are

Table 1: Comparisons of average localization errors based on actual RSSI measurement data.

| Wireless communication radius (m) | LE-WLS ALE | LE-IILA ALE | LE-FWLS-CCR ALE |
|---|---|---|---|
| 6.5 | 99.1% | 46.6% | 20.8% |
| 7 | 81.8% | 39.6% | 19.6% |
| 7.5 | 70.23% | 41.3% | 18.4% |
| 8 | 83.51% | 49.12% | 15.51% |

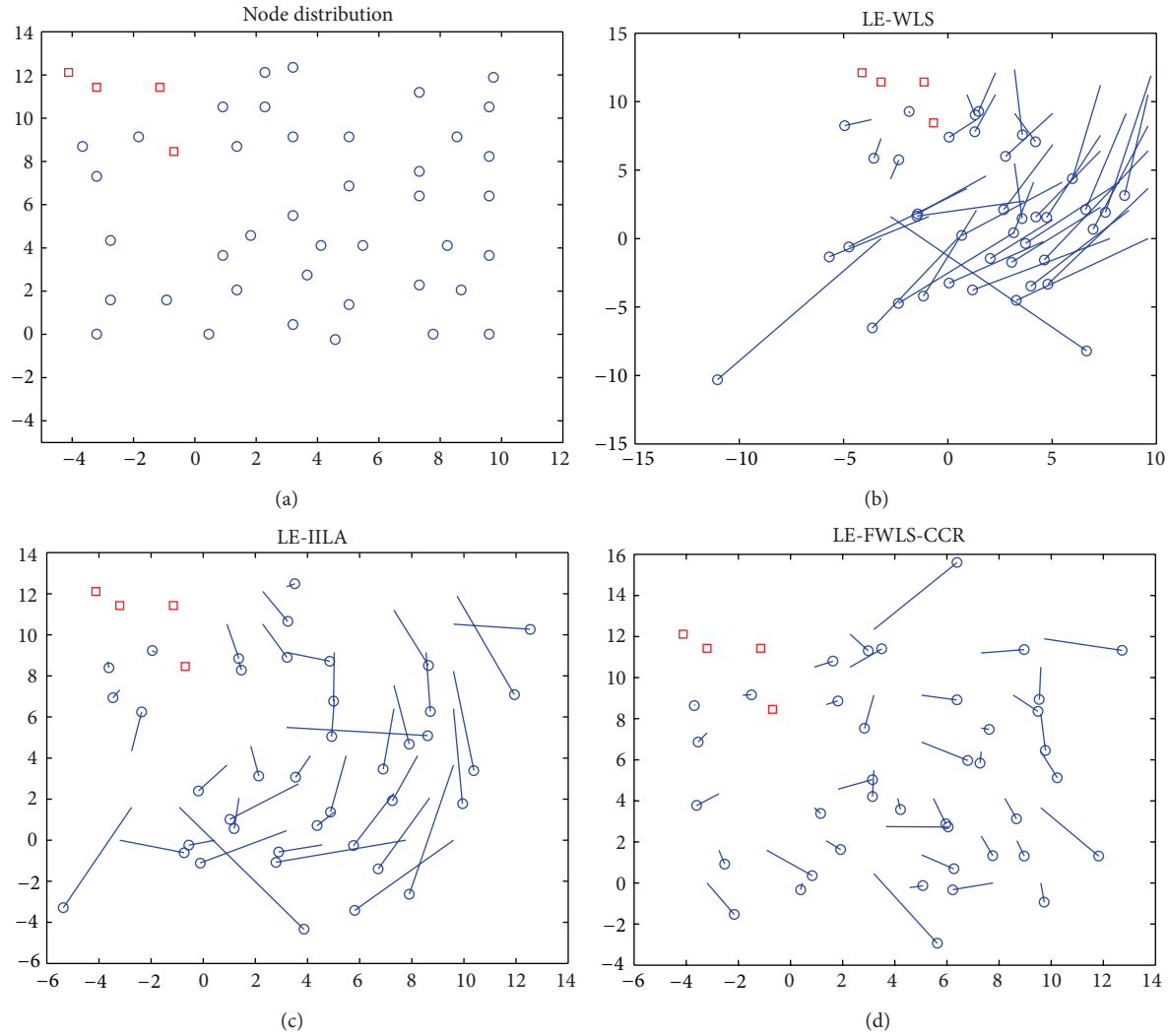



Figure 9: Localization results of actually measured data.

better than those in Figure 9(b); while localization result in Figure 9(d) is the best in the three.

## 5. Conclusion

This paper combines FWLS method and CCR method in a localization process that uses FLWS method to solve problems in location estimation caused by heteroscedasticity and uses dimensionality reduction algorithm CCR in multivariate analysis to deal with topology between original and newly added beacon nodes and error accumulation problems. The results of many groups of experiments indicate that the method proposed in this paper can effectively resolve heteroscedasticity, accumulative error, and multicollinearity problems, and its localization results are stable and have higher accuracy than previous incremental localization methods.

## Acknowledgments

The paper is sponsored by Natural Science Foundation of China (61005008); Provincial University Natural Science Research Foundation of Jiangsu Education Department

(11KJD510002, 12KJD510006); Natural Science Foundation of Jiangsu (BK2012082).

## References


[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] G. Cabri, L. Leonardi, M. Mamei, and F. Zambonelli, "Location-dependent services for mobile users," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 33, no. 6, pp. 667–681, 2003.

[3] J. Zheng and A. Jamalipour, *Wireless Sensor Networks a Networking Perspective*, John Whiley and Sons, New York, NY, USA, 2009.

[4] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller, "Mobile-assisted localization in wireless sensor networks," in *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '05)*, pp. 172–183, Miami, Fla, USA, March 2005.

[5] W. Ji and Z. Liu, "Accumulative error analysis of incremental node localization approach and its improvement in wireless sensor network," *Journal of Nanjing University of Science and Technology*, vol. 32, no. 4, pp. 496–501, 2008.

[6] F. Wang, L. Shi, and F. Ren, "Self-localization systems and algorithms for wireless sensor networks," *Journal of Software*, vol. 16, no. 5, pp. 857–868, 2005.

[7] C. Lim, P. K. Sen, and S. D. Peddada, "Accounting for uncertainty in heteroscedasticity in nonlinear regression," *Journal of Statistical Planning and Inference*, vol. 142, no. 5, pp. 1047–1062, 2012.

[8] F. Cribari-Neto and W. B. da Silva, "A new heteroskedasticity-consistent covariance matrix estimator for the linear regression model," *AStA Advances in Statistical Analysis*, vol. 95, no. 2, pp. 129–146, 2011.

[9] C. Meesookho, U. Mitra, and S. Narayanan, "On energy-based acoustic source localization for sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 365–377, 2008.

[10] W. Xiong, M. Tang, and B. Xu, "Incremental node localization approach and its improvement in wireless sensor network," *Chinese Journal of Sensors and Actuators*, vol. 24, no. 4, pp. 576–580, 2011.

[11] X. Yan, H. Qian, and J. Yu, "A localization method based on principal component analysis," *Journal of Computational Information Systems*, vol. 8, no. 22, pp. 9425–9432, 2012.

[12] H. Hyötyniemi, "Multivariate Regression Techniques and tools," http://autsys.aalto.fi/en/Publications/25041, 2013.

[13] C. Heij, P. de Boer, P. H. Franses, T. Kloek, and H. K. van Dijk, *Econometric Methods with Applications in Business and Economics*, Oxford University Press, Oxford, UK, 2004.

[14] B. Abraham and G. Merola, "Dimensionality reduction approach to multivariate prediction," *Computational Statistics and Data Analysis*, vol. 48, no. 1, pp. 5–16, 2005.

[15] M. Reiter, "Enhanced Multiple Output Regression based on Canonical Correlation Analysis with Applications in Computer Vision," 2010.

[16] L. Sun, S. Ji, and J. Ye, "Canonical correlation analysis for multi-label classification: a least-squares formulation, extensions, and analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 194–200, 2011.

[17] A. Schick, "Weighted least squares estimation with missing responses: an empirical likelihood approach," *Electronic Journal of Statistics*, vol. 7, pp. 932–945, 2013.

[18] S. Bidwell, M. E. Hassell, and C. R. Westphal, "A weighted least squares finite element method for elliptic problems with degenerate and singular coefficients," *Mathematics of Computation*, vol. 82, no. 282, pp. 673–688, 2013.

[19] I. Jolliffe, *Principle Component Analysis*, Springer, New York, NY, USA, 2nd edition, 2002.

[20] N. B. Priyantha, *The Crieket Indoor Location System*, Massaehusetts Institute of Technology, Cambridge, Mass, USA, 2005.

[21] "Measured Channel Impulse Response Data Set," http://,2013 span.ece.utah.edu/pmwiki/pmwiki.php?n=Main,2013 .MeasuredCIRDataSet,2013.