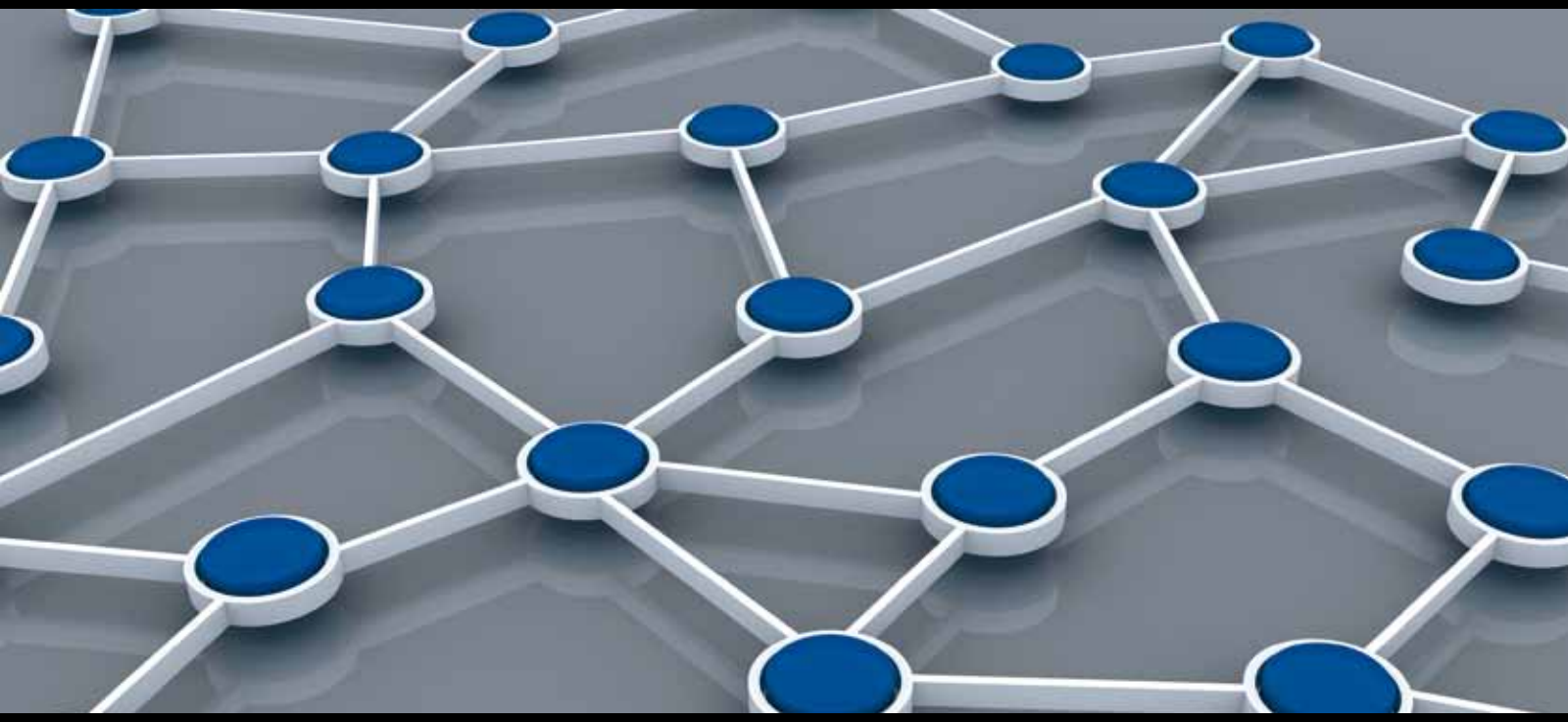
International Journal of Distributed Sensor Networks

Algorithmic Methods in Wireless Sensor Network

Guest Editors: Haigang Gong, Mei Yang, Wenzhong Li, and Nianbo Liu



Algorithmic Methods in Wireless Sensor Network

International Journal of Distributed Sensor Networks

Algorithmic Methods in Wireless Sensor Network

Guest Editors: Haigang Gong, Mei Yang, Wenzhong Li, and Nianbo Liu

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 148469, 2 pages
http://dx.doi.org/10.1155/2013/148469

Editorial
Algorithmic Methods in Wireless Sensor Network

Haigang Gong,[1] Mei Yang,[2] Wenzhong Li,[3] and Nianbo Liu[1]

[1] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China*
[2] *Department of Electrical and Computer Engineering, University of Nevada-Las Vegas, Las Vegas, NV 89119, USA*
[3] *Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China*

Correspondence should be addressed to Haigang Gong; hggong@uestc.edu.cn

Received 25 June 2013; Accepted 25 June 2013

Wireless sensor network (WSN) is characterized by dense node deployment, unreliable sensor, frequent topology change, and severe power, computation, and memory constraints. These unique characteristics pose considerable challenges on the design of large-scale WSN. The problems in different scenarios require different analytical models as well as algorithms to achieve optimal performance. This special issue provides some researches to resolve the issues in sensor networks such as traditional wireless sensor network, delay tolerant mobile sensor network, and vehicular sensor network.

The paper "*A cluster-based consensus algorithm in a wireless sensor network*" proposes an average connectivity degree cluster (ACDC) scheme gossip algorithm to improve the convergence speed and the accuracy of the consensus. A utility function is developed based on two parameters, iteration and relative error, to help the network designers make an optimal decision based on their requirements. An irregular sensor model which is based on the degree of irregular (DOI) radius is also introduced to evaluate the robustness of the algorithm.

The paper "*nonparametric bootstrap-based multihop localization algorithm for large-scale wireless sensor networks in complex environments*" presents a nonparametric bootstrap multihop localization algorithm for large-scale wireless sensor networks (WSNs) in complex environments. Authors integrate the interval analysis method with bootstrap approach for ordinary nodes localization. To reduce the computational complexity, boxes approach is utilized to approximate the irregular intersections.

The paper "*An energy distribution and optimization algorithm in wireless sensor networks for maritime search and rescue*" proposes a new method of maritime search and rescue based on wireless sensor networks. An energy dynamic distribution and optimization (EDDO) algorithm is presented to solve the problems of dynamic adaptability and life cycle limitation at sea.

The paper "*A hybrid energy- and time-driven cluster head rotation strategy for distributed wireless sensor networks*" proposes a hybrid cluster head rotation strategy which combines the advantages of both energy-driven and time-driven cluster head rotation strategies. In the hybrid rotation strategy, the time-driven strategy or energy-driven strategy will be selected according to the residual energy.

The paper "*An efficient data evacuation strategy for sensor networks in postdisaster applications*" introduces data evacuation (DE), an original idea that takes advantage of the survival time of the WSN, that is, the gap from the time when the disaster hits and the time when the WSN is paralyzed, to transmit critical data to sensor nodes in the safe zone in order to preserve "the last snapshot" of the whole network.

The paper "*Study on routing protocols for delay tolerant mobile networks*" gives a tutorial to routing protocols for delay tolerant mobile networks and investigates the state-of-the-art routing protocols for DTMNs. Some research issues are also discussed.

The paper "*Power control in distributed wireless sensor networks based on noncooperative game theory*" presents a game theoretic method to adaptively maintain the energy efficiency in distributed wireless sensor networks. The utility function was formulated under a proposed noncooperative framework, and then the existence of Nash Equilibrium (NE) has been proved to guarantee system stability. To pursue NE, an NPC algorithm was proposed to regulate heterogeneous nodes with various communication demands given the definition of urgency level.

The paper "*IPARK*: *location-aware-based intelligent parking guidance over infrastructureless VANETs*" taps into the unused resources offered by parked vehicles to perform parking guidance. In IPARK, the cluster formed by parked vehicles generates the parking lot map automatically, monitors the occupancy status of each parking space in real time, and provides assistance for vehicles searching for parking spaces.

The paper "*You take care of the drive*, *i take care of the rule*: *a traffic-rule awareness system using vehicular sensors and mobile phones*" proposes a novel traffic-rule awareness system using vehicular sensors and mobile phones. It translates traffic rules into combinations of vehicular sensors, GPS device, and Geography Information System (GIS); the system can tell whether a driver violates the traffic rules and can help him to amend his driving behavior immediately.

The paper "*Do not stuck at corners*: *a data delivery algorithm at corners in vehicular sensor networks*" proposes a data delivery algorithm called distribution-based data delivery to handle the corner problem with the help of some vehicular sensors like accelerometer, in which the big amounts of data are stuck at corners or crossroads and are transmitted back and forth with very few data packets being delivered.

The paper "*Distributed intrusion detection of byzantine attacks in wireless networks with random linear network coding*" develops a distributed algorithm to effectively detect, locate, and isolate the Byzantine attackers in a wireless ad hoc network with random linear network coding (RLNC).

The paper "*Public-transportation-assisted data delivery scheme in vehicular delay tolerant networks*" presents a destination-gathering-based driving path prediction method for taxis, which can make taxis' driving paths prescient in the initial stage of carrying passengers every time. Then a novel public-transportation-assisted data delivery (PTDD) scheme is proposed to improve the performance of data delivery of Vehicular Delay Tolerant Networks (VDTNs).

The paper "*Ant-based transmission range assignment scheme for energy hole problem in wireless sensor networks*" investigates the problem of uneven energy consumption in large-scale many-to-one sensor networks (modeled as concentric coronas) with constant data reporting. In view of the effectiveness of ant colony algorithms in solving combinatorial optimization problems, an ant-based heuristic algorithm (ASTRL) is proposed to address the optimal transmission range assignment for the goal of achieving life maximization of sensor networks.

The paper "*Application-oriented fault detection and recovery algorithm for wireless sensor and actor networks*" proposes an application-oriented fault detection and recovery (AFDR) algorithm, a novel distributed algorithm to reestablish connectivity. AFDR algorithm identifies critical actors and designates backups for them. A backup actor detects the critical node failure and initiates a recovery process via moving to the optimal position.

Haigang Gong
Mei Yang
Wenzhong Li
Nianbo Liu

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 923426, 9 pages
http://dx.doi.org/10.1155/2013/923426

Research Article

Nonparametric Bootstrap-Based Multihop Localization Algorithm for Large-Scale Wireless Sensor Networks in Complex Environments

Yongji Ren,[1,2] Ning Yu,[1] Xiao Wang,[1] Ligong Li,[1] and Jiangwen Wan[1]

[1] *School of Instrumentation Science and Opto-Electronics Engineering, Beijing University of Aeronautics and Astronautics (Beihang University), Beijing 100191, China*
[2] *Department of Command, Naval Aeronautical and Astronautical University, Yantai 264001, China*

Correspondence should be addressed to Yongji Ren; renyongji1006@sina.com

Received 8 November 2012; Revised 20 March 2013; Accepted 27 March 2013

Academic Editor: Wenzhong Li

This paper presents a nonparametric bootstrap multihop localization algorithm for large-scale wireless sensor networks (WSNs) in complex environments. Unlike most of the existing schemes, this work is based on the consideration that it is not feasible to obtain a lot of available distance measurements sample for estimation and to get exact noise distributions or enough prior information for conventional statistical methods, which is a situation commonly encountered in complex environments practically. For the first time, we introduce a nonparametric bootstrap method into multihop localization to build confidence intervals for multihop distance estimation, which can eliminate the risk of small sample size and unknown distribution. On this basis, we integrate the interval analysis method with bootstrap approach for ordinary nodes localization. To reduce the computational complexity, boxes approach is utilized to approximate the irregular intersections. Simulation results show that our proposed scheme is less affected by the variation of unknown distributions and indicate that our method can achieve high localization coverage with relatively small average localization error in large-scale WSNs, especially in sparse and complex network with smaller connectivity and anchor percentage.

1. Introduction

In the last few years, there has been a rapidly growing interest in extensive monitoring applications of wireless sensor networks (WSNs). One important reason is that, as opposed to traditional solutions, WSNs can be rapidly and randomly deployed in a large-scale monitoring region by plane or unmanned aerial vehicle (UAV), of which the environments of monitoring region are always complex even inaccessible.

As new technologies, WSNs can provide the means for long-term, all-weather, real-time, accurate, and extensive monitoring unattended, which brought us a convenient way to sense and monitor the complex environments. It is considered as an ideal system for this type of extensive environment monitoring and can fulfill the needs of various practical applications, for example, marine surveillance, ocean scientific exploration, and commercial exploitation.

For most WSNs applications, localization service is an indispensable part and an essential task. The location of the sensor nodes should be determined for meaningful interpretation of the sensed information. In recent years, a certain amount of research work has been conducted in this interesting research area, and a comprehensive survey is provided in [1–3] and the references therein. Some of the solutions are mainly designed for small-scale networks, and although this is also an interesting field, we will not contribute to the region in this paper; instead, we focus on the large-scale localization context.

As we know, out of the consideration of economic rationality, generally, it is too hard to deploy the sensor nodes in a large-scale area with relatively high density and large percentage of anchor nodes. If it is used for monitoring ocean environment, the network will be deployed sparsely and only a minority of anchor nodes can be employed for localization. In

these cases, it will be harder for ordinary nodes to communicate with enough anchor nodes directly so as to measure distances, which are necessary for localization.

To solve this problem, three types of localization schemes are proposed for large-scale localization, namely, mobile anchor assisted localization algorithms [4–6], recursive algorithms [7–9], and multihop algorithms [10–12], respectively. Hereinto, mobile anchor assisted localization methods commonly employ some expensive mobile equipment such as Autonomous Underwater Vehicle (AUV) which can roam across the monitoring region to assist localization. In recursive algorithms, some ordinary nodes that have been localized become secondary anchor nodes and broadcast coordinates to assist other nodes in estimating their locations. The localization process will inevitably suffer from the delay and adverse effects of error propagation and accumulation. Whereas for multihop localization, the ordinary node can infer the distances to its nonneighboring anchor nodes by approximating the length of the shortest path to the Euclidean distance, so as to get enough anchor nodes with known distances for localization. They not only can provide better real-time performance but also have no additional equipment requirement. In comparison, the characteristics of multihop methods are more suitable for large-scale WSNs localization. Therefore, multihop localization method has received more and more attention in recent years.

Certainly there are still many challenges for the multihop localization, especially in the complex environments. Here we define the complex environments as multiple complex terrain conditions (e.g., forest, rocks, marsh, underwater, etc.), random and sparse sensor node distribution, irregular radio propagation pattern, various unknown ambient noises, and multiple anisotropic network situations (e.g., H-type network, C-type network, etc.). There is no doubt that these adverse factors will seriously affect the accuracy of the multihop distance estimation which is the basis of the accurate sensor location estimation. To address this problem, most of the existing schemes have adopted conventional statistical methods and considered distance measurements affected by normal distribution noise and then determined the localization uncertainties through Monte Carlo analysis or nonlinear transformation techniques. For example, in [13], a dynamic localization scheme, the Monte Carlo localization (MCL), has been proposed. In [14], based on the discovery that the multihop distance errors obey normal distribution with various biases, a multihop localization algorithm that incorporates the distance estimation bias has been presented. But for the previous methods, only when the measurements sample size is large and the precondition that the errors obey normal distribution is satisfied, they could perform well in online localization.

However, for the large-scale WSNs in complex environment, it is too hard to get exact distribution characteristics of distance measurements and enough a priori information which is necessary for conventional statistical estimation method, and it is impossible to guarantee repeatable ranging conditions for reproducible measurements so as to obtain a large number of available measurements sample for Monte Carlo analysis or other similar methods. Hence, there is a need to develop a novel algorithm for producing coordinates through multihop localization approach as only very few measurements are available. And it is better that this methodology only needs less prior information and even no statistical assumptions on disturbances.

On the basis of analysis on the challenges of complex environments and limitations of existing studies for large-scale localization, a different approach is proposed and investigated, that is, a multihop localization method based on nonparametric bootstrap for large-scale WSNs in complex environments.

The bootstrap is a powerful technique for assessing the accuracy of a parameter estimator in situations where conventional techniques are not valid. The nonparametric bootstrap method was originally introduced by Efron in [15] and used to address CI estimate for statistics based on independent and identically distributed (i.i.d.) random variable from some unknown distribution $F(\mu, \sigma)$ [16]. Compared with conventional techniques, the significant advantages of the nonparametric bootstrap method are that it does not require any modeling or assumptions on the data and it is more suitable for small sample estimation. In this paper, it can be utilized to build confidence intervals effectively for multihop distance estimation online. On this basis, with interval methods, we can compute the bounds of the possible solutions that correspond to measured quantities and determine the guaranteed regions that involve the correct solution.

This paper is organized as follows. Section 2 introduces the related works and challenges for large-scale WSNs localization. Section 3 presents in details the nonparametric bootstrap and interval analysis method. Section 4 evaluates the performance of our algorithm through experiments, and Section 5 concludes this paper.

2. Related Works

2.1. Mobile Anchor Assisted Localization Algorithms. To solve the localization problems of large-scale WSNs, in [4], the authors employ a single mobile beacon to aid in localization. The sensor locations are maintained as probability distributions that are sequentially updated using Monte Carlo sampling as the mobile anchor node moves over the monitoring area. This method relies on the more powerful anchor to perform the calculation and relieves most localization tasks from the less powerful ordinary nodes.

The authors in [5] proposed a range-free localization scheme for WSNs using mobile anchor nodes equipped with four directional antennas. Therein, each mobile anchor node can determine its position via Global Position System (GPS), and then it broadcasts its coordinates as it moves through the region. The ordinary nodes detect these messages and utilize a simple processing scheme to determine their own coordinates based on those of the anchors.

In the AUV-aided localization scheme proposed in [6], AUVs keep roaming across the underwater sensor field to aid in localization. The AUVs can get coordinates from GPS while floating periodically and dive into a fixed depth and navigate through a predefined route using compass and dead-reckoning. They can estimate the distance between the AUV

and ordinary nodes through a request/response message pair exchange which contains its coordinates. Therefore, the ordinary nodes can be localized after the message exchange from three different noncoplanar AUV locations.

Although these protocols exploit the mobility of the mobile equipment to overcome the lack of adequate anchor nodes, the major drawbacks of these mobile anchor assisted localization algorithms are that the mobile machine is too expensive for WSNs, and the slow speed of AUV or other machine always introduces high localization delay. In addition, the movement of the anchor nodes will be severely restricted by the complex environments.

2.2. Recursive Localization Algorithms. Liu and Zhang [7] presented an error control mechanism for recursive localization based on the characteristics of node uncertainty and the active selection strategy of anchor nodes. The error control mechanism only utilizes local knowledge and can mitigate the effect of error propagation for both range and directional sensors to a certain extent.

Yu et al. proposed a two-stage localization approach in [8]. Firstly, localization process starts from the nodes with the largest numbers of neighbor anchors which have the priority. Then, the coordinates of all neighbor nodes are exploited to improve localization accuracy. During the procedure, a number of measures are also taken to ensure the reliability of each location estimate to avoid abnormal errors and reduce error propagation.

Vemula et al. [9] formulated the sensor localization from a probabilistic point of view and incorporated anchor position uncertainty to estimate the distribution of node coordinates, including iterative least squares and Bayesian methods, Monte Carlo importance sampling, and cost-based methods.

These schemes try to inhibit the propagation and accumulation of localization errors, but the high computational complexity and increased communication cost limit their application in practice. Additionally, this kind of method is also easy to cause the localization delay.

2.3. Multihop Localization Algorithms. DV-distance and DV-hop algorithms, as the origination of multihop localization schemes for WSNs, are proposed in [10]. In both algorithms, each anchor node broadcasts a message to its immediate neighbor nodes firstly. Then, the message is propagated in a controlled flood manner so that each ordinary node can estimate the lengths of shortest paths to anchor nodes. When the ordinary node obtains the estimates to enough anchor nodes, its position can be calculated.

Wang and Xiao [11] presented an improved multihop algorithm called i-Multihop which has higher computational complexity. At the beginning, the upper bound constraints are used to filter out the incorrect distance estimations and the estimated position is pinpointed to the intersection constrained by the correct distances. And then, the distance fitting is used to fit correct distance measurements, which makes the final estimated position not to be affected by the layout of anchor nodes.

The authors in [12] proposed three multihop localization schemes based on least squares and multilateration, namely, Taylor-LS, weighted Taylor based least squares, and constrained total least squares, respectively. Additionally, a generalized Cramér-Rao lower bound is developed to analyze the performance of multihop localization approaches.

As we can see, most of the previous schemes considered the measurement uncertainty as normal distribution even zero-mean normal distribution noise and employ conventional error-processing approaches, for example, nonlinear least squares estimator (NLSE), and so forth. However, in complex large-scale WSNs, it is practically too hard to fulfill the requirements which are necessary for these approaches, such as enough a priori information, exact distribution characteristics, and large sample size.

Specifically, for the ordinary nodes, there are not so many anchors that can communicate with them to accomplish distance measurement. The ranging process also cannot be performed again and again due to the limited energy of sensor nodes. Even when cost is not a main concern, it is also impossible to guarantee repeatable conditions for reproducible distance measurement. Hence, it is not practically feasible to obtain a lot of available distance measurements sample for estimation. Moreover, in complex environments, for example, underwater environments, the ranging process will suffer from various non-Gaussian noises, for example, the multipath and waveguide effects, surface scattering, and so forth. Obviously, for the previous noises, it is too hard to get exact distribution characteristics and enough prior information which are necessary for statistical methods. It is also unreasonable to consider that the uncertainties always obey normal distribution. Hence, in this paper, the bootstrap localization method is proposed for large-scale WSNs in complex environments.

3. Nonparametric Bootstrap-Based Localization Algorithm for Large-Scale WSNs

To solve the problems mentioned earlier, in this paper, we propose a novel multihop localization scheme which integrates a nonparametric bootstrap distance estimation method with an interval analysis approach. We firstly utilize the nonparametric bootstrap method to provide a confidence interval (CI) as the estimation for distance measurement. On that basis, interval analysis approach is employed to determine the feasible set and search the optimal estimation of coordinates.

3.1. Short Review of CI Estimate. As we know, CI is an interval about the measurement result within which the values that could reasonably be attributed to the measurand may be expected to lie with a given level of confidence. A common approach to address CI estimate based on the normal distribution can be described as follows. Let $X_1, X_2, \ldots, X_n$ be n i.i.d. Gaussian random variables with known σ from distance measurements, and suppose that we wish to find a confidence level $1 - \alpha$ interval for the mean θ. As we know, the random variable $(\widehat{\theta} - \theta)/(\sigma/\sqrt{N})$ obeys standard normal

distribution, and its probability density function (PDF) $f(x)$ can be computed by

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \tag{1}$$

and its cumulative distribution function (CDF) $F(z)$ can be computed by

$$F(z) = \int_{-\infty}^{z} f(x)\, dx. \tag{2}$$

For a given confidence level $1 - \alpha$, there exists constant $c > 0$ such that

$$\Pr\left\{-c < \frac{\widehat{\theta} - \theta}{\sigma/\sqrt{N}} < c\right\} = 1 - \alpha = 2F(c) - 1. \tag{3}$$

It follows that constant c satisfies $F(c) = 1 - (\alpha/2)$, and thus, $c = z^{\alpha} = F^{-1}(1 - (\alpha/2))$, where F^{-1} has been tabulated.

Equality (3) can be rewritten as

$$\Pr\left\{\widehat{\theta} - z^{\alpha}\frac{\sigma}{\sqrt{N}} < \theta < \widehat{\theta} + z^{\alpha}\frac{\sigma}{\sqrt{N}}\right\} = 1 - \alpha. \tag{4}$$

Interval

$$N_{\alpha} = \left[\widehat{\theta} - z^{\alpha}\frac{\sigma}{\sqrt{N}}, \widehat{\theta} + z^{\alpha}\frac{\sigma}{\sqrt{N}}\right] \tag{5}$$

is the CI corresponding to confidence level $1 - \alpha$.

However, σ must be practically estimated from the distance measurements, and this method performs well only in the case when n is large as per the central limit theorem. In the case where n is small ($n \leq 30$), this method could be invalid.

If sample size $n \leq 30$, the CI based on the Student's t-distribution is often utilized to replace the previous method. Under the circumstances, the CIs can be obtained as the following.

Let the sample variance S^2 be defined by

$$S^2 = \frac{\sum_{k=1}^{N}\left(X_k - \widehat{\theta}\right)^2}{N - 1}, \tag{6}$$

and the random variable $T = (\widehat{\theta} - \theta)/(S/\sqrt{N})$ obeys Student's t-distribution with $(N - 1)$ degrees of freedom. The PDF of T, which is denoted by $f_N(t)$, can be obtained by

$$f_N(t) = C(N)\left(1 + \frac{t^2}{N - 1}\right)^{-N/2}, \tag{7}$$

where $C(N)$ is a normalizing constant. Let $F_N(z)$ be the CDF of Student's t-distribution with $(N - 1)$ degrees of freedom and constant $c > 0$ such that

$$\Pr\left\{-c < \frac{\widehat{\theta} - \theta}{S/\sqrt{N}} < c\right\} = 1 - \alpha = 2F_N(c) - 1. \tag{8}$$

Then, constant c must satisfy $F_N(c) = 1 - (\alpha/2)$, and c can be obtained by $c = t_{N-1}^{\alpha} = F_N^{-1}(1 - (\alpha/2))$, where F_N^{-1} has been tabulated.

Equality (8) can be rewritten as

$$\Pr\left\{\widehat{\theta} - t_{N-1}^{\alpha}\frac{S}{\sqrt{N}} < \theta < \widehat{\theta} + t_{N-1}^{\alpha}\frac{S}{\sqrt{N}}\right\} = 1 - \alpha. \tag{9}$$

Interval

$$T_{\alpha} = \left[\widehat{\theta} - t_{N-1}^{\alpha}\frac{S}{\sqrt{N}}, \widehat{\theta} + t_{N-1}^{\alpha}\frac{S}{\sqrt{N}}\right] \tag{10}$$

is the CI corresponding to confidence level $1 - \alpha$. There into, both $\widehat{\theta}$ and S can be estimated from the distance measurements.

3.2. CI Based on Nonparametric Bootstrap. In most studies, the CI is commonly and directly obtained through the previous two methods, based on the hypothesis that the distance uncertainty obeys normal distribution and the sample size is sufficiently large. However, as we mentioned before, it is not the case for large-scale WSNs localization in complex environments. When improving accuracy in conventional methods is invalid, bootstrap method can be used to address the CI estimate issues. The CIs of distance measurements can be estimated by employing the nonparametric percentile bootstrap method as follows.

Let $\widehat{\theta}_{\alpha}^{*}$ represent the 100αth percentile of B bootstrap replications $\widehat{\theta}^{*}(1), \widehat{\theta}^{*}(2), \ldots, \widehat{\theta}^{*}(B)$. Percentile limit $\widehat{\theta}_{\text{lower}}$, $\widehat{\theta}_{\text{upper}}$ of intended coverage $1 - 2\alpha$, is directly obtained from the following percentiles:

$$\left[\widehat{\theta}_{\text{lower}}, \widehat{\theta}_{\text{upper}}\right] = \left[\widehat{\theta}_{\alpha}^{*}, \widehat{\theta}_{1-\alpha}^{*}\right]. \tag{11}$$

The nonparametric percentile bootstrap methodology to determine the upper and lower bounds on the CI for distance will be described in detail in the bootstrap distance estimation subsection.

3.3. Nonparametric Bootstrap-Based Distance Estimation Approach. The basic concept of bootstrap method is to produce a large number of independent bootstrap distance estimates by resampling the original distance estimate; that is, $X = (d_1, d_2, \ldots, d_n)$, which consists of n measurements at random from unknown probability distribution F.

A bootstrap resample $X^{*} = (d_1^{*}, d_2^{*}, \ldots, d_n^{*})$ is obtained as a random sample of size n randomly drawn with replacement from the original measurement set X. Sample X has true mean θ; drawing various other samples X from the distribution F, their means will present distribution $\widehat{\theta}$. The distribution of real estimated mean $\widehat{\theta}$ is approximated by the distribution of pseudoestimated mean $\widehat{\theta}^{*}$ from bootstrap resample X^{*}.

Let $X = (d_1, d_2, \ldots, d_n)$ represent the set of the corresponding distance measurements. Using the nonparametric percentile bootstrap method on this set, we generate B resamples; $X_j^{*} = (d_{1j}^{*}, \ldots, d_{nj}^{*})$, $j = 1, \ldots, B$. And then, we calculate the mean of all measurements in X_j^{*} to obtain $\widehat{\theta}_{X_j^{*}}^{*}$ given by

$$\widehat{\theta}_{X_j^{*}}^{*} = \frac{1}{N}\sum_{k=1}^{N} d_{k,j}^{*}, \tag{12}$$

where N is often smaller than 5 (which is a reasonable value and common situation in complex WSNs localization practically), and $j = 1, \ldots, B$. The mean of this distribution is given as

$$\widehat{\theta}^*_X = \frac{1}{B}\sum_{j=1}^{B}\widehat{\theta}^*_{X^*_j}. \tag{13}$$

We sort bootstrap estimates $\widehat{\theta}^*_{X^*_j}$ in an ascending order. Let the sorted distance be given by

$$\widehat{\theta}^*_{X^*_1} \le \widehat{\theta}^*_{X^*_2} \le \widehat{\theta}^*_{X^*_3} \le \cdots \le \widehat{\theta}^*_{X^*_{B-1}} \le \widehat{\theta}^*_{X^*_B}. \tag{14}$$

The desired $100 \cdot (1-\alpha)\%$ nonparametric CI for the distance is given by $(\widehat{\theta}^*_{X(Q_1)}, \widehat{\theta}^*_{X(Q_2)})$, where Q_1 is the integer part of $B\alpha/2$, $Q_2 = B - Q_1 + 1$, and $Q_3 = B/2$ is the integer part of $B/2$.

Furthermore, note that

$$\begin{aligned}
\widehat{\theta}^*_{X(Q_1)} &= \frac{1}{N}\sum_{k=1}^{N} d^*_{k,Q_1}, \\
\widehat{\theta}^*_{X(Q_2)} &= \frac{1}{N}\sum_{k=1}^{N} d^*_{k,Q_2}, \\
\widehat{\theta}^*_{X(Q_3)} &= \frac{1}{N}\sum_{k=1}^{N} d^*_{k,Q_3}.
\end{aligned} \tag{15}$$

Q_1, Q_2, and Q_3 are referred to as the upper, lower, and middle, respectively.

Finally, we can obtain the interval

$$\left[\widehat{\theta}_{\text{lower}}, \widehat{\theta}_{\text{upper}}\right] = \left[\widehat{\theta}^*_{X(Q_1)}, \widehat{\theta}^*_{X(Q_2)}\right] \tag{16}$$

as the CI of distance corresponding to confidence level $1-\alpha$. Median point $\widehat{\theta}^*_{X(Q_3)} \cong \widehat{\theta}^*_X$ is almost identical to the mean. The upper, middle, and lower points are used by the algorithm to estimate the CIs for the distance, respectively. For $\alpha = 0.05$ and $B = 1000$, we can get $Q_1 = 25$, $Q_2 = 976$, and $Q_3 = 500$. And the CI corresponding to confidence level 95% can be described as $[\widehat{\theta}^*_{X(25)}, \widehat{\theta}^*_{X(976)}]$.

3.4. Determination of Coordinates Based on Interval Analysis Method. When we obtain the CI from a small sample set of distance measurements, the interval analysis method will be utilized to determine the positions of ordinary nodes.

Consider a network consisting of m anchors with known positions and n ordinary nodes which need to be localized. The location of node N_i can be described by $\mathbf{X}_i = [x_i, y_i, z_i]^T$. The CI of measured distance from the ordinary node N_a to one of its anchor nodes N_i

$$\left[\widehat{d}_{ai_{\text{lower}}}, \widehat{d}_{ai_{\text{upper}}}\right] \tag{17}$$

can be obtained through nonparametric bootstrap distance estimation approach, and the Euclidean distance can be defined as

$$D_{ai} = \|X_a - X_i\|_2. \tag{18}$$

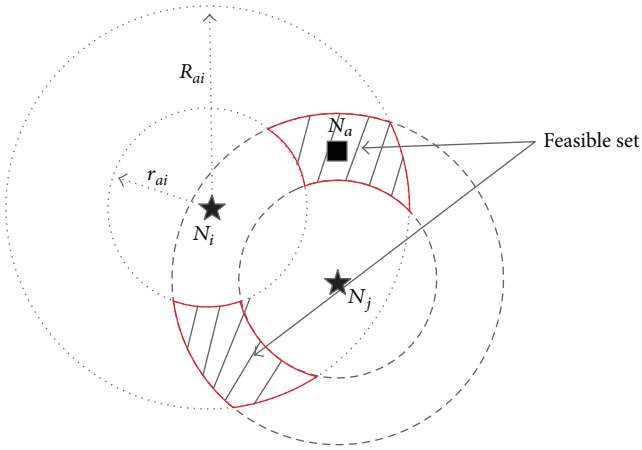


Figure 1: Feasible set of ordinary node N_a (top view).

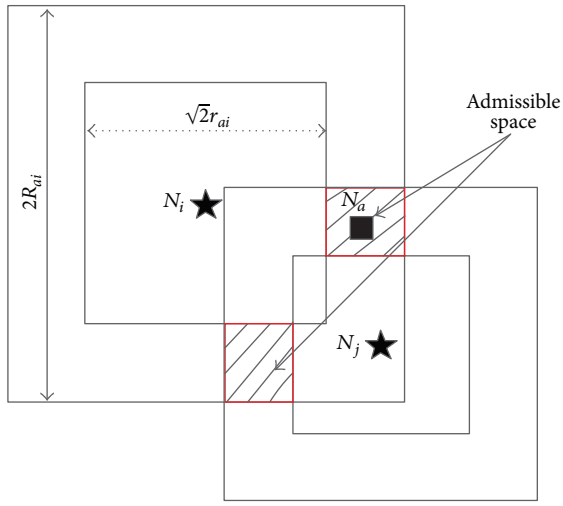


Figure 2: Approximate intersections (top view).

Obviously, the desired position estimate is guaranteed to be bounded by an admissible space which can be denoted as

$$S(x) = \bigcap_{i=1}^{k}\left\{\widehat{d}_{ai_{\text{lower}}} \le \|X_a - X_i\|_2 \le \widehat{d}_{ai_{\text{upper}}}\right\}. \tag{19}$$

$S(x)$ is the intersection of all the spherical caps of center X_i and of internal and external radii $r_{ai} = \widehat{d}_{ai_{\text{lower}}}$ and $R_{ai} = \widehat{d}_{ai_{\text{upper}}}$, respectively. Clearly, $S(x)$ has a complicated geometrical shape (see Figure 1).

To reduce the computational burden, boxes approach will be utilized to approximate the irregular intersection of all feasible solutions. For each anchor ordinary pair, their admissible space represents a region between two boxes of length $\sqrt{2}r_{ai}$ and $2R_{ai}$, respectively (see Figure 2).

As we know, interval analysis is usually used to model quantities that vary around a central value within certain bounds. With simple operations, the interval analysis allows to consistently deal with problems involving interval data. Based on the interval analysis theory, the admissible space

of N_a to N_i can be denoted by $\vartheta^I = [\vartheta^{I^-}, \vartheta^{I^+}]$, where ϑ^I is a closed and connected subset of $\mathbb{R}^3$, and ϑ^{I^-} and ϑ^{I^+} are the minimal and maximal bounds of ϑ^I, respectively. Set theory operations, such as intersection, can be applied to intervals. Consider two intervals ϑ^I_{a1} and ϑ^I_{a2}, and their intersection is always an interval which can be denoted as $\vartheta^I_{a1} \cap \vartheta^I_{a2} = [\{\vartheta \mid \vartheta \in \vartheta^I_{a1}, \vartheta \in \vartheta^I_{a2}\}]$. The intersection can be computed by

$$\vartheta^I_{a1} \cap \vartheta^I_{a2} = \left[\max\left\{\vartheta^{I\ -}_{a1}, \vartheta^{I\ -}_{a2}\right\}, \min\left\{\vartheta^{I\ +}_{a1}, \vartheta^{I\ +}_{a2}\right\}\right]. \quad (20)$$

Specifically, the area ϑ_{ai} encircled by the circumscribed and inscribed squares as shown in Figure 2, that is, N_a's admissible space to N_i, can be computed by:

$$\begin{aligned}\vartheta_{ai} &= [x_a - R_{ai}, x_a + R_{ai}] \times [y_a - R_{ai}, y_a + R_{ai}] \\ &\times [z_a - R_{ai}, z_a + R_{ai}] - \left[x_i - \frac{1}{2}\sqrt{2}r_{ai}, x_i + \frac{1}{2}\sqrt{2}r_{ai}\right] \\ &\times \left[y_i - \frac{1}{2}\sqrt{2}r_{ai}, y_i + \frac{1}{2}\sqrt{2}r_{ai}\right] \\ &\times \left[z_i - \frac{1}{2}\sqrt{2}r_{ai}, z_i + \frac{1}{2}\sqrt{2}r_{ai}\right].\end{aligned} \quad (21)$$

Using the same previous method, we can get N_a's another admissible space. Finally, the feasible set W, an interval vector set contains all the possible coordinates, can be obtained by simple enumeration of the intersection points among the two boxes.

Therefore, the feasible set $S(x)$ of ordinary node N_a can be rewritten as

$$\Omega_a = \bigcap_{i=1}^{k}\left\{\vartheta \in \vartheta^I : \vartheta^I_{ai} = \left[\vartheta^{I\ -}_{ai}, \vartheta^{I\ +}_{ai}\right]\right\}. \quad (22)$$

Clearly, the intersection of all subfeasible set ϑ^I_{ai} is a set of boxes. Regarding the coordinates of all subboxes' centers as samples of $\mathbf{X}_a$, we can get a sample set

$$\Omega = \{\Theta_1, \Theta_2, \ldots, \Theta_n\}, \quad (23)$$

and the centre of Θ_n can be found by $\vartheta^*_n = (\vartheta^- + \vartheta^+)/2$. The optimum point estimate, that is, the desired coordinates, can be obtained by

$$\begin{aligned}\widehat{X}_a = \arg\min_{X_a} \quad & \sum_{i}^{k}\left(\left\|\vartheta^*_n - X_i\right\|_2 - d_{ai}\right)^2 \\ \text{subject to} \quad & X_a \in S(x).\end{aligned} \quad (24)$$

Finally, the coordinates can be given as

$$\widehat{X}_a = \left[\widehat{\vartheta}^*_n(1), \widehat{\vartheta}^*_n(2), \widehat{\vartheta}^*_n(3)\right]^T. \quad (25)$$

4. Performance Evaluation

In this section, we conduct extensive simulations to evaluate the performance of our proposed localization scheme, called Nonparametric Bootstrap Based Multihop Localization Algorithm (NBMLA). All simulations are run in Matlab R2012a. To reduce the influence of outliers, we take the average of 100 simulation runs as the final data points.

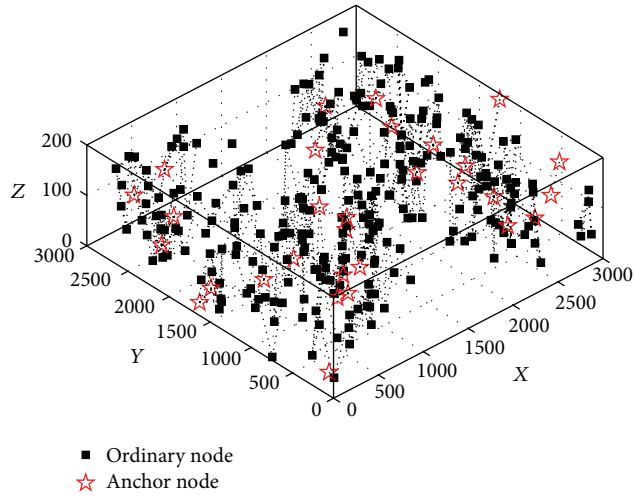


Figure 3: Topology of large-scale WSNs.

4.1. Simulation Settings. In our simulation experiments, 400 nodes with adjustable transmission range R are randomly distributed in a large-scale three dimensional region with a size of 3000 m × 3000 m × 200 m (see Figure 3). We control the density and connectivity of the network by changing the transmission range while keeping the area of deployment the same. Different anchor percentages are also considered in our simulation. Besides our scheme, we also simulate an improved localization algorithm named Taylor-LS for comparison which is almost the same as [12].

We mainly consider three performance metrics: localization coverage, localization error, and time complexity. Localization coverage is defined as the ratio of the localized nodes to the total sensor nodes. Average localization error is the average distance between the estimated positions and the real positions of all ordinary nodes. Distribution boxplots of node localization errors show the average errors, median errors, and maximum outliers. The analysis of time complexity mainly focuses on the communication cost and computation complexity.

4.2. Localization Coverage. In this subsection, we analyze the performance of localization coverage with different network connectivity and anchor percentage, respectively. Figure 4 shows the performance of localization coverage with changing network connectivity while anchor percentage is 15%. We can observe that the localization coverage of our scheme increases monotonically with the connectivity ranging from 4 to 13. But when the connectivity reaches a high value, the coverage rate becomes relatively large and will not change much after that. For example, when the network connectivity is 10, the localization coverage becomes 97% and then changes slower. The reason is that when the network connectivity reaches a certain point, most nodes can get enough anchor nodes by multihop approach and localize themselves.

As shown in Figure 5, the localization coverage of our proposed scheme increases with the anchor number when we change the anchor nodes from 20 (5% anchors) to 80 (20% anchors) with step 10, while keeping the network connectivity

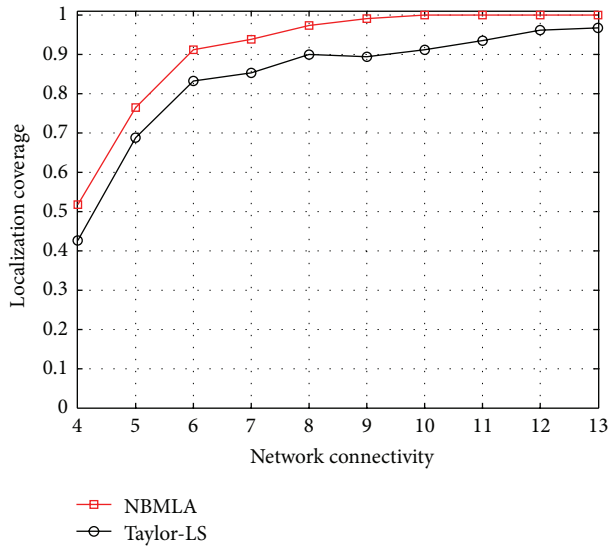


Figure 4: Localization coverage versus network connectivity.

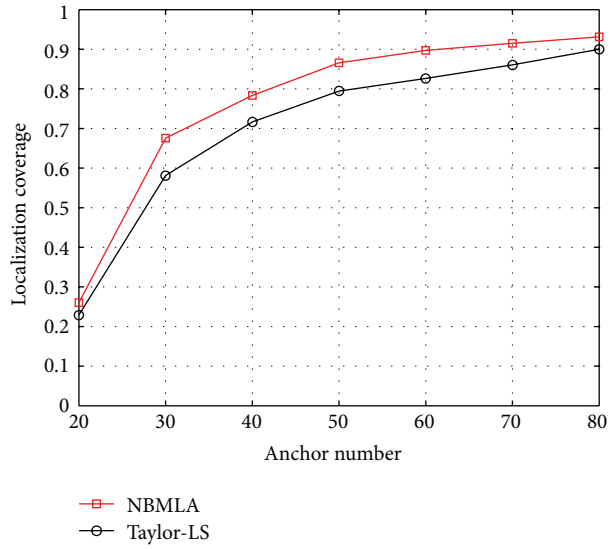


Figure 5: Localization coverage versus anchor number.

as 9. We can see that when the anchor number reaches a certain value, for example, 50 anchors, the coverage rate will not change much after that. We can also see that when the anchor rate is 15% and the connectivity is 9, our scheme can localize more than 95% nodes in the large-scale WSNs. Compared with the results of Taylor-LS, NBMLA can achieve better performance in terms of localization coverage, especially in the low anchor percentage situation.

4.3. Localization Error. As mentioned before, in complex environment the ranging process usually encountered some non-Gaussian noises which stem from multiple factors, for example, multipath effect, multihop distance estimation, and anisotropic network condition. This makes it harder for ordinary nodes to get exact distribution characteristics of the distance measurement noises. For comparison with classical

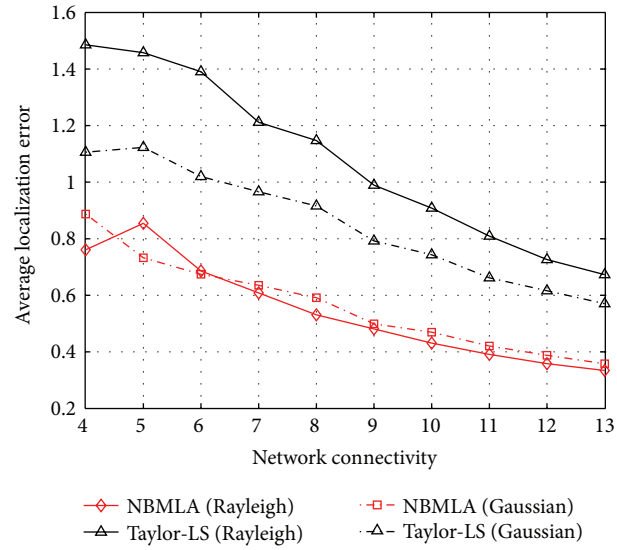


Figure 6: Average localization error versus connectivity.

approaches and showing the performance of our algorithm in non-Gaussian noises context, the measurement noise has been assumed to follow Rayleigh distribution, with three percent of real distances as the standard deviations. In other words, the knowledge of the error bound roughly equals nine percent of real distances. This is a reasonable assumption and can be satisfied by the existing distance measurement technologies when suffering multiple adverse factors.

Figure 6 plots the relationship between the average localization error and network connectivity when the anchor percentage is 15%. Besides the given Rayleigh distribution, we also simulate a normal distribution noise situation with mean value and standard deviation (10 m, 10 m) for comparison with the classical approaches relying on normal distributions hypotheses. We can observe that the average localization error of our scheme decreases significantly with the increase of network connectivity. It should be noted that our scheme can achieve relatively high localization accuracy even with low network connectivity. This indicates the good localization performance of our proposed scheme in sparse region. And it can be seen that our scheme outperforms the Taylor-LS method in both Gaussian and non-Gaussian cases. It indicates that our scheme is less affected by the variation of distributions.

In Figure 7, we vary the anchor percent, which is ranging from 5% to 20%, and get the accuracy comparisons with different anchors. For our scheme, with the number of anchor nodes varying from 20 to 80, the localization error decreases by 40%. However, the localization error of Taylor-LS decreases only by 20%, when the network connectivity is 9. This suggests that in sparse networks, our scheme can achieve higher localization accuracy just by increasing a little number of anchor nodes.

4.4. Discussions. Finally, we analyze the time complexity of the schemes which mainly focuses on the communication cost and computation complexity. For a network with M

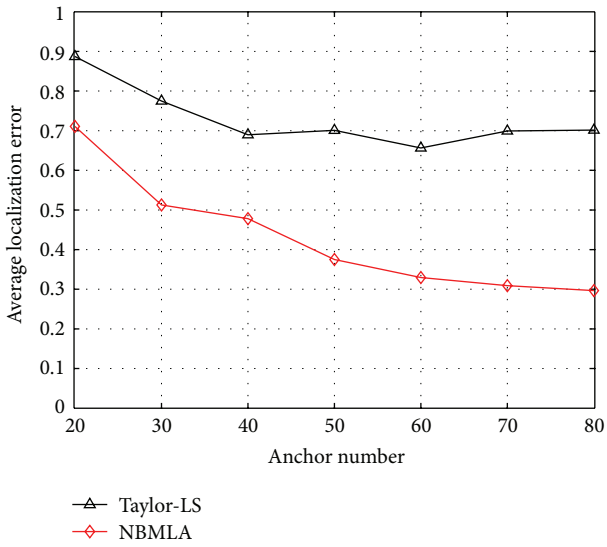


Figure 7: Average localization error versus anchor number.

anchor nodes and N ordinary nodes, the overall communication complexity of our scheme is $O(MN)$. Firstly, NBMLA needs the M times flooding initiated by M anchors to make the ordinary nodes know their hop counts, whose overhead is $O(MN)$. Besides the flooding, our scheme requires each anchor to propagate its hop counts to three or four hops' neighboring nodes by a confined flooding. This also has the overhead of $O(MN)$. The communication complexity of Taylor-LS equals to that of NBMLA. But when the algorithms need more distance samples for parametric statistics or location refinement, our scheme can achieve much lower communication cost compared with Taylor-LS. This is due to the fact that NBMLA obtained lots of distance samples by bootstrap resampling approach rather than repeating communication with anchors which introduces a large communication cost for Taylor-LS. This characteristic is particularly important when the network is sparse. Of course, in this case, the computation complexity of our scheme will be larger than Taylor-LS, because the resampling process is necessary when the original distance sample size is very small. For example, if there are only 5 effective distance measurement samples, the NBMLA needs to resample at least B times (>200) for estimating parameter and constructing the CI. Thus, the computation complexity of NBMLA is $O(BN)$. Fortunately, in our scheme, the location estimate can be obtained by simple enumeration instead of complex matric arithmetic. And compared with the communication cost, the computation cost of our method is moderate and acceptable.

5. Conclusions

In this paper, we present a novel multihop localization algorithm which integrates the nonparametric bootstrap method with interval analysis approach for large-scale WSNs in complex environments. In the first phase, the nonparametric bootstrap method is utilized to build confidence interval of distance form a small sample set of available measurements with unknown noise distributions. On that basis, we adopt interval analysis approach to estimate the positions of ordinary nodes. The simulation results demonstrate that our algorithm can get high localization coverage and accuracy while resisting against the variation of unknown noise distributions. Compared with the traditional method, the proposed method can greatly improve the average localization accuracy. Further studies are being conducted to extend the method to mobile WSNs in complex environments.

Acknowledgments

The authors are grateful to the anonymous reviewers for their industrious work and insightful comments. This work is supported by the National Natural Science Foundation of China under Grant nos. 61001138 and 61201317.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, 2002.

[2] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on wireless sensor networks technology and evolution," *Sensors*, vol. 9, no. 9, pp. 6869–6896, 2009.

[3] M. Erol-Kantarci, H. T. Mouftah, and S. Oktug, "A survey of architectures and localization techniques for underwater acoustic sensor networks," *IEEE Communications Surveys & Tutorials Home*, vol. 13, pp. 487–502, 2011.

[4] R. Huang and G. V. Záruba, "Monte Carlo localization of wireless sensor networks with a single mobile beacon," *Wireless Networks*, vol. 15, no. 8, pp. 978–990, 2009.

[5] O. Chia-Ho, "A localization scheme for wireless sensor networks using mobile anchors with directional antennas," *IEEE Sensors Journal*, vol. 11, no. 7, pp. 1607–1616, 2011.

[6] M. Erol, L. F. M. Vieira, and M. Gerla, "AUV-aided localization for underwater sensor networks," in *Proceedings of the 2nd Annual International Conference on Wireless Algorithms, Systems, and Applications (WASA '07)*, pp. 44–54, August 2007.

[7] J. Liu and Y. Zhang, "Error control in distributed node self-localization," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Article ID 162587, 2008.

[8] K. Yu, Y. J. Guo, and M. Hedley, "TOA-based distributed localisation with unknown internal delays and clock frequency offsets in wireless sensor networks," *IET Signal Processing*, vol. 3, no. 2, pp. 106–118, 2009.

[9] M. Vemula, M. F. Bugallo, and P. M. Djurić, "Sensor self-localization with beacon position uncertainty," *Signal Processing*, vol. 89, no. 6, pp. 1144–1154, 2009.

[10] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1–4, pp. 267–280, 2003.

[11] C. Wang and L. Xiao, "Sensor localization in concave environments," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, article 1, 2008.

[12] J. W. Wan, N. Yu, R. J. Feng, Y. F. Wu, and C. M. Su, "Localization refinement for wireless sensor networks," *Computer Communications*, vol. 32, pp. 1515–1524, 2009.

[13] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, Philadelphia, Pa, USA, 2004.

[14] R. J. Feng, "Multihop localisation with distance estimation bias for 3D wireless sensor networks," *Electronics Letters*, vol. 48, no. 14, pp. 884–886, 2012.

[15] B. Efron, "Bootstrap methods: another look at the jackknife," *Annals of Statistics*, vol. 7, no. 1, pp. 1–26, 1979.

[16] A. M. Zoubir and B. Boashash, "The bootstrap and its application in signal processing: an attractive tool for assessing the accuracy of estimators and testing hypothesis for parameters in small data-sample situations," *IEEE Signal Processing Magazine*, vol. 15, no. 1, pp. 56–76, 1998.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 547124, 14 pages
http://dx.doi.org/10.1155/2013/547124

Research Article

A Cluster-Based Consensus Algorithm in a Wireless Sensor Network

Yanwei Li,[1] Zhenyu Zhou,[2] and Takuro Sato[1]

[1] *Graduate School of Global Information and Telecommunication Studies, Waseda University, 1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo 169-0051, Japan*
[2] *School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China*

Correspondence should be addressed to Yanwei Li; aliyanweia@ruri.waseda.jp

Received 20 October 2012; Revised 30 December 2012; Accepted 21 January 2013

Academic Editor: Nianbo Liu

In this paper, we propose an average connectivity degree cluster (ACDC) scheme gossip algorithm to improve the convergence speed and the accuracy of the consensus, when a common decision is needed for a certain phenomenon in a distributed network. We analyze the effects of the initial value, the network topology (regular and irregular), and the number of clusters on the algorithm convergence rate as well as the accuracy of the value when reaching consensus. A utility function is developed based on two parameters, iteration and relative error, to help the network designers make an optimal decision based on their requirements. An irregular sensor model which is based on the degree of irregular (DOI) radius is introduced to evaluate the robustness of the algorithm. The simulation results demonstrate that for any initial value and network topology, the proposed ACDC gossip algorithm can yield results that are 50% closer to the real average value than the referenced standard gossip and grid cluster gossip algorithms. With different DOI values, our ACDC gossip algorithm can still reach lower relative error compared with other gossip algorithms, which demonstrates that our algorithm is robust enough to be executed in the network.

1. Introduction

The advancement of radio equipped modules and miniaturization of electronic components motivate the development of wireless sensor network (WSN) in which numerous distributed sensor nodes are usually deployed to perform a wide variety of applications, such as monitoring, surveillance, security, health care, and load balancing [1, 2]. Nodes are usually deployed randomly in an ad hoc manner, and for certain tasks, the detection values at different nodes are conditionally independent. Conventionally, tasks are executed in a centralized manner that is straightforward to implement. However, it is not scalable for an increasing number of nodes and sometimes it is expensive and impossible to deploy and maintain such a central controller [3]. Thus, the management technique and distributed decision-making algorithm that organize these multiple distributed agents to carry out a task cooperatively have been extensively studied in recent years. Individual detection by one node in the distributed dynamic WSN system is not sufficient to perform decision making without knowledge of the global network. A consistent decision must be reached among these geographically dispersed sensor nodes through some type of information exchange mechanism. This decision, based on common interests, is referred to as reaching consensus using the detection values of the sensor nodes.

Although the consensus algorithm has been thoroughly studied in the control area, it is of vital important in the distributed sensor network. It is acted as a way to achieve globally optimal decision in a totally decentralized way, without sending all the sensors data to a fusion center [4]. Recently, the most attractive consensus algorithm is the gossip algorithm [5], where pairs of nodes are selected at random to exchange and update their values. Compared with routing algorithm, it is robust and easily implemented. It is not necessary to put much effort on route discovery and route maintenance, and it is a distributed iterative information exchange scheme. However, random information exchange between neighbors also leads to overhead and increases the time to reach consensus in the network. In addition, the

connectivity of the network affects the accuracy of the final consensus value. Therefore, the following two issues should be taken into account when designing the gossip strategy in the network.

(i) Decrease the amount of time required to reach consensus, that is, the convergence rate of the algorithm [6].

(ii) Improve the accuracy of the node value when reaching consensus, that is, the convergence accuracy of the algorithm.

Lots of research has worked on improving the convergence rate of gossip algorithm. Geographic gossip, which combined the gossip algorithm with geographic routing, was recently proposed [7]. This algorithm increases the diversity of the pairwise gossip operation by randomly choosing pairwise gossip nodes within the entire network rather than selecting them from adjacent nodes. The improved approach of geographic gossip was path averaging [8], where the average was performed at each node along the route between the exchanging pair nodes. However, in these two mechanisms, the probability of packet loss increased when sending messages along longer routes. Additionally, as the distance between pairwise nodes that are exchanging information increases, extra energy is consumed to set up and maintain the two-way route between them. The broadcast gossip algorithm which takes advantage of the broadcast characteristic of the wireless medium was proposed [9]. This scheme enables all the neighbors of the wake-up node to listen to the data transmission and perform updates. The other approach that makes use of the broadcast characteristic of wireless medium was the eavesdropping gossip [10], where each node can overhear the data broadcasted by its neighbors and the exchange pair of one node is optimally based on all the data that it received. Subsequently, cluster-based gossip algorithm has been proposed [3, 11, 12]. In [3], each node had a timer which was decremented by 1 at each time and the cluster head was chosen as one node's timer expires. This cluster formation method is simple and easy to implement and to distribute. However, it is impractical in reality because some of the nodes may not have chance to join a cluster. In [11], an algorithm that combines cluster and geographic routing was proposed for a large-scale sensor network. The network is firstly divided into grid clusters, and then the standard gossip algorithm is executed to reach a local consensus in each cluster area. A representative node is subsequently chosen in each cluster, and pairwise gossip is executed among these representative nodes via multihop routing until the consensus goal is reached. However, in the cluster stage, the nodes are divided into groups according to their locations. Thus, an imbalance in node numbers in different cluster slows the convergence rate for reaching consensus. In [12], the authors analyzed the data transmission scheme in the cluster, based wireless network, but they did not mention how to form a cluster, and if the cluster head is collapses, the whole network can no longer reach consensus.

Referring to cluster mechanisms, different cluster algorithms are proposed. For example, the Lowest Identifier (LID) [13] which chooses the node with the lowest ID as a cluster head is a simple clustering method. Its cluster formation method is similar to [3] and its cluster head chosen method is similar to [11]. Highest-Connectivity Degree Algorithm (HCDA) [14] is a connectivity-based cluster formation algorithm which is based on the neighbor number of a node. In HCDA, there is no restriction on the number of nodes in a cluster. When the number of nodes in one cluster is too large, the burden of the cluster head becomes too heavy which may lead to communication bottleneck. The lifetime of the whole network is short because of the imbalance of the network load. There are also some other algorithms, such as distributed clustering algorithm [15], distributed mobility adaptive clustering [16], and weighted clustering algorithm [17] which introduce weight on the selection of a cluster head. All of these above algorithms have not considered the impacts of the number of nodes in one cluster on the network capacity and throughput. Therefore, in this paper, we introduce a throughput/capacity aware cluster mechanism for the gossip algorithm and evaluate its convergence performance.

The contributions of this paper are the following.

(i) A new Average Connecting Degree Cluster (ACDC) based gossip algorithm is proposed to improve the convergence speed and the consensus accuracy.

(ii) Most research studies regarding consensus only consider the network topology as a random geographic graph (RGG). Few studies have considered the impact of regular, random, and small-world graphs on the convergence rate of consensus [18, 19]. No study has analyzed the convergence rate for an irregular network topology. Hence, in this paper, we investigate the proposed gossip algorithm in irregular network topology, such as C-shaped, I-shaped, and O-shaped topologies.

(iii) A utility function that combines the convergence rate and consensus accuracy is proposed to provide an optimal choice reference for WSN users with regard to their purposes.

(iv) An irregular sensor model is introduced to evaluate the robustness of the algorithm.

This paper is organized as follows: Section 2 presents the proposed network model and describes the problem to be solved. Section 3 provides details about the ACDC-based gossip algorithm. Section 4 summarizes the algorithm performance evaluation and analysis. Section 5 presents our conclusions.

2. Network Model and Problem Formulation

Assume that n static sensor nodes are independently deployed in a unit square area and that the network topology is represented as $G = (V, r, E)$, where $V = \{1, 2, 3, \ldots, n\}$ represents the set of nodes and r is the connectivity radius. A pair of nodes (i, j) is connected and can directly communicate with each other if their Euclidean distance is smaller than r. The edge set is saved in E and the set of node's neighbors in

one hop is denoted by $N(i) = \{j \in V; (i, j) \in E\}$. The degree of this node, which is equal to its number of neighbors, can be defined as $d_i = |N(i)|$ [20].

Each node i in the network has an initial value $x_i(0)$, representing an observation of some type. The initial value vector of all these nodes can be defined as $X(0) = [x_1(0), x_2(0), \ldots, x_n(0)]$. In this paper, we deal with the average consensus which means that the consensus equilibrium value is equal to the average value of the initial value held by each node. It has been reported that the average consensus is reached for the case in which the communication topology is fixed and connected [21]. A connected network is one in which a path exists between every pair of nodes [20]. The average of these values is $\overline{x} = (1/n)\sum_{i=1}^{n} x_i(0)$ [21]. At kth iteration, each node i maintains an estimation $x_i(k)$ that is generally different from that of other nodes. A vector $X(k) = [x_1(k), x_2(k), \ldots, x_n(k)]$ is used to define the values of all the nodes. Suppose the network is connected and the communication relationship is symmetric; that is, node i and node j can receive the information from each other correctly based on the wireless link between them for a given time slot. The ultimate goal of consensus is to drive the estimated vector value $x(k)$ infinitely close to the average vector $\overline{X} = [\overline{x}, \overline{x}, \ldots, \overline{x}]$ with a minimal amount of information exchange. To match the distributed nature of WSN, an asynchronous time model is adopted by the gossip algorithm to trigger the node wake up and execute the gossip algorithm. The clock in each node is assumed to have a tick rate based on the Poisson process.

During the gossip algorithm process, at the $(k-1)$th iteration, node i randomly chooses a neighbor node j to exchange information, and their values are updated according to the following equation:

$$x_i(k) = x_j(k) = \frac{\left[x_i(k-1) + x_j(k-1)\right]}{2}. \quad (1)$$

The metric proposed in [20] is used to evaluate the convergence rate of reaching consensus. This metric defines the normalization of difference between consensus value and the real average value.

Definition 1. For the randomized algorithm using ε as the restriction on the calculation, for any $0 < \varepsilon < 1$, the averaging time $T_{\text{ave}}(\varepsilon)$ is the earliest time at which the nodes' value $x(k)$ is ε close to the average vector with a probability greater than $1 - \varepsilon$ [20]:

$$T_{\text{ave}}(\varepsilon) = \sup_{x(0)\neq 0} \inf \left\{k : \Pr\left(\frac{\|x(k) - \overline{x}\|}{\|x(0) - \overline{x}\|} \geq \varepsilon\right) \leq \varepsilon\right\}. \quad (2)$$

With (2), we define the purpose of the algorithm as satisfying the probability in (2) with at least iteration number k. In Section 3, we introduce a new algorithm to meet this goal.

3. Average-Connectivity-Degree-Cluster- (ACDC-) Based Gossip Algorithm

In this section, we first introduce the average connectivity degree cluster (ACDC) scheme. Then, based on this cluster mechanism, we execute the gossip algorithm in the network to reach a consensus.

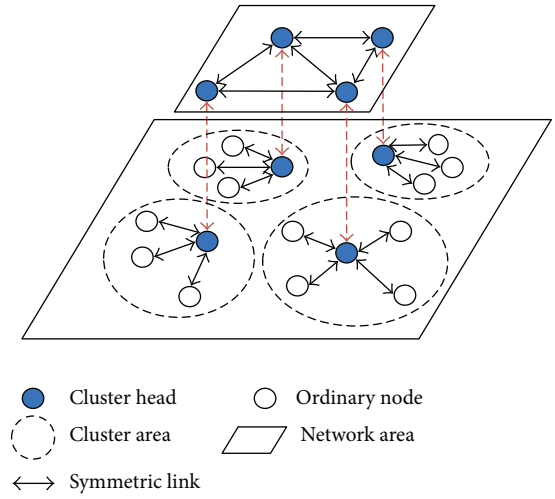


Figure 1: Model of the two-tier system model.

Assume that in an n-node wireless network, each node has the capability of transmitting at C bits per second. It has been shown [22] that the throughput obtained by a node in the network for a randomly chosen destination is only $\Theta(C/\sqrt{n})$ bits per second with using a noninterference protocol, even if the nodes are optimally located, the traffic pattern is optimally assigned, and the transmission range is optimally chosen. Note that the expression $f(n) = \Theta(g(n))$ is used as an asymptotic notation which means when $n \to \infty$, the function $f(n)$ is equal to $g(n)$ within a constant factor.

Suppose that the network is divided into m clusters. The network system model is shown in Figure 1. Here, we define the cluster level as the first tier. The cluster head level forms the second tier. Assume that different clusters are connected by a backbone network. Assume the throughput of each cluster is $C_i\,(i \in m)$. Then, the throughput of the network, C_T, can be obtained by calculating the throughput of all the clusters in the network during a certain time interval. Usually, the throughput of each cluster is different because the number of nodes in each cluster varies. Normally, the affordability of backbone network is constant. Consequently, when the throughputs of some clusters are much higher than others, congestion arises in the backbone network. The whole network may be even paralyzed in the worst circumstances.

Then, we define

$$C_T = mC_i \quad \left(C_i = C_j, i \neq j,\ i, j \in m\right). \quad (3)$$

From the network perspective, the throughput of each cluster should be [18]

$$S = \Theta\left(\frac{C_T}{\sqrt{m}}\right)(\text{bits/sec}). \quad (4)$$

From the cluster perspective, the throughput of each node is

$$S = \Theta\left(\frac{C_i}{\sqrt{n/m}}\right)(\text{bits/sec}). \quad (5)$$

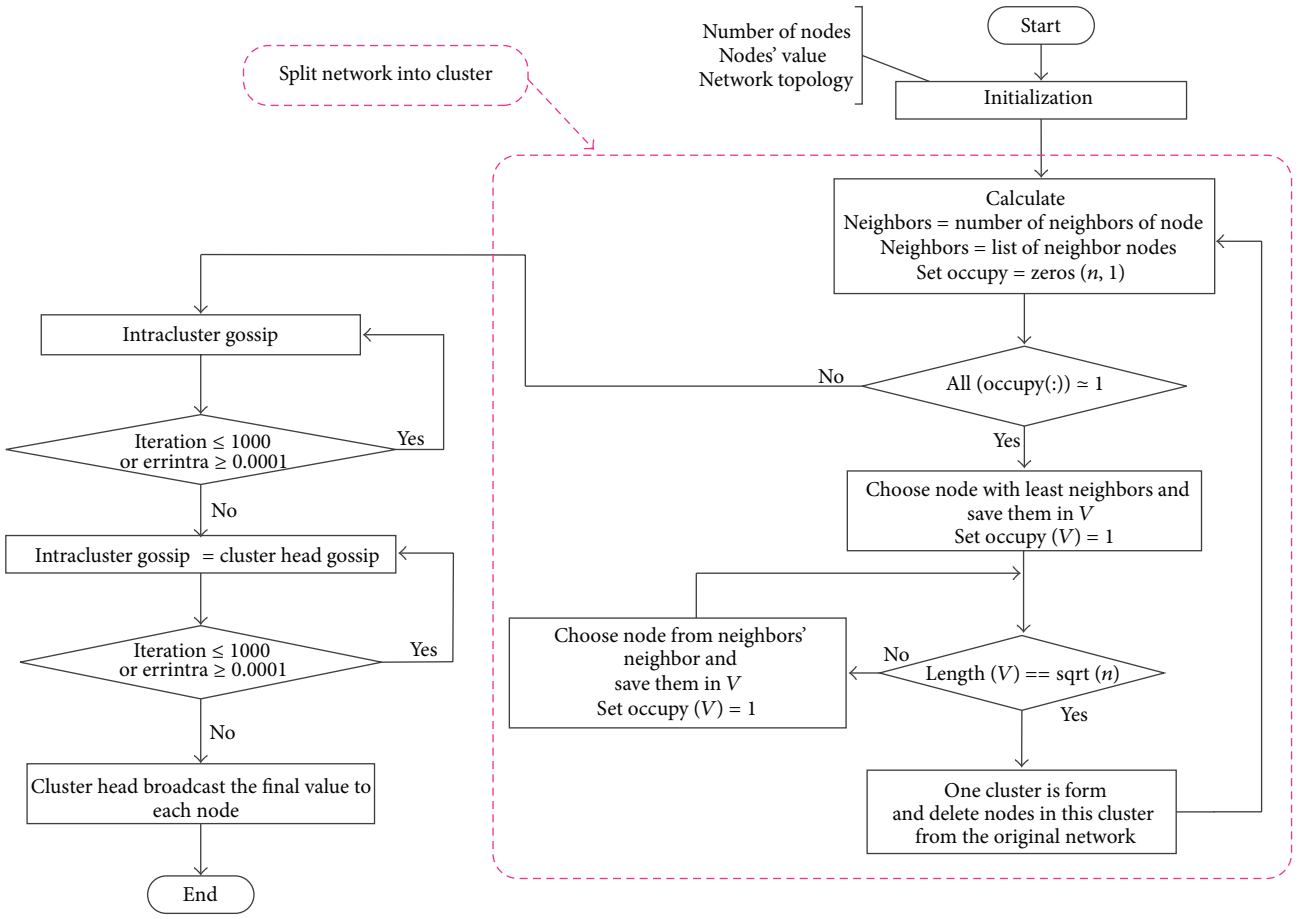


Figure 2: Flowchart of the ACDC gossip algorithm.

The throughput of a cluster is composed of the throughput of each node in the cluster. Thus, according to (3), (4), and (5), we obtain

$$\frac{C_T}{\sqrt{m}} = m \times \frac{C_i}{\sqrt{n/m}}$$
$$\Longrightarrow \frac{mC_i}{\sqrt{m}} = m \times \frac{C_i}{\sqrt{n/m}} \quad (6)$$
$$\Longrightarrow m = \sqrt{n}.$$

Consequently, if the relationship between the number of nodes and the number of cluster can be maintained as described in (6), the network can be stabilized. The number of nodes in the cluster can also be calculated as

$$n_c = \frac{n}{m} = \sqrt{n}. \quad (7)$$

This cluster scheme is used to establish a stable cluster-based network. Based on this cluster scheme, the load in each cluster is the same. Compared with grid cluster based gossip, this scheme alleviates the data transmission burden in certain clusters. The flow chart describing this algorithm is given in Figure 2.

In this algorithm, each node belongs to only one cluster. Although we obtain the optimal number of nodes in each cluster with the help of the backbone network, there initially is no backbone network and or central controller in the network. Knowing the degree of each node, the one with the most neighbors is chosen as the cluster head. Then, the nearest $(\sqrt{n} - 1)$ neighbors are chosen to form a cluster. The reason that we choose the node with the most neighbors as

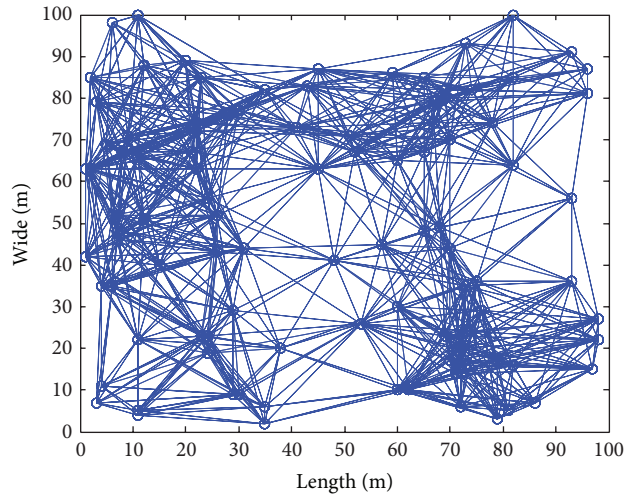


Figure 3: Map of a random geometric graph.

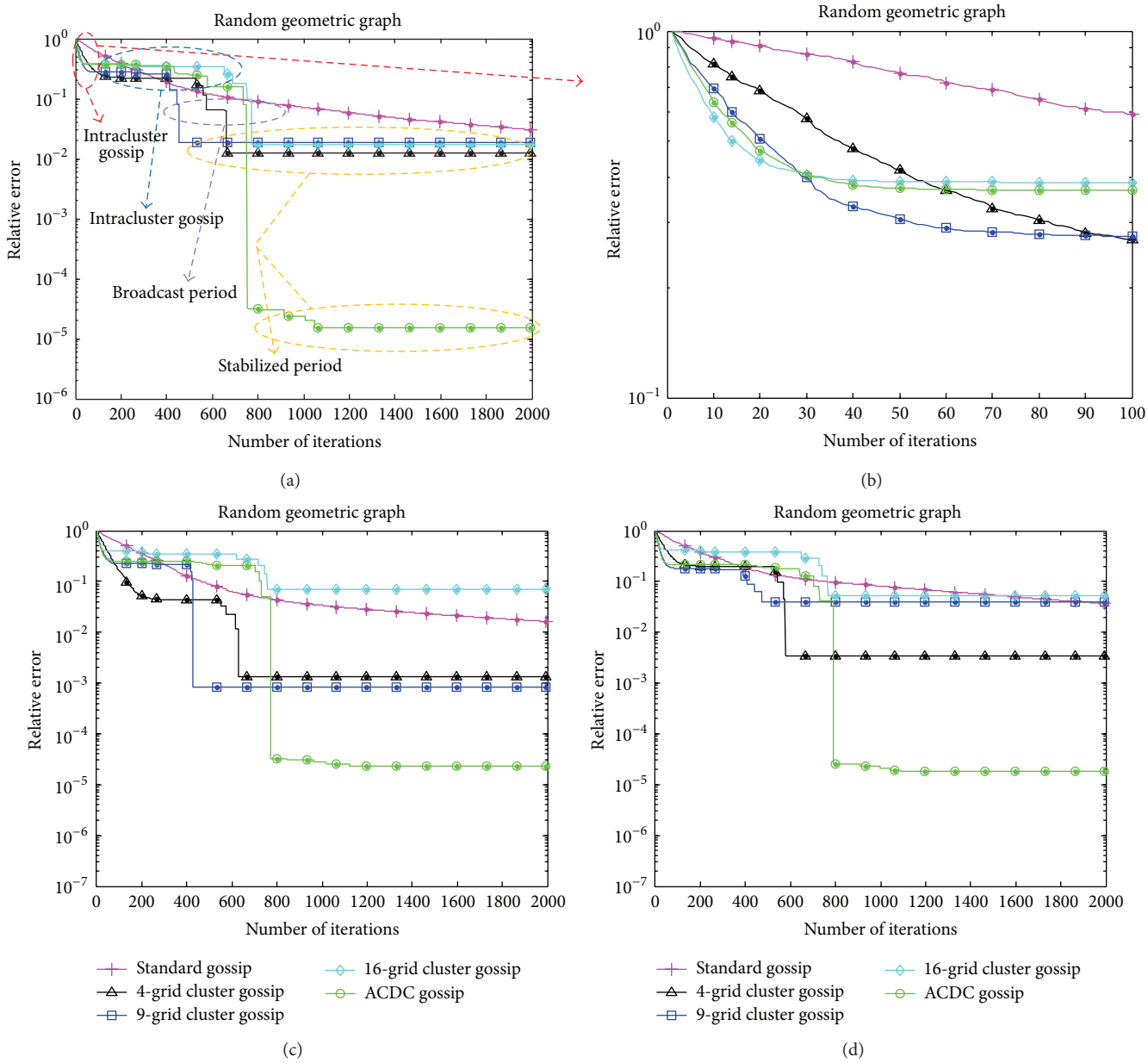


FIGURE 4: (a) Convergence rate comparison with initialization using linear variation. (b) Convergence rate comparison of the first 100 iterations of (a). (c) Convergence rate comparison with initialization using Gaussian distribution. (d) Convergence rate comparison with initialization using independent identical distribution.

the cluster head is that it is more convenient and faster to broadcast the final decision to other nodes via the wireless link connected between them. If there are less than $(\sqrt{n}-1)$ neighbors around the chosen cluster head, we will first form the cluster and then choose the second highest degree node in the network. Then, we can add some of its neighbors to the cluster until the number of nodes in the cluster equals to $(\sqrt{n}-1)$. During the calculation, if we cannot determine an integer according to the root square of (7), the number that is closest to the root square can be selected as the defined number of nodes in each cluster. After the cluster formation, the remaining nodes choose the nearest cluster to join. Consequently, variation in the number of nodes in each cluster is small, and a balanced throughput can be obtained. Note that the cluster is virtually formed in the proposed method. The original connections between different nodes in the network are not changed. When consensus is reached in different clusters, the gossip algorithm is executed at the cluster head level. Because the transmission radius of each node is constant, information exchange between cluster heads can only be accomplished by multihop transmission. When consensus is reached at the cluster head level, the cluster head broadcasts its value to the nodes that are located in its cluster and directly connected to it. Nodes update their values when they receive data from the cluster head. However, if the node is not connected directly with the cluster

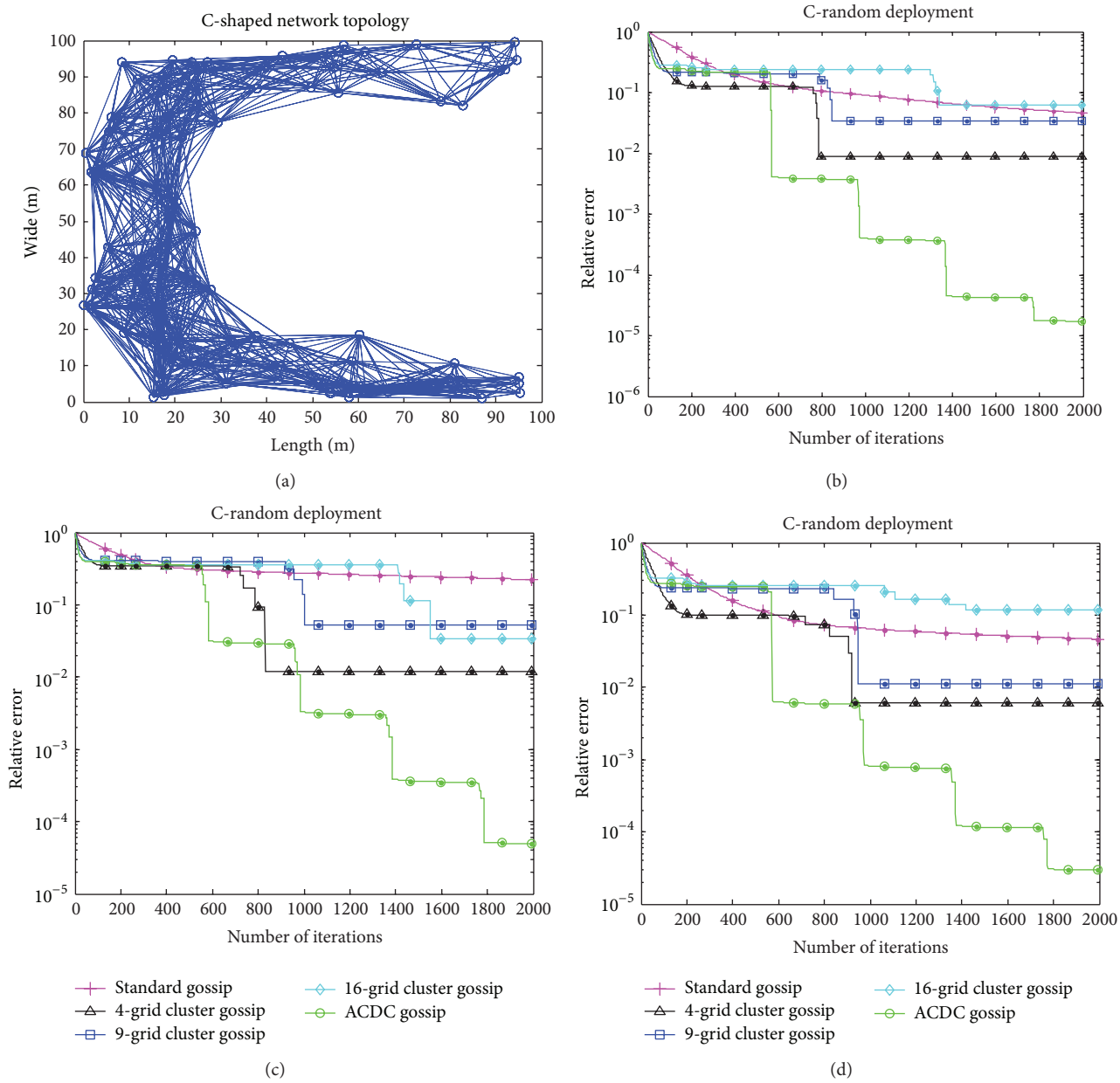


FIGURE 5: (a) C-shaped network topology. (b) Convergence rate comparison with initialization using linear variation. (c) Convergence rate comparison with initialization using Gaussian distribution. (d) Convergence rate comparison with initialization using independent identical distribution.

head, it must obtain this new value through its neighbors. In Figure 2, the part circled in red dash is ACDC cluster formation process.

4. Numerical Results

The performance of the gossip algorithm is evaluated based on the system model developed in the previous section. Suppose that 100 independent nodes are randomly deployed in a $100 \times 100\ \mathrm{m}^2$ area and that each node has a transmission radius of 30 m. The convergence rate and convergence accuracy of our proposal are compared with those of the standard gossip algorithm [20] and the grid cluster gossip algorithm [10]. In the standard gossip, a node randomly chooses one of its neighbors with which to perform information exchange. In the grid gossip algorithm, the whole network is first split into grids with equal areas. Because the network area is square, the number of grids into which the network area can be divided is l^2 (l is a positive integer, $l \in N^+$). Here, if $l = 2$, 3, and 4, the square network area is divided into 4, 9, and 16 grids accordingly. In the ACDC algorithm, a cluster is divided according to the number of nodes in the network. Hence, according to (7), there are 10 clusters in the ACDC algorithm consisting of 100 nodes deployed within the network. A representative node in each grid is chosen to execute the communication with nodes in the first and second

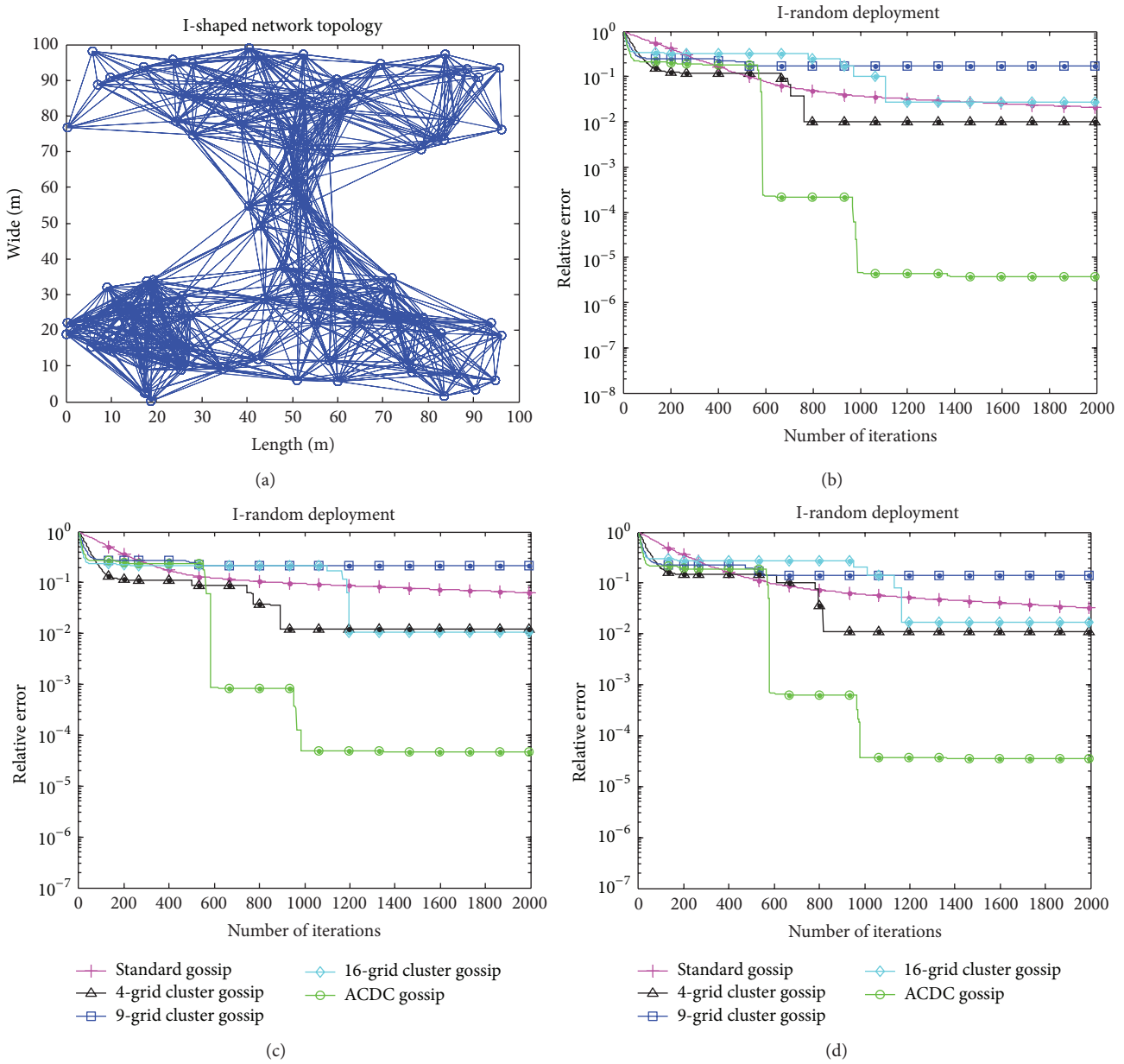


FIGURE 6: (a) I-shaped network topology. (b) Convergence rate comparison with initialization using linear variation. (c) Convergence rate comparison with initialization using Gaussian distribution. (d) Convergence rate comparison with initialization using independent identical distribution.

tiers. During the simulation, we set the iteration number $K = 2000$ and the relative error constraint $\mu = 0.0001$.

4.1. Convergence Performance Evaluation. The relative error is defined in (8). This expression is used to represent the closeness of the consensus value to $\overline{X}$ after determined amount of iterations:

$$\text{err} = \frac{\|x(k) - \overline{x}\|}{\|x(0) - \overline{x}\|}. \tag{8}$$

Based on different initial values, Gaussian distribution, independent identical distribution, and linear variation, the performance is evaluated using the Monte Carlo method. The same boundary is set for the above three types of initial value distributions. The convergence rates based on regular network topology (e.g., random geometric graph (Figure 3)) and irregular network topology (e.g., C-shaped, I-shaped, and O-shaped (Figures 5–7)) are also evaluated.

Based on the network topology in Figures 3 and 4 shows the convergence rates of standard gossip, grid cluster gossip, and the proposed ACDC gossip algorithm. The following six conclusions can be derived from Figure 4.

(1) In Figure 4(a), the convergence rate comparison initialized with linear variation is presented. The convergence rate of the cluster-based gossip algorithm is observed to be faster than that of the standard gossip

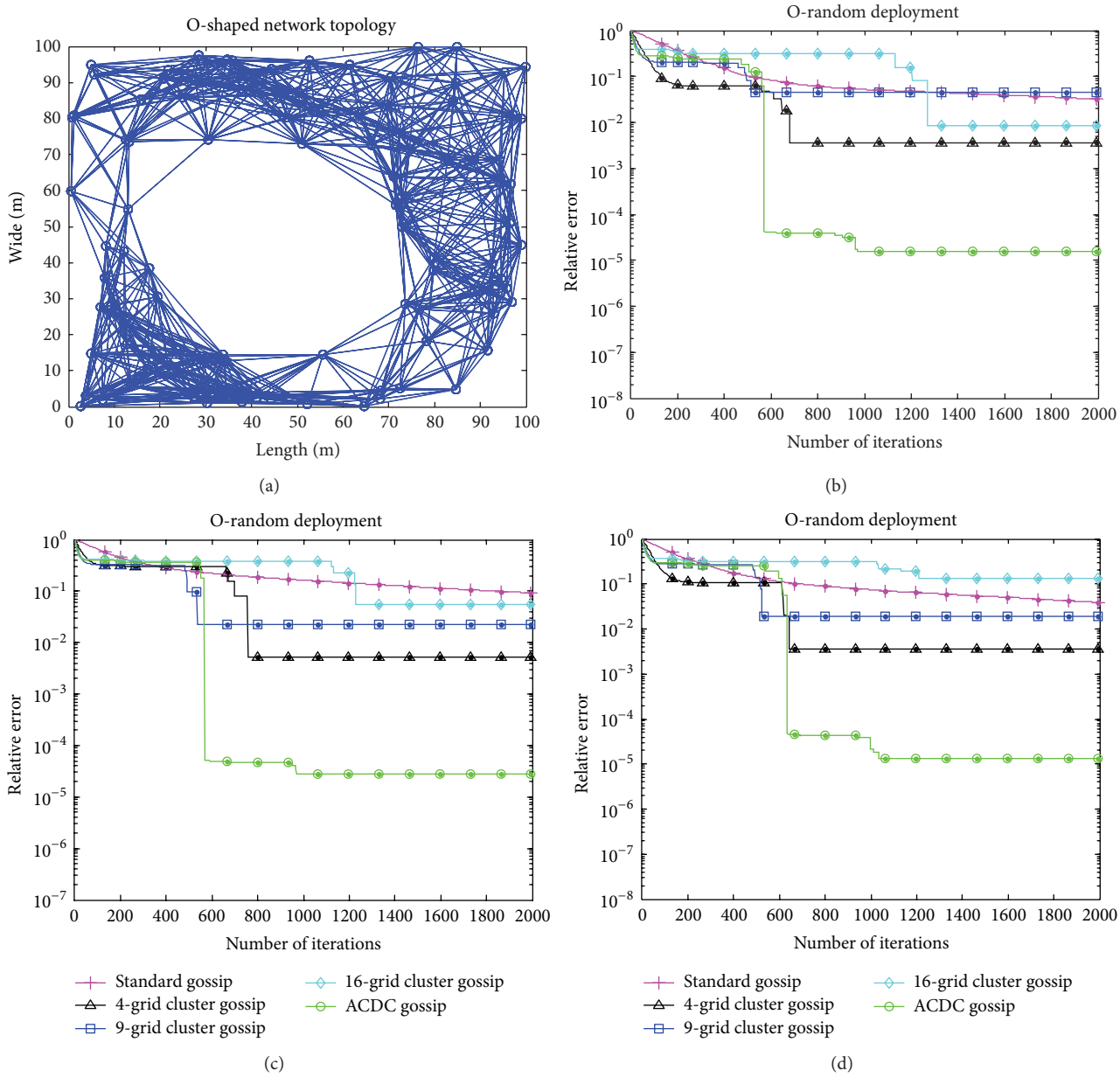


FIGURE 7: (a) O-shaped network topology. (b) Convergence rate comparison with initialization using linear variation. (c) Convergence rate comparison with initialization using Gaussian distribution. (d) Convergence rate comparison with initialization using independent identical distribution.

algorithm. Based on 2,000 iterations, the ACDC gossip algorithm is nearly 50% closer to the real average value than the grid cluster-based gossip algorithm with 4 grids. However, it takes a longer time for the ACDC algorithm to reach consensus compared with other grid gossip algorithms. For example, 410, 620, and 780 iterations are needed for the 9-grid, 4-grid, and 16-grid cluster-based gossip algorithms, respectively, to reach consensus. However, nearly 1,010 iterations are needed for the ACDC gossip algorithm to reach consensus.

(2) In Figure 4(b), the first 100 iterations of Figure 4(a), representing intracluster gossip are shown. In the first 30 iterations, the 16-grid gossip algorithm is observed to be faster than the other algorithms. This result is due to the presence of 16-pair nodes exchanging information at the same time in the former algorithm, whereas there are 4, 9, and 10 pairs of nodes in 4-grid, 9-grid, and ACDC algorithms, respectively. Based on this figure, 35, 35, 90, and more than 100 iterations are required for the ACDC gossip, 16-grid gossip, 9-grid gossip, and 4-grid gossip algorithms, respectively, to reach consensus in each cluster. Therefore, during the intracluster gossip, the relative error of 16-grid gossip algorithm is less than that of the ACDC gossip algorithm in the same iteration, which means that the

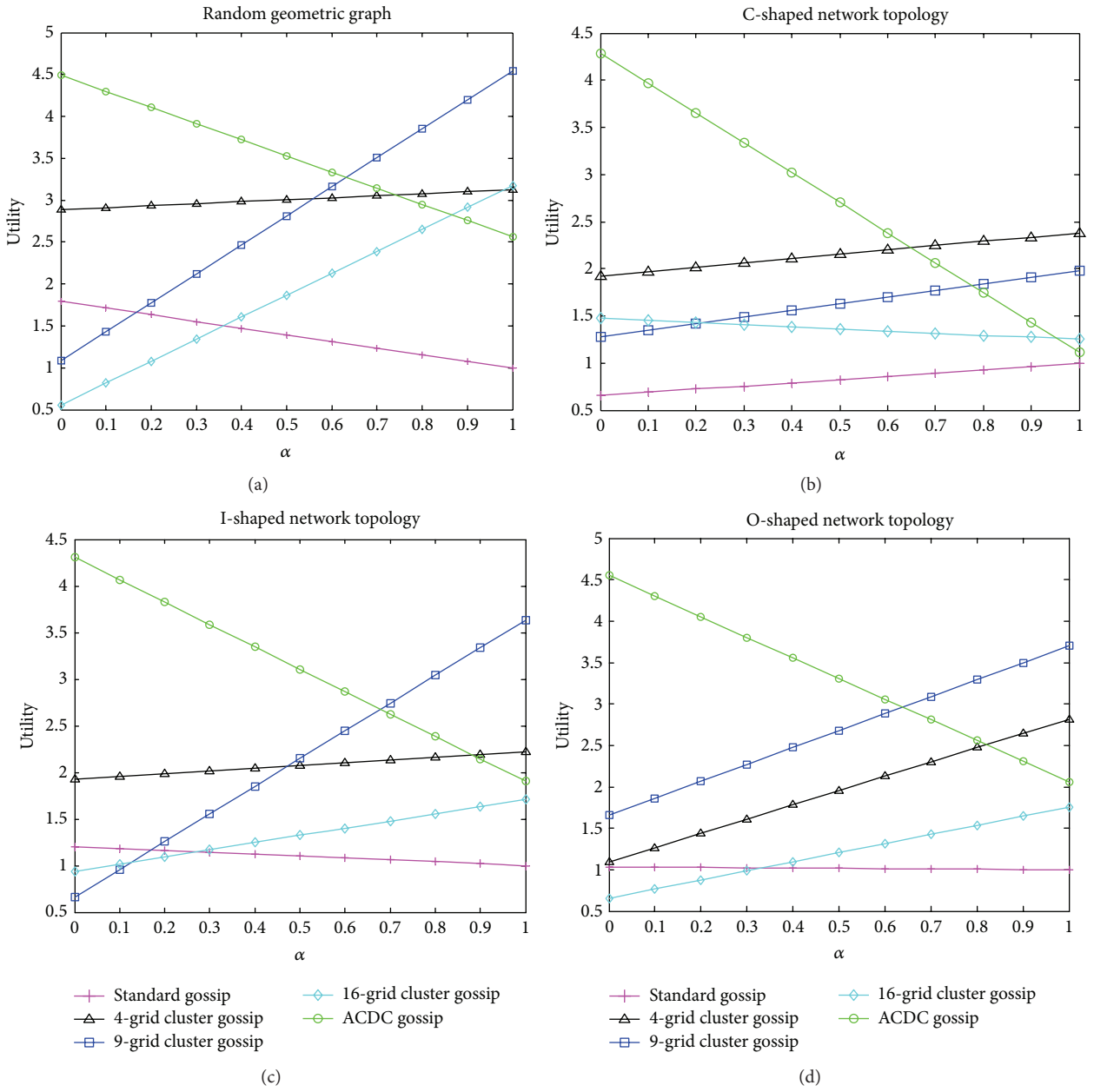


FIGURE 8: Utility of different algorithm for different network topology. (a) Random geometric graph (b) C-shaped network topology (c) I-shaped network topology (d) O-shaped network topology.

convergence rate of the 16-grid cluster based gossip algorithm is faster than ACDC gossip algorithm. Similar results can be obtained for the comparison between the 16-grid and 4-grid gossip algorithms or 16-grid and 9-grid gossip algorithms. Please refer to the appendix for the proof.

(3) In Figure 4(a), the area circled in blue represents intercluster gossip. During this period, there are 16, 9, 4, and 10 cluster heads for the 16-grid, 9-grid, 4-grid, and ACDC gossip algorithms, respectively, in the second tier. At each iteration, only two cluster heads are chosen to exchange their information. The values of all other nodes in the network are kept constant. Therefore, the variation in the relative error is notably small.

(4) In Figure 4(a), the area circled in purple indicates the broadcast period. In this period, the cluster head broadcasts consensus value to nodes in its cluster area. All of the nodes in the network update their value at the same time. Hence, there is a rapid decrease in the relative error.

(5) In Figure 4(a), the area circled in yellow illustrates the stabilized period, which means that consensus has been reached. As shown in the figure, the relative

Table 1: Parameters of iteration and relative error.

Topology	Algorithm	Iteration	Relative error
Random geometric graph	Standard gossip	2000	$1.62 * 10^{-2}$
	4-grid gossip	640	$1.3 * 10^{-3}$
	9-grid gossip	440	$8.2874 * 10^{-2}$
	16-grid gossip	630	$2.771 * 10^{-1}$
	ACDC gossip	780	$3.2186 * 10^{-5}$
C-shaped	Standard gossip	2000	$0.2.222 * 10^{-1}$
	4-grid gossip	840	$1.19 * 10^{-2}$
	9-grid gossip	1010	$0.05.27 * 10^{-2}$
	16-grid gossip	1600	$3.37 * 10^{-2}$
	ACDC gossip	1800	$5.1369 * 10^{-5}$
I-shaped	Standard gossip	2000	$6.34 * 10^{-2}$
	4-grid gossip	900	$1.19 * 10^{-2}$
	9-grid gossip	550	$2.177 * 10^{-1}$
	16-grid gossip	1170	$1.153 * 10^{-1}$
	ACDC gossip	1050	$4.9625 * 10^{-5}$
O-shaped	Standard gossip	2000	$9.23 * 10^{-2}$
	4-grid gossip	710	$8.07 * 10^{-2}$
	9-grid gossip	540	$2.2 * 10^{-2}$
	16-grid gossip	1140	$2.215 * 10^{-1}$
	ACDC gossip	970	$2.8143 * 10^{-5}$

error of the ACDC gossip algorithm is 10^{-5} after 2,000 iterations, which is much lower than those of the grid gossip (approximately 10^{-2}) and standard gossip ($10^{-1.7}$) algorithms.

(6) In Figure 4(c), the results for an initial values with Gaussian distribution and a comparison of the convergence rates for different gossip algorithms are shown, and Figure 4(d) compares the convergence based on initialization with independent identical distribution values. The results shown in both of these two figures confirm that the proposed algorithm can attain higher convergence accuracy than the conventional algorithms, which is independent on the initial values. The consensus value and the relative error vary with different initialization values.

In Figure 5, the convergence rates based on a C-shaped network topology are illustrated. Based on this figure, the ACDC-based gossip algorithm is able to achieve a much lower relative error compared with other gossip algorithms, indicating that the value held by each node is much closer to the desired value.

In Figure 6, the convergence rates based on an I-shaped network topology are shown. For an I-shaped topology, the relative error for the cluster-based gossip algorithm with 9 grids is the most inefficient. The second worst relative error is observed for the 16-grid cluster-based gossip algorithm. Furthermore, the 4-grid cluster-based gossip algorithm reaches consensus more quickly compared with the ACDC gossip algorithm, but the relative error of the latter algorithm is 50% lower because the number of nodes at the cluster head level

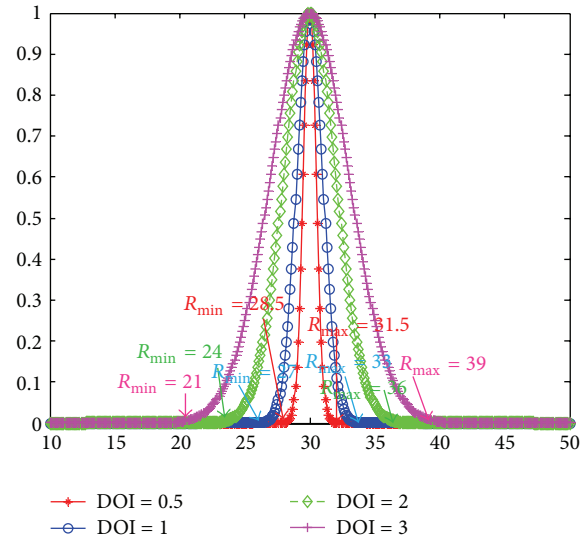


Figure 9: The relationship between R_{min}, R, and R_{max}.

in the 4-grid gossip algorithm is less than that in the ACDC algorithm.

In Figure 7, the convergence rates based on an O-shaped network topology are presented. The results for an O-shaped topology also demonstrates that the proposed ACDC algorithm is superior to the referenced algorithm with regard to accuracy. The 16-grid cluster gossip algorithm is the slowest scheme to reach consensus for this topology.

Based on the results presented in Figures 4–7, it can be concluded that our ACDC gossip algorithm is superior to the referenced algorithms in terms of convergence accuracy. This incense in accuracy is because the algorithm's accuracy is primarily based on the cluster scheme. In particular, in irregular network topologies, the grid cluster is divided according to node locations. There are clusters with no nodes or small numbers of nodes in an area. The imbalanced number of nodes in different clusters results in a lower accuracy in the final consensus value. For example, in the C-shaped topology, there are steps observed in ACDC gossip which is shown in the green line because several loops are executed to reach consensus in each of the first 2,000 iterations. We can also observe that the increase in the number of clusters in the grid-cluster gossip algorithm would not improve the convergence consensus accuracy. Therefore, we omit the simulations when the network is divided into 25 or even more grids.

4.2. Utility Comparison. A utility function is developed to evaluate the efficiency of these algorithms. We define the ability with the mathematical formula in (A.7). This function consists of two parameters, iteration and relative error. Here, we define the iteration as t, while the relative error as e:

$$U = \alpha \cdot \frac{2000}{t} + (1 - \alpha) \cdot \left(-\log_{10} e\right) \quad \alpha \in [0, 1], \tag{9}$$

where α is a weight parameter that represents the importance placed on iteration or relative error by users. In this equation, when the t value is smaller, the value of U will be larger.

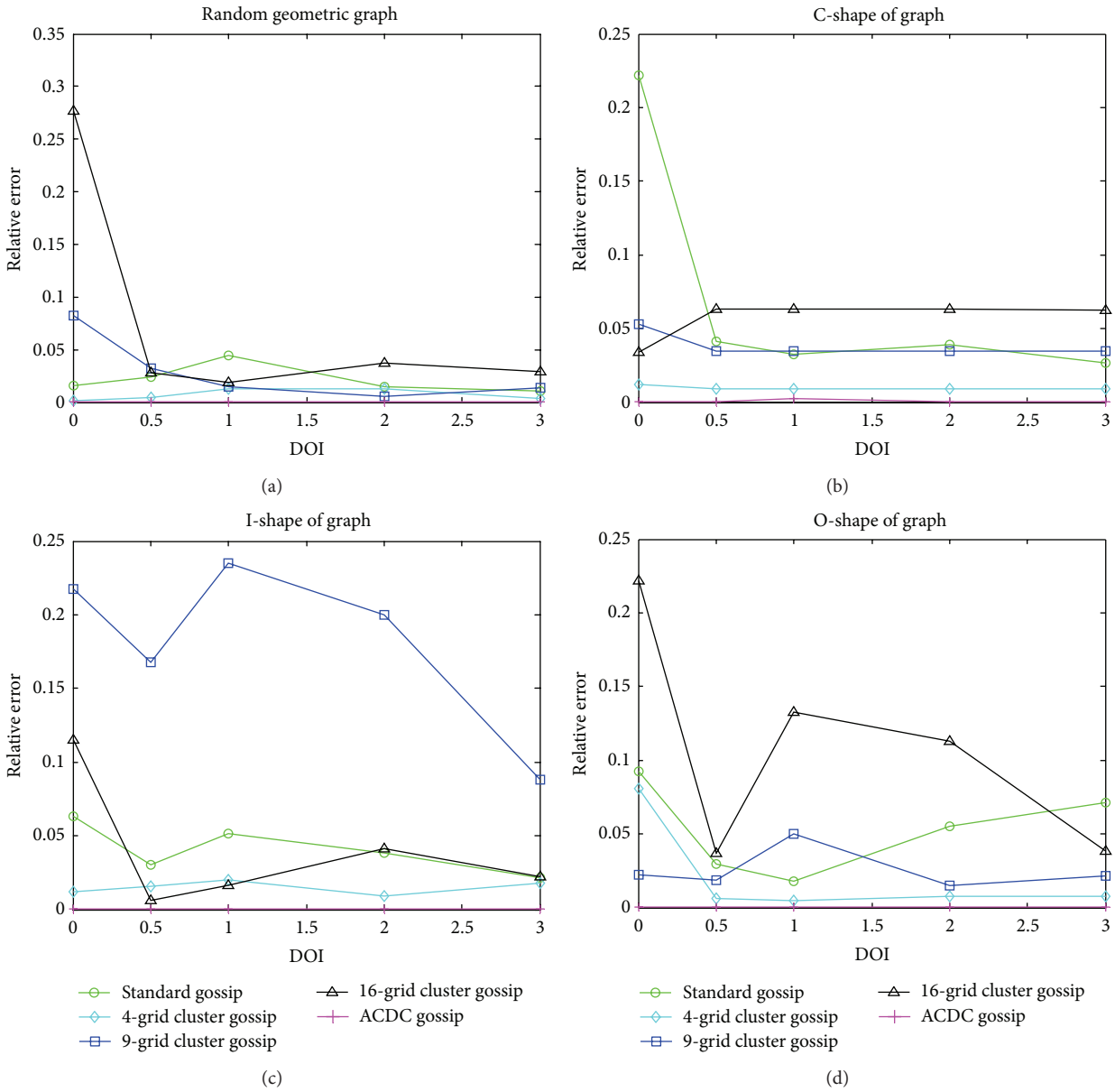


FIGURE 10: Consensus relative error for different network topology. (a) Random geometric graph. (b) C-shaped network topology. (c) I-shaped network topology. (d) O-shaped network topology.

A larger value of U can also be obtained with a smaller e value. According to this equation, the network designers can find an optimal pair of values from which they can obtain the highest utility. The consensus iteration number and relative error of Gaussian distribution for different topologies are given in Table 1.

Using the simulation results in Table 1 as parameters for our utility function in (A.7), we obtain the graphs in Figure 8.

Based on Figure 8, it can be concluded that, when α is greater than 0.6, that is, the network designers pay much more attention on the convergence rate, the 4-grid gossip algorithm is a better choice in the C-shaped network topology, while the 9-grid gossip algorithm is a better choice for attaining higher utility in other types of topologies. When the accuracy requirement is not very strict and minimizing the amount of the time spent on data transmission is desired, a grid cluster-based gossip algorithm is the superior choice. If α is less than 0.6, the network designers place greater importance on the accuracy of the executed algorithm. The utility of our proposed algorithm is much greater than that of the other algorithms making it a viable choice.

4.3. Robustness Analysis. Robustness in WSN means how the algorithm is capable to perform in some situations, such as packet loss, node death, or node mobility. For node death and node mobility, the gossip algorithm executes data change with random pair of nodes to combat the topology change. It is not necessary to consider the routing protocol.

Table 2: Parameters of DOI and relative error.

Topology	Algorithm	Relative error	DOI				
			0	0.5	1	2	3
Random geometric graph	Standard gossip		$1.62 * 10^{-2}$	$2.39 * 10^{-2}$	$4.42 * 10^{-2}$	$1.49 * 10^{-2}$	$1.07 * 10^{-2}$
	4-grid gossip		$1.3 * 10^{-3}$	$4.9 * 10^{-3}$	$1.31 * 10^{-3}$	$1.3 * 10^{-2}$	$3.3 * 10^{-3}$
	9-grid gossip		$8.29 * 10^{-2}$	$3.25 * 10^{-2}$	$1.47 * 10^{-2}$	$5.8 * 10^{-3}$	$1.42 * 10^{-2}$
	16-grid gossip		$2.77 * 10^{-1}$	$2.84 * 10^{-2}$	$1.93 * 10^{-2}$	$3.7 * 10^{-2}$	$2.93 * 10^{-2}$
	ACDC gossip		$3.22 * 10^{-5}$	$3.57 * 10^{-5}$	$4.07 * 10^{-5}$	$2.41 * 10^{-5}$	$3.29 * 10^{-5}$
C-shaped	Standard gossip		$2.22 * 10^{-1}$	$4.11 * 10^{-2}$	$3.22 * 10^{-2}$	$3.92 * 10^{-2}$	$2.7 * 10^{-2}$
	4-grid gossip		$1.19 * 10^{-2}$	$8.9 * 10^{-3}$	$8.9 * 10^{-3}$	$8.9 * 10^{-3}$	$8.9 * 10^{-3}$
	9-grid gossip		$5.27 * 10^{-2}$	$3.46 * 10^{-2}$	$3.47 * 10^{-2}$	$3.46 * 10^{-2}$	$3.47 * 10^{-2}$
	16-grid gossip		$3.37 * 10^{-2}$	$6.29 * 10^{-2}$	$6.29 * 10^{-2}$	$6.29 * 10^{-2}$	$6.28 * 10^{-2}$
	ACDC gossip		$5.14 * 10^{-5}$	$2.11 * 10^{-5}$	$2.6 * 10^{-3}$	$1.32 * 10^{-5}$	$3.63 * 10^{-5}$
I-shaped	Standard gossip		$6.34 * 10^{-2}$	$3.02 * 10^{-2}$	$5.15 * 10^{-2}$	$3.84 * 10^{-2}$	$2.14 * 10^{-2}$
	4-grid gossip		$1.19 * 10^{-2}$	$1.59 * 10^{-2}$	$1.99 * 10^{-2}$	$0.9 * 10^{-2}$	$1.81 * 10^{-2}$
	9-grid gossip		$2.18 * 10^{-1}$	$1.68 * 10^{-1}$	$2.35 * 10^{-1}$	$1.99 * 10^{-1}$	$8.78 * 10^{-2}$
	16-grid gossip		$1.15 * 10^{-1}$	$6.4 * 10^{-3}$	$1.62 * 10^{-2}$	$4.14 * 10^{-2}$	$2.22 * 10^{-2}$
	ACDC gossip		$4.96 * 10^{-5}$	$3.02 * 10^{-5}$	$2.81 * 10^{-5}$	$3.3 * 10^{-5}$	$2.7 * 10^{-5}$
O-shaped	Standard gossip		$9.23 * 10^{-2}$	$2.96 * 10^{-2}$	$1.8 * 10^{-2}$	$5.54 * 10^{-2}$	$7.11 * 10^{-2}$
	4-grid gossip		$8.07 * 10^{-2}$	$6.1 * 10^{-3}$	$4.8 * 10^{-3}$	$7.9 * 10^{-3}$	$7.8 * 10^{-3}$
	9-grid gossip		$2.2 * 10^{-2}$	$1.88 * 10^{-2}$	$5.03 * 10^{-2}$	$1.48 * 10^{-2}$	$2.12 * 10^{-2}$
	16-grid gossip		$2.22 * 10^{-2}$	$3.72 * 10^{-2}$	$1.33 * 10^{-1}$	$1.13 * 10^{-1}$	$3.86 * 10^{-2}$
	ACDC gossip		$2.81 * 10^{-5}$	$1.25 * 10^{-5}$	$2.04 * 10^{-5}$	$2.12 * 10^{-5}$	$1.99 * 10^{-5}$

If some of the neighbors around one node die or move out of the communication range of the node, it can pick up the other neighbor to exchange information. Here, we do not consider the isolated node case. Therefore, all the algorithms mentioned in this paper are robust to the network topology change.

Packet loss is usually caused by variation in wireless links. The radio irregularity of each node, which is caused by the propagation medium and hardware devices, is the main reason for the asymmetric links [23]. When a signal propagates within a wireless medium, it may be reflected, diffracted, and scattered [24]. Reflection occurs when a signal wave impacts on an object which has larger dimensions than the wavelength of the signal. It usually occurs with impacts with the surface of the earth, buildings, and walls. Diffraction occurs when the signal is obstructed by a sharp irregular surface and scattering occurs when the medium through which the signal wave travels contain a large numbers of objects with dimensions that are smaller than the signal wavelength. Consequently, the transmission radios in different directions are various. The other reason of irregular radius is the hardware difference of each node, especially the antenna gain, even if the manufacture is the same.

To evaluate whether our algorithm works well in an irregularity radio scenario, we introduce the irregular sensor model which is proposed in [25]. For each sensor node, the radio propagation range is predefined as R, and the effective radio range R_{eff} is defined by the Gaussian distribution with a mean of R and a standard derivation of DOI, where DOI represents the degree of irregularity of R_{eff}. Here, we define the range of R_{eff} is from $R_{\min} = R - 3 * \text{DOI}$ to $R_{\max} = R + 3 * \text{DOI}$.

The DOI model defines an upper and lower bound on signal propagation. If the distance between two nodes is no larger than $R_{\min}$, we assume that the wireless link between these two nodes is symmetric which means they can receive each other's data correctly. If the distance between two nodes is beyond $R_{\max}$, there is no communication between them. If the distance between two nodes is between $R_{\min}$ and $R_{\max}$, we assume there are three kinds of possible scenarios, symmetric communication and asymmetric communication, which means one node can receive the other node's data correctly while opposite link is disrupted, and no communication.

We defined DOI $= [0, 0.5, 1, 2, 3]$, where DOI $= 0$ means the transmission radius is constant. When DOI $= [0.5, 1, 2, 3]$, the relationship between $R_{\min}$, R, and $R_{\max}$ is shown in Figure 9.

Table 2 and Figure 10 show the relative error value when the network reaches consensus along with the change of DOI. Compared with other gossip algorithms, we find that no matter what kind of network topology, our algorithm performs with the least relative error value, nearly 10^{-3} times smaller than the other algorithms.

5. Conclusions

In this paper, we investigated the convergence rates of gossip algorithms in a dense, randomly deployed WSN with three

types of sensor observation attributes. The use of an average-connectivity-degree-cluster (ACDC-) based gossip algorithm was proposed to improve the convergence rate of consensus decisions. We analyzed the effects of sensor observation attributes, network topology, and the number of clusters on the convergence rate that reached consensus. We also developed a utility function that considers the number of iterations and relative error. An irregular sensor model is introduced to evaluate the robustness of the algorithm. The simulation results show that the proposed ACDC gossip algorithm is much more accurate than the standard gossip and grid cluster-based gossip algorithms for any type of topology. When users place more importance on the algorithm accuracy, the proposed ACDC algorithm should be selected. However, if minimizing the amount of time spent on information transmission is desired, the 4-grid or 9-grid cluster-based gossip algorithms are better choices. We also analyzed and proved that our ACDC algorithm is robust enough in all the network topologies used in this paper. In the future, our objective will be to improve the convergence rate of the proposed algorithm.

Appendix

This appendix is used to prove that the convergence rate of 16-grid gossip is faster than the other gossip algorithms during the intracluster communication period. Take the 16-grid gossip and ACDC algorithms as examples.

From the kth iteration to the $(k+1)$th iteration, there are 16 pairs in the first tier of the 16-grid cluster:

$$\begin{aligned}\mathrm{err}_{16} &= \frac{\left\|X(k+1)-\overline{X}\right\|}{\left\|X(0)-\overline{X}\right\|} \\ &= \frac{\sqrt{\sum_{i=1}^{84}\left(x_i(k)-\overline{x}\right)^2+\sum_{j=1}^{16}\left(x_j(k+1)-\overline{x}\right)^2}}{\left\|X(0)-\overline{X}\right\|}.\end{aligned} \tag{A.1}$$

There are 10 pairs in the first tier in ACDC cluster:

$$\begin{aligned}\mathrm{err}_{\mathrm{ACDC}} &= \frac{\left\|X(k+1)-\overline{X}\right\|}{\left\|X(0)-\overline{X}\right\|} \\ &= \frac{\sqrt{\sum_{i=1}^{90}\left(x_i(k)-\overline{x}\right)^2+\sum_{j=1}^{10}\left(x_j(k+1)-\overline{x}\right)^2}}{\left\|X(0)-\overline{X}\right\|}.\end{aligned} \tag{A.2}$$

To compare (7) and (8), we need to compare $\sqrt{\sum_{i=1}^{84}(x_i(k)-\overline{x})^2+\sum_{j=1}^{16}(x_j(k+1)-\overline{x})^2}$ and $\sqrt{\sum_{i=1}^{90}(x_i(k)-\overline{x})^2+\sum_{j=1}^{10}(x_j(k+1)-\overline{x})^2}$.

The value of each node at the kth iteration is assumed to be equal in both 16-grid and ACDC schemes.

We then obtain

$$\begin{aligned}&\sqrt{\sum_{i=1}^{90}\left(x_i(k)-\overline{x}\right)^2+\sum_{j=1}^{10}\left(x_j(k+1)-\overline{x}\right)^2} \\ &\quad=\sqrt{\sum_{i=1}^{84}\left(x_i(k)-\overline{x}\right)^2+\sum_{i'=1}^{6}\left(x_{i'}(k)-\overline{x}\right)^2+\sum_{j=1}^{10}\left(x_j(k+1)-\overline{x}\right)^2} \\ &\sqrt{\sum_{i=1}^{84}\left(x_i(k)-\overline{x}\right)^2+\sum_{j=1}^{16}\left(x_j(k+1)-\overline{x}\right)^2} \\ &\quad=\sqrt{\sum_{i=1}^{84}\left(x_i(k)-\overline{x}\right)^2+\sum_{i''=1}^{6}\left(x_{i''}(k+1)-\overline{x}\right)^2+\sum_{j=1}^{10}\left(x_j(k+1)-\overline{x}\right)^2}.\end{aligned} \tag{A.3}$$

The $\sum_{j=1}^{10}(x_j(k+1)-\overline{x})^2$ part in (9) and (A.1) are supposed to be equal. Thus, we only need to compare $\sum_{i'=1}^{6}(x_{i'}(k)-\overline{x})^2$ and $\sum_{i''=1}^{6}(x_{i''}(k+1)-\overline{x})^2$.

Using (1), we obtain

$$x_{i''}(k+1)=\frac{x_{i'}(k)+x_{j'}(k)}{2}. \tag{A.4}$$

Then, the comparison is equivalent to the comparison between $(x_{i'}(k)-\overline{x})^2+(x_{j'}(k)-\overline{x})^2$ and $2(x_{i''}(k+1)-\overline{x})^2$.

If we denote

$$\begin{aligned}&f_1=\left(x_{i'}(k)-\overline{x}\right)^2+\left(x_{j'}(k)-\overline{x}\right)^2, \\ &f_2=2\left(x_{i''}(k+1)-\overline{x}\right)^2, \\ &a=x_{i'}(k), \qquad b=x_{j'}(k), \qquad c=\overline{x}.\end{aligned} \tag{A.5}$$

We obtain

$$\begin{aligned}f=f_1-f_2&=\left[(a-c)^2+(b-c)^2\right]-2\left(\frac{a+b}{2}-c\right)^2 \\ &=a^2+b^2+2c^2-2ac-2bc \\ &\quad-\frac{a^2+b^2+4c^2+2ab-4ac-4bc}{2} \\ &=\frac{a^2+b^2-2ab}{2}=\frac{(a-b)^2}{2}.\end{aligned} \tag{A.6}$$

Because $a\neq b$

$$\begin{aligned}&f=f_1-f_2>0 \\ &\Longrightarrow\ f_1>f_2.\end{aligned} \tag{A.7}$$

Therefore, we can conclude that the in the first 100 iterations, 16-grid gossip performs faster than the other algorithm. Similar results can be obtained compared to the 16-grid gossip and other grid gossip algorithms.

Acknowledgment

Y. Li would like to thank the support of the Ph.D. Fellowship Program of the China Scholarship Council.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor network: a survey," *Computer Network*, vol. 38, pp. 393–422, 2002.

[2] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.

[3] W. J. Li and H. Y. Dai, "Cluster-based distributed consensus," *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 28–31, 2009.

[4] S. Sardellitti, M. Giona, and S. Barbarossa, "Fast distributed average consensus algorithms based on advection-diffusion Processes," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 826–842, 2010.

[5] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[6] W. Ren and R. W. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control: Theory and Applications*, Springer, London, UK, 2010.

[7] A. D. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, "Geographic gossip: efficient averaging for sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 1205–1216, 2008.

[8] F. Benezit, A. G. Dimakis, P. Thiran, Vetterli, and M. Gossip, "Along the way: order-optimal consensus through randomized path averaging," in *Proceeding of the Allerton Conference on Communication, Control, and Computing*, pp. 26–28, Allerton, Ill, USA, September 2007.

[9] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, "Broadcast gossip algorithms for consensus," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2748–2761, 2009.

[10] D. Üstebay, B. N. Oreshkin, M. J. Coates, and M. G. Rabbat, "Greedy gossip with eavesdropping," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3765–3776, 2010.

[11] K. I. Tsianos and M. G. Rabbat, "Fast decentralized averaging via multi-scale gossip," in *Proceeding of International Conference on Distributed Computing in Sensor System*, pp. 21–23, Santa Barbara, Calif, USA, June 2010.

[12] M. Zheng, M. Goldenbaum, S. Stanczak, and Y. Haibin, "Fast average consensus in clustered wireless sensor networks by superposition gossiping," in *Proceedings of the IEEE Wireless Communication and Networking Conference*, pp. 1–4, Paris, France, April 2012.

[13] D. J. Baker and A. Ephremides, "The Architectural organization of a mobile radio network via a distributed algorithm," *IEEE Transactions on Communications*, vol. 29, no. 11, pp. 1694–1701, 1981.

[14] M. Gerla and J. Tzu-Chieh Tsai, "Multicluster, mobile, multimedia radio network," *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.

[15] S. Basagni, "Distributed clustering for Ad Hoc networks," in *In Proceeding of the International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 23–25, Perth, Australia, June1999.

[16] S. Basagni, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," in *Proceedings of the 50th IEEE Vehicular Technology Conference (VTC '99)*, pp. 19–22, Amsterdam, The Netherland, September 1999.

[17] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: a weighted clustering algorithm for mobile Ad hoc networks," *Journal of Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[18] O. S. Saber, "Ultrafast consensus in small-world networks," in *Proceeding of American Control Conference*, pp. 8–10, June 2005.

[19] S. A. Aldosari and J. M. F. Moura, "Distributed detection in sensor networks: connectivity graph and small world networks," in *Proceedings of the 39th Asilomar Conference on Signals, Systems and Computers*, pp. 230–234, November 2005.

[20] R. Diestel, *Graph Theory*, vol. 173, Springer, Heidelberg, Germany, 4th edition, 2010.

[21] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[22] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[23] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," *MobiSys 2004*, pp. 125–138, 2004.

[24] T. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.

[25] C. H. Wu and Y. C. Chung, "Heterogeneous wireless sensor network deployment and topology control based on irregular sensor model," *Advances in Grid and Pervasive Computing Lecture Notes in Computer Science*, vol. 4459, pp. 78–88, 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 725869, 8 pages
http://dx.doi.org/10.1155/2013/725869

Research Article

An Energy Distribution and Optimization Algorithm in Wireless Sensor Networks for Maritime Search and Rescue

Huafeng Wu,[1] Qiannan Zhang,[1] Su Nie,[1] Wei Sun,[1] and Xinping Guan[2]

[1] *Merchant Marine College, Shanghai Maritime University, Shanghai 200135, China*
[2] *School of Electronic, Information and Electrical Engrineering, Shanghai Jiaotong University, Shanghai 200240, China*

Correspondence should be addressed to Huafeng Wu; hfwu@shmtu.edu.cn

Received 16 October 2012; Accepted 2 January 2013

Academic Editor: Haigang Gong

Currently, maritime search and rescue (MSR) is mainly depending on the search party, while the searching objects are waiting passively. Therefore, a new method of MSR which is based on the wireless sensor network (WSN) techniques is proposed in this paper. WSN could be self-organized into network and transmit nodes information, such as position information, for search party to accomplish the search and rescue work. However, the application encounters the problems of dynamic adaptability and life cycle limitation at sea. An energy dynamic distribution and optimization algorithm (EDDO), which is based on genetic algorithm (GA), is presented to handle with these problems. The algorithm satisfies the connectivity and energy saving of the network, and the GA with elitism-based immigrants approach is put forward to optimize the poor individuals when the positions of some nodes have changed. Simulation results show that the algorithm can quickly adapt to a dynamic network and reduce energy consumption at the same time.

1. Introduction

Maritime search and rescue (MSR) generally searches the targets by the technology of satellite and radar. There are some new technologies that have been applied to MSR, such as machine vision. However, the search work is dependent on the effort by the search and rescue party, and the rescue targets can only passively wait for the search and rescue. In recent years, with the development of the wireless sensor network (WSN), there are improvements both in computing ability and energy consumption of sensor nodes. This will be helpful for its application in MSR. In MSR-WSN, the sensor nodes were put in the life jackets and can be organized into an ad hoc network. The network locates its nodes and transmits the data of position to the sink nodes which were put on the survival craft. The sink nodes transmit the data to the base station of the SAR ships first, and then the search and rescue network center can receive the information about the targets. By this way, it will be much more quickly and accurately for the rescuers to find the targets.

Wireless sensor networks are suitable for the application of maritime search and rescue because of its features such as self-organization, self-adaptive, multihop, and robustness. In the circumstances of MSR, there is not have any base station, and nodes are highly dynamic and requires smaller size and longer lifetime. However, related researches do not pay enough attention to the application environment of MSR where WSN's nodes could not be replaced and they are moving all the time at sea. Therefore, it is crucial for nodes that their transmit power can adapt to dynamic environmental changes, the network lifetime can be extended, and the connectivity can meet the requirements at the same time.

Evolution algorithm (EA) is a bionic algorithm based on nature selection in biological evolution theory. EA has a good adaptability to complex engineering optimization problems. There are many researches about the EAs in topology control and covering algorithm of WSN. In MSR-WSN, the topology will change and the energy deployment will not be suitable for the topology when the environment changes. The most simple and direct way of energy deployment is to restart the EAs from scratch in case that an environmental change is detected.

But it is more efficient to develop other solutions which make use of knowledge gathered from the original environments. Elitism-based immigrant scheme is a representative one in dynamic environments.

The remainder of the paper is organized as follows. Section 2 introduces the related topology control and is optimized by EAs and energy-based topology control in WSN. Section 3 presents the proposed WSN energy distribution and optimization solution at sea which is based on the genetic algorithm (GA) in EAs. Section 4 introduces the setup of the simulation and the results of analysis. Section 5 concludes this paper and suggests some topics for future research.

2. Related Work

Many studies in MSR research the runaway boat drift model or the best search area via applying the operations research, fuzzy mathematics, fluid mechanics, simulation theory, and so on. There are also some methods which introduce the new information techniques into MSR, for instance, the computer simulation method based on Monte Carlo. Others about multispectral analysis, machine vision, and satellite remote sensing technology are also explored in MSR [1, 2]. A common defect exists in this technology that the targets can only passively be searched; the searching objects cannot send their location information to the search and rescue party.

WSN is capable of real-time monitoring and collecting the objects information and can send the information to gateway nodes. The rapid and self-organization deployment and network survivability features of WSN will meet the environment and application characteristics in MSR. Kim et al. [3] designed a kind of equipment which can send GPS location information and mayday distress signal to help locate victims in offshore area; they also referred to WSN technology, but the key technical issues, such as how to organize the network, are not addressed.

An important purpose of wireless sensor networks topology is to extend the life cycle of the network as possible. In MSR-WSN, the lifetime of the network is also very important. In [4], the optimal coverage problem was taken as a 0/1 sequence problem in which "0" stands for the node was sleeping, "1" stands for the node was working, and the evolution algorithm was adopted to optimize the 0/1 sequence. This method can save the energy effectively, while in MSR-WSN there will not be so many nodes to make some of them sleep, and no one in water can lose the connectivity with others; thus the sleeping mechanism is not the best one. Konstantinidis and Yang [5] defined the dense deployment and power assignment problem (d-DPAP) in WSNs and proposed a multiobjective evolutionary algorithm based on decomposition (MOEA/D) hybridized with a problem-specific generalized subproblem-dependent heuristic (GSH). In their method, the d-DPAP is decomposed into a number of scalar subproblems, and the subproblems are optimized in parallel by using neighborhoods information and problem-specific knowledge. But this method does not take full account of the dynamic of the nodes.

The algorithms DRNG and DLMST were proposed in [6], and they are based on the proximity graph theory. In both algorithms, each node collects the information of the surrounding neighbors and then determines their own transmit power to ensure the connectivity of network. Chen et al. [7] proposed the SPAN protocol in which each node can sleep if its two random neighbors can communicate with each other; otherwise the node must be in working state. In the ASCENT [8] protocol of Cerpa and Estrin, the nodes can determine the working state of their own according to the detected local communication situation and decide the transmit power according to the packet loss rate.

Glauche et al. [9] defined critical node degree as the required size of the node degree that can keep network full connectivity. On the account that they proposed a distributed algorithm to ensure that the ad hoc network can communicate, the algorithm adjusts nodes' communication radius to guarantee the satisfaction of the critical node degrees. Jiang and Bruck [10] controlled the network topology by adjusting the radius of nodes communication and proposed an algorithm to ensure that any source node can communicate with the sink node. But in this paper, the algorithm does not take account of the energy savings and the network lifetime. Liqun et al. [11] proposed a power control method based on continuous time slot scheduling and proved the NP-completeness of it. This method requires the network to maintain strict time synchronization, and the fairness between the nodes is not considered. The application-oriented fault detection and recovery algorithm (AFDR) [12] is mainly to limit the impact of critical node failure on coverage and connectivity in wireless sensor and actor networks (WSANS). The algorithm needs a moving backup actor to detect the critical node failure, and then the recovery process is initiated, yet the backup actor in MSR-WSN is not practicable because sensor nodes cannot move by themselves and cannot be replaced.

All of these researches are very helpful for the topology control of our MSR-WSN, but they are mainly for static wireless sensor networks and have a poor adaptability for MSR-WSN. For instance, the methods in [4, 7] could extend the lifetime of the network, but sleeping mechanisms is not suitable for MSR-WSN. Therefore, it is necessary to carry out specific research related to the two requirements that are higher dynamic adaptability and longer lifetime of MSR-WSN.

3. Model and Algorithm Design

3.1. System Model. In this section, we present our network model and then formulate the problem of energy dynamic distribution and optimization of nodes in the MSR-WSN.

In the MSR-WSN, there have been a certain number of homogeneous sensor nodes and a small amount of sink nodes. The sensor nodes are placed in lifejackets with limited energy, and the number of them is N. The sink nodes are placed in survival crafts with unlimited energy, and they can move by themselves. Sensor nodes can be organized into a network and identify the sink node which has entered the convergence range of the network. The sensor nodes transfer

data to the sink node after identifying the sink node. The sensor nodes are responsible for regularly transmitting the information, such as nodes' location or monitoring data, to the sink node. Each sensor node communicates to the sink node directly or in the form of a multihop. The network topology dynamically changes as the movement of sensor nodes and sink nodes. Then the energy distribution should change with the topology, so that the energy can be saved, and consequently the lifetime of network can be extended.

We assume a perfect medium access control, such as SMAC [13], which ensures that there are no collisions at any sensor during data communication, and we adopt the simple but relevant path loss communication model as in [14]. In this model, the transmit power level that should be assigned to a sensor i to reach a sensor j is $P_i = \beta \times d_{ij}^{\alpha}$, where $\alpha \in [2, 6]$ is the path loss exponent and $\beta = 1$ is the transmission quality parameter. The energy loss due to channel transmission is d_{ij}^{α}, d_{ij} is the Euclidean distance between sensors i and j, and $R_C^i = d_{ij}$ is sensor i's communication range, s.t. $R_C^i \leq R_{\max}$, where $R_{\max}$ is a fixed maximum communication distance, which is constrained by the maximum power that sensors can transmit, that is, $P_{\max}$.

The residual energy of sensor i, at time t, is calculated as follows:

$$E_i(t) = E_i(t-1) - \left[\left(r_i(t) + 1\right) \times P_i \times \text{amp}\right], \quad (1)$$

where $(r_i(t) + 1)$ is the total traffic load that sensor i forwards towards H at t, $r_i(t)$ is the traffic load that i receives and relays, "+1" is the data packet generated by i to forward its own data information, and amp is the power amplifier's energy consumption. In EDDO, we assume that the region A is large and N is relatively small. Consequently, the sensors should spread and communicate through long transmission distances. Thus, the energy consumed by the transceiver electronics is negligible and can be ignored because the transmit power is the main factor on sensors' total energy consumption [15, 16].

3.2. Problem Formulation. The problem of energy distribution and optimization is given as follows:

(i) A: a 2D plane, where the sensor nodes are distributed;

(ii) N: the number of the sensor nodes in region A;

(iii) E: the initial energy of sensor nodes;

(iv) R_c: the maximum communication distance of the sensor nodes;

(v) $P_{\max}$: the maximum emission energy levels.

Decision variables of a network design X as follows:

(i) (x_j, y_j): the location of sensor j;

(ii) P_j: the transmission power level of sensor j.

Objectives are to maximize connectivity and minimize energy consumption of the network.

The network connectivity is defined as the fact that each node can communicate with the sink node directly or indirectly, and the network energy consumption is defined as the sum of all sensor nodes' emission energy.

Connectivity: for any $c_{jH} \in C^{(N)}, 1 \leq j \leq N$, H is the sink node, there is $c_{jH} = 1$, $C^{(N)}$ is the connectivity matrix of the network, and c_{jH} is the connectivity status of a sensor j, which is denoted as:

$$c_{jH} = \begin{cases} 1, & \text{if } j \text{ is connected;} \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where sensor j directly communicates with H, or if it sustains some neighbors with positive advance towards H (i.e., neighbors are closer to H than j [17]), considering the many-to-one communication nature of WSNs as follows:

$$\text{Energy consumption} : E_{\min} = \min\left\{P_1 + P_2 + \cdots + P_j + \cdots P_N\right\}. \quad (3)$$

During the process of EDDO, $c_{jH} = 0$ represents that sensor j is disconnected, then the transmit power P_{ij} of j should automaticly convert to $P_{(i+1)j}$. This process is repeated until $c_{jH} = 1$ or $i = n$. If $i = n$ and $c_{jH} = 0$, it means that sensor j is not within the scope of cover. In this case, sensor j should send information by the transmit power P_n at intervals until $c_{jH} = 1$.

3.3. Solution Representation and Ordering. In this paper, a candidate solution X consists of N items which corresponding to the N sensor nodes. The jth item of X has two parts, which represent the location (x_j, y_j) and the transmit power P_j of sensor j, respectively.

The location (x_j, y_j) is assumed that has been obtained by the GPS, and it will change with the environmental conditions. The moving of some nodes may lead to the transmit power to be reallocated. The location information in solution X will affect the local topology of node j, and it can help the objective function to evaluate or select a new solution. It mainly depends on the value of P_{ij} that whether X is the optimal solution, the nth transmit power of sensors is the selection content of an operator, P_{ij} is the ith transmit power of sensor j, and it belongs to n items which are from P_0 to P_n, $P_0 = 0$, $P_n = P_{\max}$. $P_0 = 0$ means that sensor j is disconnected. Dividing the transmit power into n levels is in order to operate on the genetic coding conveniently.

3.4. The Energy Distribution Based on Genetic Algorithm. In recent years, genetic algorithm (GA) is a popular optimized method which has a parallel search and group optimization features. GA is a kind of bionic algorithm; this type of algorithms can solve many complex engineering problems without the mathematical properties of them.

3.4.1. The Algorithm Coding Mapping and Population Initialization. According to the real-coded ideological of K. H. Jin, each individual represents a power allocation scheme of MSR-WSN. The genome of the population of individuals is

$$g = \{C_1, C_2, \ldots, C_N\}, \quad (4)$$

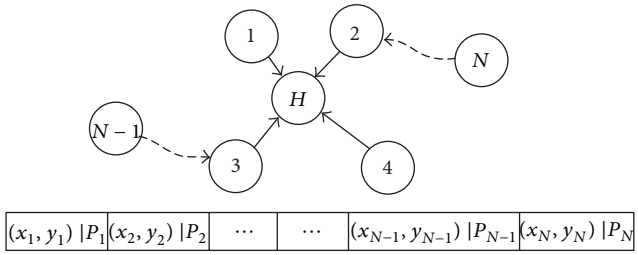


Figure 1: Initial solutions X of N individuals (Pareto charts).

where chromosome $C_j = P_j$, and P_j is the transmission power of the sensor node j. If $C_j = 0$, then the chromosome is empty which represents that sensor j is disconnected. The genome can not only indicate node power allocation scheme, but also indicate the number of the activity sensor nodes in the communication network.

$x_j(t)$ is the jth individual in the population t, and $x_j(t)$ contains n genes as follows:

$$x_j(t) = \left(x_{1j}(t), x_{2j}(t), \ldots, x_{nj}(t)\right), \tag{5}$$

where $j = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, t_{\max}$. n is the number of the chromosome in the individual and the number of variables in the vector, N is the population size, $t_{\max}$ is the maximum evolution generation, and gene $x_{ij}(t)$ corresponds to P_{ij}.

During the population initialization process, each node determines its own transmit power according to the distance of the sensor and H or the distance between sensors. Firstly, the sensors are sorted based on their distance to H such as Algorithm 1 referenced [18], where 1 is the closest and N is the farthest sensor location with respect to H, respectively. X is showed in Figure 1, where the transmission power of each of the node j is proportional to the transmitting radius R_C^j, $R_C^j \le R_{\max}$. R_C^j needs to meet the condition that sensor j can connect to its closest node k $(k < j)$, and the X is initially generated as follows.

Algorithm 1. The representation process for each solution Y

Input: A solution Y;

Output: A solution X;

Step 1: Calculate the dense-to-spread ordering of Y to get X;

Step 2: for each (x_j, y_j) in X do

$$P_{ij} = \begin{cases} \left(d_{j,H}\right)^{\alpha}, \\ \quad \text{if } (x_H, y_H) \text{ is } j\text{'s closest location, } d_{j,H} \le R_{\max} \\ \left(d_{j,k}\right)^{\alpha}, \\ \quad \text{if } (x_k, y_k) \text{ is } j\text{'s closest location, } k < j, P_k \ne 0, \\ \quad d_{j,k} \le R_{\max}. \end{cases} \tag{6}$$

This process of topology generating is simple and convenient, but it does not have the dynamic adaptability and is not conducive to the followup topology control. In MAR-WSN, the sink node H can move freely, and the sensor nodes will move with water waves. Therefore, the distribution of node power should be optimized constantly as the time goes by, so that the energy consumption level of each node can adapt to the network well.

3.4.2. The Selection of the Objective Function. For the given solution, we need to accurately estimate its quality (fitness value), which is determined by the fitness function. In this paper, we aim at finding a solution which ensures the nodes that can communicate with H with as little as possible of the energy consumption. The quality of solution $X(t)$ mainly depends on the connectivity and energy consumption.

$F(C_j)$ is the fitness value of chromosome C_j which represents the transmit energy level of sensor j, and $F(C_j)$ is defined as the ratio of sensor j's transmit power and the total power of all nodes, as follows:

$$F\left(C_j\right) = \frac{P_j}{\sum_{j=1}^{N} P_j}. \tag{7}$$

In addition, we made the value as the comparison reference which is the ratio of the average power of all nodes and the total transmits power. The comparison reference is calculated and simplified as follows:

$$\frac{\left(\sum_{j=1}^{N} P_j\right)/N}{\sum_{j=1}^{N} P_j} = \frac{1}{N}. \tag{8}$$

The fitness function can select out the nodes whose transmit power is less than the average power in the network. Then the selected nodes will be the cross-object nodes which instead of those nodes whose transmit power is more than average power. By this way, the nodes whose transmit power is overload will be optimized, and they will not be dead quickly.

3.4.3. Selection Mechanism. Select operation can improve the average quality of the population by selecting high-quality chromosome into the next generation of the population. Chromosome selection is based on the fitness value. Selection operation selects the chromosomes according to the constraint environment of each individual (that are the distances between the sensor and its neighbors), the position dynamically change information, and the fitness of X as follows:

When $F(C_j) > 1/N$, then the C_j will be eliminated.

When $F(C_j) \le 1/N$, then C_j will be the cross-object.

3.4.4. Crossover and Evolution. The genetic algorithm is based on two basic genetic operation that are cross-operation and evolution operation. The cross-operation handles the current solution in order to find a better solution. The evolution operation can help the GA to avoid falling into local optimal solution [8]. The effect of the genetic algorithm depends on these two steps. In this paper, we crossed the genes

$x_{1j}(t), x_{2j}(t), \ldots, x_{nj}(t)$ on the chromosome when the chromosome C_j needs to be evaluated. The cross-objects come from the solution which is selected by the selection operation and is the immigration solution selected (Section 3.5). Then we crossed the binary coding that corresponds to the power. We adopted single-point crossover, and the new chromosome is updated.

The updating formula after crossover operation is such as

$$x_{ij}(t+1) = \begin{cases} h_{ij}(t+1), & f\left(h_{ij}(t+1) < f\left(x_{ij}(t)\right)\right) \\ x_{ij}(t), & f\left(h_{ij}(t+1) \geq f\left(x_{ij}(t)\right)\right), \end{cases} \quad (9)$$

where $h_{ij}(t+1)$ is the new chromosome after the crossover operation.

The new chromosome should be evaluated that it could communicate with H, after that the updating operation can be performed. The evaluation processed as the Algorithm 2.

Algorithm 2. The updating process

Input: The new chromosome $h_{ij}(t+1)$;

Output: The external chromosome in this generation.

Step 1: Calculate the connectivity status of a sensor j.

Step 1.1: Generate the $C^{(N)}$ by the new P_{ij} of $h_{ij}(t+1)$.

Step 1.2: Return c_{jH}

Step 2: If $c_{jH} = 1$.

3.5. The Energy Dynamic Distribution and Optimization Based on GA. For dynamic optimization problem, the GAs should maintain a certain level of solution diversity to keep the solution adapting to the changes in the environment; then the convergence problem becomes very important. To address this problem, the random immigrants approach is a quite natural and simple way [19, 20]. It was proposed by Grefenstette with the inspiration from the flux of immigrants that wander in and out of a population between two generations in nature. It maintains the diversity level of the population through replacing some individuals of the current population with random individuals, called random immigrants, every generation. As to which individuals in the population should be replaced, usually there are two strategies: replacing random individuals or replacing the worst ones [21]. In order to avoid that random immigrants disrupt the ongoing search progress too much, especially during the period when the environment does not change, the ratio of the number of random immigrants to the population size is usually set to a small value, for example, 0.2.

However, if the environment of nodes changes slowly, the introduced random immigrants may divert the searching force of the GA and hence may degrade the performance. On the other hand, if the environment of nodes only changes slightly in terms of severity of changes, random immigrants may not have any actual effect even when a change occurs because individuals in the previous environment may still be quite fit in the new environment. Based on the above considerations, an immigrants approach, called elitism-based immigrants [22], is proposed for GAs to address the energy deployment problem.

For each generation t, after the normal genetic operations (i.e., selection and recombination), the elite $E(t-1)$ from previous generation is used as the base to create immigrants. From $E(t-1)$, a set of $r_{\text{ei}} \times N$ individuals are iteratively generated by mutating $E(t-1)$ with a probability p_m^i, where N is the population size and r_{ei} is the ratio of the number of elitism-based immigrants to the population size. The generated individuals then act as immigrants and replace the worst individuals in the current population. It can be seen that the elitism-based immigrants' scheme combines the idea of elitism with traditional random immigrants' scheme.

The whole energy dynamic distribution and optimization (EDDO) algorithm is described as Algorithm 3. In the genetic operation of Algorithm 3, if the mutation probability p_m^i is satisfied, the elite $E(t-1)$ will be used to generate the new immigrants by the mutation operation; otherwise, $E(t-1)$ itself will be directly used as the new immigrants. It uses the elite from previous population to guide the immigrants toward the current environment, which is expected to improve GA's performance in dynamic environments.

Algorithm 3. The energy deployment general framework

Input: network parameters $(A, N, n, E, P_{\max}, t_{\max})$;

Output: the external population (EP)

Step 0-Setup: Set EP := Φ; $t := 0$; $\text{IP}_t := \Phi$;

Step 1-Initialization: Generate an initial solution as Algorithm 1: $\text{IP}_0 = \{X_1, \ldots, X_N\}$;

Step 2: For $i = 1, \ldots, N$ do

Step 2.1-Genetic Operators: Generate a new solution Y using the genetic operators.

Step 2.2-Perform elitism-based immigration: Generate $r_{ei} \times N$ immigrants by mutating $E(t-1)$ with p_m^i.

Step 2.3-Update Populations: Update the population as Algorithm 2.

Step 3-Stopping criterion: If stopping criterion is satisfied, i.e.

4. Simulation and Analysis

4.1. Simulation Setup. In the environment of MSR-WSN, we designed the region A which is 1000 m × 1000 m, and 100 sensor nodes are randomly distributed in the region. The medium access control, such as SMAC, is introduced in Section 3.1, and its parameters were set according to [5]. In our experimental studies, the parameters of algorithm are set as in [9]. That is, the number of power levels $n = 40$, max number of generations $t_{\max} = 300$, crossover rate $r_c = 0.9$, mutation rate $r_m = 0.5$, tournament size $M = 10$, and neighborhoods size is 2. Moreover, in all simulation studies

the following network parameters are set [23]: $R_s/R_{\max} = 100/200, E = 10\,\text{J}, d_{\min} = 100\,\text{m}, a = 2$, and amp $= 100\,\text{pJ/bit/m}^2$.

During the experiment, to imitate the dynamic nature of the actual environment, we made the nodes move randomly like

$$x(T+1) = x(T) + \eta\cos\theta; y(T+1) = y(T) + \eta\sin\theta, \quad (10)$$

where η, θ were randomly generated number and $0 \le \eta \le 1$, $0 \le \theta \le 2\pi$, T represented time and it's unit was second.

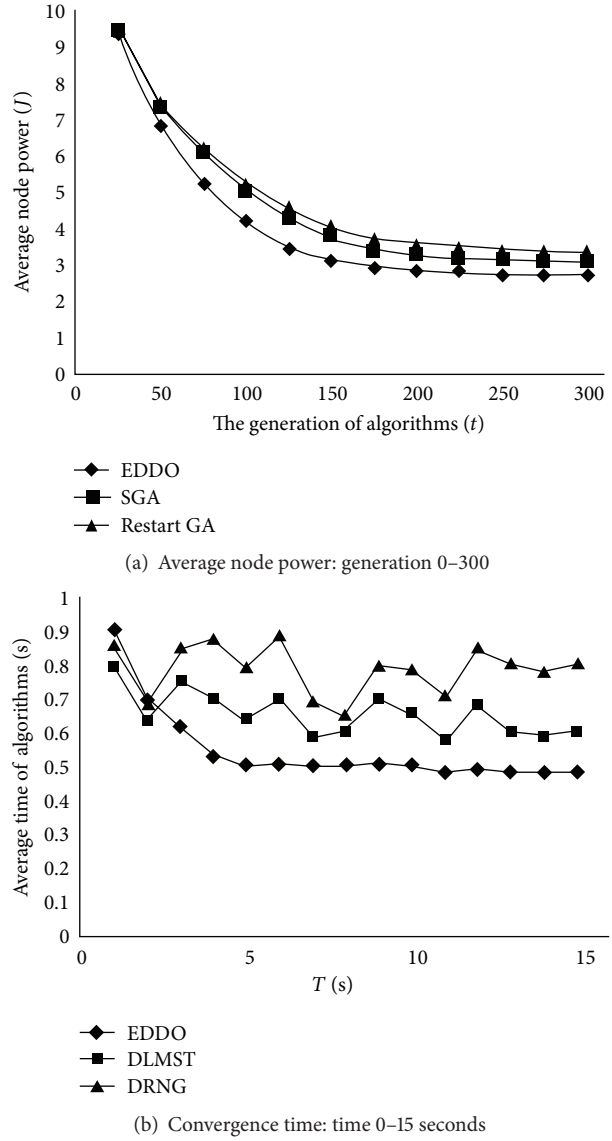
4.2. Simulation Scheme. The simulation experiments were realized through MATLAB 7.0 to verify feasibility and validity of the proposed algorithm.

We compared the EDDO algorithms with SGA and Restart GA at first as in Section 4.2.1. After that we compared the EDDO with other algorithms about energy as DRNG and DLMST in Section 4.2.1.

4.2.1. Comparisons with SGA and Restart GA. Figure 2(a) shows that average node power obtained from the proposed EDDO algorithm, the SGA, and Restart GA. We can see that with the increasing of the population generation, all the algorithms could achieve the convergence solution. But the EDDO algorithm is faster than the SGA and Restart GA because of its elite scheme. And the final average node power of the algorithm is lower than that of SGA by about 0.3 J in one convergence.

Figure 2(b) shows that the convergence time of the EDDO algorithm would be much lower than the time of SGA and the Restart GA. It indicates that the EDDO algorithm can adapt to the dynamic movement of nodes because the algorithm makes full use of the excellent solution in the original environment and the immigrant also contributes to the convergence speed. With time goes by, the convergence time of the algorithm becomes relatively stable and is lower than the SGA by about 13.7%.

4.2.2. Comparisons with Other Algorithms. Figure 3 shows the comparison with the DLMST and DRNG algorithms [4] in which every sensor determines its power based on the information of neighbors. We can see that the average power of three algorithms is almost the same in Figure 3(a) and the average power of the whole time of the EDDO algorithm is a little higher than the other two algorithms. However, the running time of these three algorithms is different in Figure 3(b) and the EDDO algorithm is the best one here. It dues to the algorithm in this paper can seek the optimization solution overall the nodes, while in the DLMST and DRNG every node takes account of itself only. The local information could help DLMST and DRNG to adapt the dynamic change of networks in a certain extent, but it is not very helpful for the whole network. And the two algorithms' execution time is not so stable than that in the EDDO algorithm.



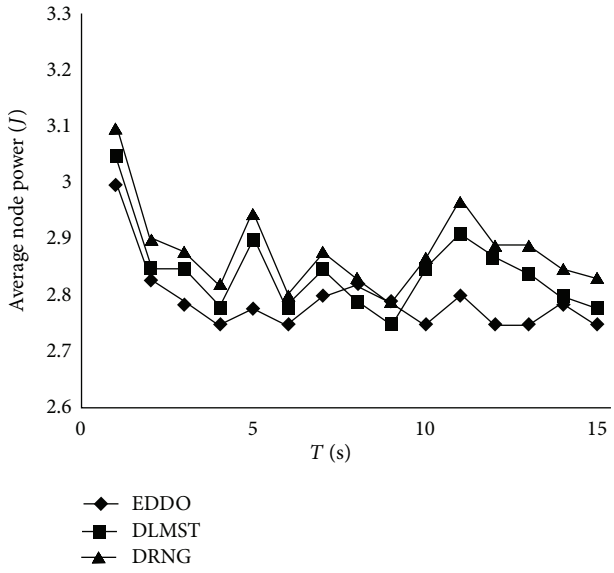
(a) Average node power: generation 0–300

(b) Convergence time: time 0–15 seconds

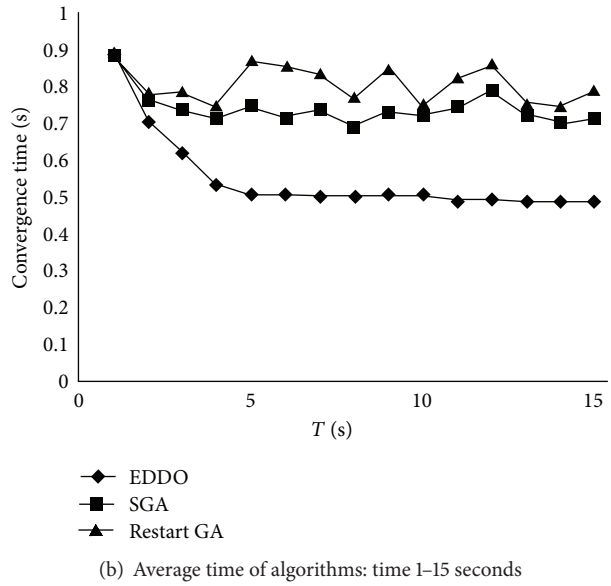
Figure 2: Comparisons with SGA and Restart GA.

5. Conclusion

In this paper, the energy distribution problem of MSR-WSN is defined as a deployment solution that should take account of the connectivity and the lifetime of the network. The problem is modeled in the environment at sea, and the appliance process of the network has been described at first. In the procedure of solving the problem, the dynamic nature and energy consumption of the nodes is focused on in the harsh environment of sea. Introducing the genetic algorithms with elitism-based immigrants, we made the energy distribution method adapt to the dynamic change of the nodes position and convergent more quickly. In the future, we will study a more efficient and perfect topology control method of the MSR-WSN.



(a) Average node power: time 1–15 seconds



(b) Average time of algorithms: time 1–15 seconds

Figure 3: Comparisons with DRNG and DLMST algorithms.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (51279099), the Shanghai Natural Science Foundation (12ZR1412500), the Innovation Program of Shanghai Municipal Education Commission (13ZZ124), and the "Shu Guang" Project supported by the Shanghai Municipal Education Commission and Shanghai Education Development Foundation (12SG40).

References

[1] R. Xin and Z. Yongxin, "Incremental tree induction for detection of the rescue target in the marine casualty," in *Proceedings of the WRI Global Congress on Intelligent Systems (GCIS '09)*, vol. 9, pp. 432–435, IEEE Computer Society, Washington, DC, USA, May 2009.

[2] J. Peng and C. Shi, "Remote sensing application in the maritime search and Rescue," in *Remote Sensing*, vol. 2, pp. 1–24, 2012.

[3] J. J. Kim, D. J. Kim, J. D. Kim, C. H. Hong, B. B. Lee, and K. N. Ko, "The device for generation the distress signal and monitoring system for a survivor based on WSN," in *Proceedings of the International Conference on Electronics and Information Engineering (ICEIE '10)*, pp. V1-81–V1-86, August 2010.

[4] Z. Zhan, J. Zhang, and Z. Fan, "Solving the optimal coverage problem in wireless sensor networks using evolutionary computation algorithms," in *Proceedings of the 8th International Conference on Simulated Evolution and Learning*, pp. 166–176, Seal, 2010.

[5] A. Konstantinidis and K. Yang, "Multi-objective energy-efficient dense deployment in Wireless Sensor Networks using a hybrid problem-specific MOEA/D," *Applied Soft Computing Journal*, vol. 11, no. 6, pp. 4117–4134, 2011.

[6] N. Li and J. C. Hou, "Topology control in heterogeneous wireless networks: problems and solutions," in *Proceedings of the 23rd Annual Conference of the IEEE Computer and Communications Societies*, pp. 232–243, IEEE Computer Society, Hong Kong, China, 2004.

[7] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2002.

[8] A. Cerpa and D. Estrin, "ASCENT: adaptive self-configuring sensor networks topologies," *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 272–285, 2004.

[9] I. Glauche, W. Krause, R. Sollacher, and M. Greiner, "Continuum percolation of wireless ad hoc communication networks," *Physica A*, vol. 325, no. 3-4, pp. 577–600, 2003.

[10] A. Jiang and J. Bruck, "Monotone percolation and the topology control of wireless networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 327–338, Miami, Fla, USA, March 2005.

[11] F. Liqun, S. C. Liew, and H. Jianwei, "Power controlled scheduling with consecutive transmission constraints: complexity analysis and algorithm design," in *Proceedings of the 26th Annual IEEE Conference on Computer Communications (INFOCOM '07)*, pp. 1530–1538, IEEE, 2009.

[12] J. Du, L. Xie, X. Sun, and R. Zheng, "Application-oriented fault detection and recovery algorithm for wireless sensor and actor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 273792, 9 pages, 2012.

[13] S. Toumpis, "Mother nature knows best: a survey of recent results on wireless networks based on analogies with physics," *Computer Networks*, vol. 52, no. 2, pp. 360–383, 2008.

[14] S. Toumpis and L. Tassiulas, "Optimal deployment of large wireless sensor networks," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2935–2953, 2006.

[15] X. Liu and P. Mohapatra, "On the deployment of wireless data back-haul networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1426–1435, 2007.

[16] Y. Chen, C. N. Chuah, and Q. Zhao, "Network configuration for optimal utilization efficiency of wireless sensor networks," *Ad Hoc Networks*, vol. 6, no. 1, pp. 92–107, 2008.

[17] T. Melodia, D. Pompili, and I. F. Akyildiz, "On the interdependence of distributed topology control and geographical routing

in ad hoc and sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 520–531, 2005.

[18] A. Konstantinidis, K. Yang, Q. Zhang, and D. Zeinalipour-Yazti, "A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks," *Computer Networks*, vol. 54, no. 6, pp. 960–976, 2010.

[19] J. J. Grefenstette, "Genetic algorithms for changing environments," in *Proceedings of the 2nd International Conference on Parallel Problem Solving from Nature*, pp. 137–144, Brussels, Belgium, September 1992.

[20] H. G. Cobb and J. J. Grefenstette, "Genetic algorithms for tracking changing environments," in *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 523–530, Urbana-Champaign, Ill, USA, June 1993.

[21] F. Vavak and T. C. Fogarty, "A comparative study of steady state and generational genetic algorithms for use in nonstationary environments," in *Proceedings of the AISB Workshop on Evolutionary Computing*, pp. 297–304, Brighton, UK, April 1996.

[22] H. Cheng and S. Yang, "Genetic algorithms with elitism-based immigrants for dynamic shortest path problem in mobile ad hoc networks," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 3135–3140, May 2009.

[23] M. Cardei, M. O. Pervaiz, and I. Cardei, "Energy-efficient range assignment in heterogeneous wireless sensor networks," in *Proceedings of the 2nd International Conference on Wireless and Mobile Communications (ICWMC '06)*, pp. 1–11, July 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 109307, 13 pages
http://dx.doi.org/10.1155/2013/109307

Research Article

A Hybrid Energy- and Time-Driven Cluster Head Rotation Strategy for Distributed Wireless Sensor Networks

Guoxi Ma and Zhengsu Tao

Department of Electronic, Information and Electrical Engineering, Shanghai Jiaotong University, No. 800 Dongchuan Road, Shanghai 200240, China

Correspondence should be addressed to Zhengsu Tao; zstao@sjtu.edu.cn

Received 4 October 2012; Revised 15 December 2012; Accepted 3 January 2013

Academic Editor: Wenzhong Li

Clustering provides an effective way to extend the lifetime and improve the energy efficiency of wireless sensor networks (WSNs). However, the cluster heads will deplete energy faster than cluster members due to the additional tasks of information collection and transmission. The cluster head rotation among sensors is adopted to solve this problem. Cluster head rotation strategies can be generally divided into two categories: time-driven strategy and energy-driven strategy. The time-driven strategy can balance energy consumption better, but it is not suitable for heterogonous WSNs. The energy-driven cluster head rotation strategy has high energy efficiency, especially in heterogonous networks. However, the rotation will become increasingly frequent with the reduction of the nodes residual energy for this strategy, which causes lots of energy waste. In this paper, we propose a hybrid cluster head rotation strategy which combines the advantages of both energy-driven and time-driven cluster head rotation strategies. In our hybrid rotation strategy, the time-driven strategy or energy-driven strategy will be selected according to the residual energy. Simulations show that the hybrid strategy can enhance the energy efficiency and prolong network lifetime in both homogeneous and heterogeneous networks, compared with either single time-driven or energy-driven cluster head rotation method.

1. Introduction

The advances in MEMS-based sensor technology and wireless communications recently have enabled the development of low-cost, low-power, multifunctional sensor nodes that are in small size and have short communication distance and low computational ability in wireless sensor networks (WSNs). These small sensor nodes are capable of sensing the environment, storing and processing the collected sensor data, and interacting and collaborating with each other within the network [1]. The major challenge in the design of WSNs is the energy management. Due to the strict energy constraint and nonrechargeable energy provision, the energy resource of sensors should be managed wisely to extend the lifetime of networks. Therefore, much attention has been paid to develop low-power hardware design, collaborative signal processing techniques, and energy-efficient algorithms at various WSNs [2, 3].

Neighboring sensor nodes usually have the data of similar events because they collect events within a specific area. If each node transmits the collected data to the sink individually, lots of energy will be wasted for transmitting similar data. In order to achieve high energy efficiency and extend the network lifetime, sensors are often hierarchically organized into clusters, each of which has its own cluster head (CH). Within each cluster, sensors transmit data to their CH over a relatively short distance, which in turn forwards the data (or it is further aggregated) to the sink via a single-hop or a multihop path through other CHs like the illustration in Figure 1. In hierarchical clustering protocol, network is operated based on rounds. Each round begins with the cluster set-up phase and follows with a steady-state phase. In cluster set-up phase, sensor nodes elect themselves to be CHs at any given time with a certain probability and local cluster network is formed by each CH. In steady-state phase, each noncluster head node, which we call a cluster member, collects events and transmits them to its cluster head. After that, each cluster head aggregates the collected data to prevent duplicated transmissions of similar events and transmits the aggregated data to the sink node.

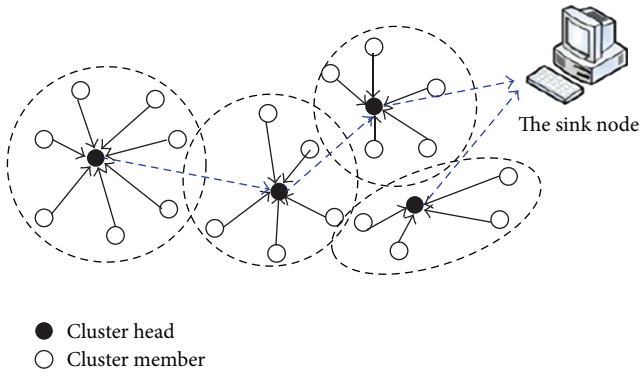


Figure 1: Clustering scheme of wireless sensor network.

From the description above, we can see that CHs undertake much heavier responsibilities than cluster members including long distance communication to sink node, intracluster data collection, and aggregation. Therefore, CHs consume much more energy than cluster members, and the fast energy depletion problem of them has become a serious issue in the cluster-based sensor networks. In traditional clustering schemes, powerful backbone nodes are proposed to serve as the cluster heads [4], which may increase the system deployment cost and is not suitable for many scenarios, for example, the intrusion detection and large-scale sensor networks which are often automatically deployed by the vehicles or planes. In such cases, the cluster heads are tiny and unattendant battery-powered sensors, which require innovative design techniques to ensure high energy efficiency. The only way to prolong the lifetime of CHs lies in organization method. Then, CHs rotation strategy has been exploited to balance the energy consumption among cluster members and heads in hierarchical clustering WSNs.

The CHs rotation strategy is a technology of cluster topology maintenance. It plays a very important role in the reconstruction of the hierarchical cluster structure for avoiding premature death of CH nodes and increasing network lifetime. Many cluster head rotation strategies have been proposed in the recent research literatures. Such strategies can be divided into two categories: time-driven strategy and energy-driven strategy. Heinzelman et al. [5] firstly proposed LEACH cluster protocol in which cluster head is rotated periodically among the sensor nodes to ensure the balanced energy consumption of networks. By such periodical rotation of cluster heads among the sensors, energy is evenly depleted among nodes and the premature exhaustion of battery in any sensor can be avoided. But the time-driven based CH rotation method doesnot consider the residual energy of cluster heads which will cause much unnecessary energy waste and low energy efficiency when the network energy is powerful, especially in heterogeneous WSNs. However, time-driven CH rotation method can maintain fixed energy efficiency at any time.

To better solve this problem, the CH rotation mechanism based on energy driven is proposed in EDAC [6] and EDCR [7, 8], where the cluster head candidacy election phase and rotation phases are triggered only when the residual energy of any cluster head falls below a dynamic threshold. This kind of energy-driven CH rotation strategy can mitigate the impact of the cluster head rotation on the network and achieve high energy efficiency in heterogeneous WSNs. But under this rotation mechanism, the cluster head rotation will become increasingly frequent along with the reduction of residual energy of the nodes. As a result, more and more energy will be used in rotation rather than data transmission, which will in turn reduce the energy efficiency of the network even to zero.

We notice that the residual energy of nodes will become almost equal in heterogeneous WSNs after a long period of network operation. The heterogeneous WSNs will become nearly the same as the homogeneous WSNs in terms of residual energy. Based on the respective features of time-driven and energy-driven CH rotation strategies above, we can develop new CHs rotation approach to achieve better energy balance and higher energy efficiency. We can use the energy-driven CH rotation strategy to achieve high energy efficiency when the energy of WSNs is abundant and unbalanced. After a long time of network operation, the energy of WSNs nodes is low and almost balanced, therefore; we adopt time-driven CH rotation strategy to ensure stable data transmission efficiency. This is the theoretical basis of our research. In this paper, we build the energy consumption model for contention-based CH selection and data transmission. By analyzing the energy consumption relationship between the cluster set-up phase and the steady-state data transfer phase, we propose a hybrid cluster head rotation strategy for WSNs in which the CH rotation strategy based on time or energy will be selected according to the network energy efficiency. We give the optimal mechanism that when we switch from energy-driven CH rotation to time-driven CH rotation to obtain the maximal energy efficiency for WSNs. Theoretical analysis and simulation results show that hybrid CH rotation strategy can successfully balance the energy consumption of the network and prolong the network lifetime. Moreover, it can obtain better energy efficiency and transfer more useful data to sink than either time-driven or energy-driven CH rotation method in both homogeneous and heterogeneous sensor networks. As a fundamental technology of cluster topology maintenance, our rotation strategy can be used in any cluster protocols.

The remainder of the paper is organized as below. Section 2 reviews research work in this area; Section 3 introduces the system assumptions used in this paper; Section 4 establishes and analyzes energy consumption for contention-based CHs selection and data transmission. Section 5 puts forward hybrid energy- and time-driven CH rotation strategy; Section 6 elaborates on our simulation efforts and the analysis results we obtained; Section 7 offers concluding remarks and points out future directions for research work.

2. Literature Review

As an important technology of cluster topology maintenance, cluster head rotation strategy has a great signification for the performance of cluster protocol in WSNs. By restoring the current topology and building a new one, it prolongs the network lifetime and prevents the premature death of sensors. However, it doesnot attract enough attention and extensive

research work is only dedicated to the study of cluster organization and scheduling algorithms. In this section, we briefly describe the most popular research that is most relevant to our approach.

Clustering provides an effective way for extending the lifetime of WSNs. Experiments have proved that the networks based on cluster hierarchy structure protocol are 7~8 times more efficient than traditional dispersed flat nodes protocol in measuring the lifetime of nodes. So, more efficient hierarchical network algorithms based on cluster architecture are designed to extend the network lifetime. Heinzelman et al. [5] proposed the LEACH algorithm, in which the CH is dynamically rotated among all sensors in the cluster for a period of time. Although there are advantages in using this distributed cluster algorithm, it doesnot consider the residual energy of sensor nodes and creates lots of overhead frames. Heinzelman also proposed the improved LEACH named LEACH-C algorithm [5] that elects cluster head based on the residual energy of sensor nodes.

SEP [9] extends the LEACH by adding a small percentage of higher energy nodes than the normal nodes in the network. HEED [10] elects CHs according to their total residual energy and a secondary parameter such as node degree. It has a much higher overhead compared with other algorithms because of CHs selection and rotation. The energy-efficient clustering method proposed in ANTCLUST [11] assumes that nodes sense their locations either using GPS or some localization techniques. Such an assumption is suitable for location-based information gathering systems but less applicable to low-cost ad hoc sensor networks. ACE [12] is an emergent algorithm to produce uniform clusters. Meanwhile, many other cluster-based hierarchy algorithms are improved based on LEACH [13–20]. All of the above algorithms share a common feature that their CH rotation mechanisms are all time driven. That is, the CH will be changed after a predetermined number of data gathering rounds. EDAC [6] and EDCR [7, 8] algorithms use method based on dynamic energy threshold to initiate a new CH election phase. This energy driven CH rotation strategy can achieve better energy efficiency and produce less overhead, especially in heterogeneous sensor networks.

Both energy-driven and time-driven CH rotations are faced with an urgent issue that how to select the optimum number of data transmission rounds. If a CH election phase is triggered after a smaller number of data transmission rounds, it will result in excessive overhead during the CH election phase. On the other hand, after a large number of data transmission rounds, the CH nodes will not have enough energy to act as ordinary sensor nodes after relinquishing the CH role. In [21], the authors proved that the energy-driven CH rotation strategy is better than the time-driven CH rotation strategy in a single cluster analysis model, and they gave a suboptimal solution for rotation energy threshold in the single cluster model. In [22], authors provided a method to obtain the optimal energy threshold parameter for energy-driven CH rotation strategy. For our hybrid CH rotation strategy, the main challenge lies in the mechanism how to select the best CH rotation strategy, energy driven or time driven, to obtain the maximum energy efficiency and extend the network lifetime. In the latter part, we will study it theoretically and analyze the simulation result, which constitutes the major part of this paper.

3. Preliminaries

Before describing our proposed algorithm in detail, we will introduce the characteristics of the network model used in our implementations. We consider a WSN consisting of sensors and sink and make the following assumptions.

3.1. Assumptions on Node and Energy of the Network. As Figure 2 shows, we consider a sensor network of N nodes which is randomly dispersed in an $M \times M$ meters square area to continuously monitor the sense area, and all nodes including sink are stationary after disposition. Our assumptions about the sensor nodes and network are as below.

(1) Nodes are equipped with wireless transmission and reception equipment and they have the capability of data fusion and power adjusting.

(2) Time Division Multiple Access (TDMA) schedules the data transmission from normal nodes to its cluster head.

(3) Nodes can estimate the approximate distance to another node based on the received signal strength.

(4) Links between nodes are symmetric.

(5) The network runs a periodic data gathering application, in which the sensor generates traffic at an average rate of λ bits/second and sends it to its CH, which in turn delivers it to sink.

3.2. Energy Consumption Model. A typical sensor node includes three basic units: sensing unit, processing unit, and transceivers. For our energy model of communication and transmission scheme, we assume a free space propagation channel model [23]. We ignore the power consumption of node for sensing because it is constant at any time and cannot be reduced with whatever means. Therefore, the energy model of a sensor includes the power for data aggregating, data receiving, and data transmission, in accordance with the radio hardware energy dissipation, both the free space (d^2 Power loss) and the multipath fading (d^4 Power loss) channel models. If the distance is less than the fading threshold d_0, the free space model is used; otherwise, the multipath model is used. If E_{Tx} denotes the energy consumption of transmitting, E_{Rx} denotes the energy consumption of receiving, and E_{Dx} denotes the energy consumption of aggregating, the energy for transmitting, receiving, and aggregating l bits over distance d is computed as follows:

$$E_{\mathrm{Tx}} = \begin{cases} \left(E_{\mathrm{elec}} + \varepsilon_{\mathrm{fs}} * d^2\right) l, & d < d_0, \\ \left(E_{\mathrm{elec}} + \varepsilon_{\mathrm{amp}} * d^4\right) l, & d \geq d_0, \end{cases} \tag{1}$$

$$\begin{aligned} E_{\mathrm{Rx}} &= E_{\mathrm{elec}} l, \\ E_{\mathrm{Dx}} &= E_{\mathrm{DA}} l, \end{aligned} \tag{2}$$

The amplifier energy $\varepsilon_{\mathrm{fs}}$ and $\varepsilon_{\mathrm{amp}}$ are the energy required for power amplification in the two models, respectively.

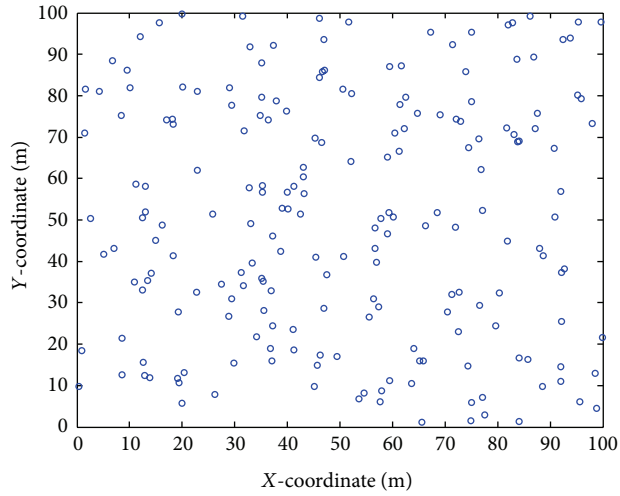


Figure 2: Data collection sensor network model.

The electronics energy E_{elec} depends on the factors such as the digital coding and modulation. E_{DA} is the energy used for aggregating unit sensor data. The typical values for the above parameters in current sensor technologies are as follows: $E_{\text{elec}} = 50\,\text{nJ/bit}$, $\varepsilon_{\text{fs}} = 10\,\text{pJ/bit/m}^2$, $\varepsilon_{\text{amp}} = 0.0013\,\text{pJ/bit/m}^4$, and $E_{\text{DA}} = 5\,\text{nJ/bit/signal}$.

3.3. Contention-Based Cluster Head Election Model. We assume that N nodes are divided into κ clusters. Then each cluster includes $n = N/\kappa$ nodes. We can guarantee the connectivity of clusters by adjusting the communication range and the number κ of cluster heads. One continuous cluster set-up phase and steady-state data transfer phase are regarded as a cluster round. Within a round, only one node can be elected as the cluster head in a cluster. For the sake of simplicity, we assume that the frame length of command frame and data frame is l bits. All sensor nodes initially consider themselves as potential candidates CH. We donot require that all sensor nodes are homogeneous in energy. The probability of being a cluster head depends on nodes' residual energy. Meanwhile, we assume that all nodes have data to transmit during each round. To ensure that all nodes die at approximately the same time, the nodes with more energy should be cluster heads rather than those with less energy. It means that a sensor node with more residual energy has more opportunity to be the cluster head than those with less residual energy in a neighboring circle within radius R. This can be achieved by setting the probability of becoming a cluster head as a function of a node's energy and the total energy of the network; thus

$$P_{ri} = \frac{E_{ri}}{E_{r\text{total}}}, \tag{3}$$

where E_{ri} is the current energy of node i and

$$E_{r\text{total}} = \sum_{i=1}^{n} E_{ri}. \tag{4}$$

When the cluster set-up phase is triggered, each cluster head will send the residual energy information of all cluster members to the sink, and the sink node will calculate and broadcast the total energy of the network $E_{r\text{total}}$ to all nodes. We assume that the CH advertisement phase is limited to a time interval of T time units and the sensor node i announces its candidacy within a radius of R at a time instance T_{ri}:

$$T_{ri} = T\left(1 - P_{ri}\right) + k_i. \tag{5}$$

Here, k_i is a random time unit that is introduced to reduce the collision possibility of among sensor node advertisements, and $P_{ri} \in [0, 1]$ represents the probability of being electing the CH for node i among all nodes in its neighboring circle in terms of its residual energy. Each sensor elects itself to be a cluster head at the beginning of the round r (which starts at time t) with a different probability. Hence, from (5), the node with the highest residual energy will have the biggest probability of being elected as a CH in a given neighborhood. Those sensor nodes that receive an advertisement message from any other sensor node will abandon their quest to become a CH and join the corresponding cluster. Obviously, only one node can be chosen as the CH in one cluster for this round. Thus, if there are n nodes in the cluster, we have

$$E\,(\text{CH}) = \sum_{i=1}^{n} P_{ri} = 1. \tag{6}$$

We will not elaborate on the cluster head election. For example, in HEED, the cluster heads are elected based on the residual energy of nodes and node connection degree. The major difference of these cluster head selection protocols lies in the information exchange between the cluster head candidate and cluster members. Once the cluster protocol is determined, the energy consumption for cluster head election and data transmission will be fixed regardless whether the energy-driven or time-driven CH rotation strategy is adopted, which only influences the time when we change the CH rotation strategy. Therefore, our proposed hybrid cluster head rotation strategy can be applied to various contention-based clusters protocols so long as the head rotation strategy is adopted.

4. Energy Consumption Analysis for One Cluster Rotation

In this section, we will analyze the energy consumption of in contention-based cluster algorithm. By analyzing the energy consumption ratio between data gathering and cluster head selection in one CH round for both time-driven and energy-driven cluster head rotation strategies, we will study more efficient cluster head rotation methods.

4.1. Energy Consumption for Cluster Set-Up Phase. Once the nodes have elected themselves to be CHs using the probabilities in (3), the CH nodes must let all the other nodes within their communication range know that they have chosen this role for the current round. To achieve this purpose, each CH node broadcasts an advertisement message (ADV) using a nonpersistent carrier-sense multiple access (CSMA) MAC protocol [24]. This message contains the necessary clustering information such as the node's ID, a frame header that distinguishes this message as an announcement message and

so forth. Noncluster head nodes determine their cluster heads in this round in accordance with the priority and strength of the received advertisement signal from cluster heads.

After every node has joined its own cluster, the CH calculates its TDMA schedule and broadcasts it to its cluster members. Then, the cluster set-up phase is finished and the entire network is divided into many clusters. Then the steady-state phase begins, in which the nodes collect data periodically. Noncluster head nodes send their data to their CH in the allotted time slot according to the TDMA schedule. The CH uses a data fusion algorithm to merge the received data and then send it to the sink. Next, we will analyze the energy consumption used for cluster head election in cluster set-up phase and data transmission in steady-state phase.

According to (3), in round r of cluster head set-up phase, the probability of one node being successfully selected as a cluster head within a time slot can be calculated as follows:

$$P_{r,s} = \sum_{i=1}^{n} P_{ri} \prod_{j=1, j\neq i}^{n} \left(1 - P_{rj}\right). \tag{7}$$

Here, $P_{r,f}$ denotes the probability that no one node has been selected as cluster head within a time slot in round r of the cluster head set-up phase. There is a certain probability that no nodes broadcast any cluster head advertisement frames within a slot, and we call this probability as idle rate of time slot denoted by $P_{r,i}$:

$$\begin{aligned} P_{r,f} &= 1 - P_{r,s}, \\ P_{r,i} &= \prod_{i=1}^{n} \left(1 - P_{ri}\right). \end{aligned} \tag{8}$$

Within one cluster, the cluster head election process will terminate when a node wins the election and becomes the cluster head. The other cluster members will apply to join it. We assume that $P_r(\tau)$ is the probability that the cluster head is elected successfully in τ time slot of round r. Consider

$$P_r(\tau) = P_{r,s} {P_{r,f}}^{\tau-1}, \quad \tau = 1, 2, 3, \ldots. \tag{9}$$

From (9), we can calculate the average mathematical expectation for the number of time slots τ when the cluster head is elected successfully. Consider

$$E(\overline{\tau}) = \sum_{\tau=1}^{\infty} \tau P_r(\tau) = \sum_{\tau=1}^{\infty} \tau P_{r,s} {P_{r,f}}^{\tau-1} = \frac{1}{P_{r,s}}. \tag{10}$$

Because each candidate node in the cluster set-up phase sends a cluster head advertisement frame with probability P_{ri}, if we use F_r to indicate the average expected number of advertisement frames in each time slot of round r, it can be given by

$$E\left(\overline{F}_r\right) = \sum_{i=1}^{n} P_{ri} * 1 = 1. \tag{11}$$

According to the formula (10), (11), we can calculate the expected mathematical number of advertisement frames in the whole cluster head election process of round r as follows:

$$C(r) = \frac{1}{\sum_{i=1}^{n} \mathrm{P}_{ri} \prod_{j=1, j\neq i}^{n} \left(1 - P_{rj}\right)}. \tag{12}$$

In fact, if $\overline{N}_{r,a}$ sensor nodes send their advertisement frames, we have

$$E\left(\overline{F}_r\right) = \overline{N}_{r,a}\left(1 - P_{r,i}\right). \tag{13}$$

In the cluster with n nodes, the remaining $n - \overline{N}_{r,a}$ nodes are responsible for receiving advertisement frames. The probability of a time slot in nonidle rate is $1 - P_{r,i}$. Therefore, we can calculate the average number of receiving cluster head announcement frame in the cluster set-up phase of round r as follows:

$$\begin{aligned} H(r) &= \left(n - \overline{N}_{r,a}\right) \overline{F}_r \overline{T}_r \left(1 - P_{r,i}\right) \\ &= \frac{n - n\prod_{i=1}^{n}\left(1 - P_{ri}\right) - 1}{\sum_{i=1}^{n} \mathrm{P}_{ri} \prod_{j=1, j\neq i}^{n} \left(1 - P_{rj}\right)}. \end{aligned} \tag{14}$$

According to (12) and (14), the energy consumption of cluster head election in one round can be given by

$$E_{\text{ch-select}} = C(r)\left(E_{\text{elec}} l + \varepsilon_{\text{amp}} {d_{\text{CH}}}^2 l\right) + H(r) E_{\text{elec}} l. \tag{15}$$

Here, d_{CH} denotes the average distance between the cluster head and other members. The left part of expression (15) represents the energy consumption of sending cluster head frame and the right part is the energy consumption of receiving cluster head circular frame. When cluster head has been elected successfully, all cluster members will send request frames to join it. Then, the CH broadcasts the TDMA schedule among its members. It is obvious that the clusters will consume the least energy if no conflict occurs. Therefore, the lower bound energy consumption of a round in the cluster set-up phase is

$$\begin{aligned} E_{\text{ch-setup}} = {} & E_{\text{ch-select}} + 2(n-1)\left(E_{\text{elec}} l + \varepsilon_{\text{amp}} {d_{\text{CH}}}^2 l\right) \\ & + 2(n-1) E_{\text{elec}} l. \end{aligned} \tag{16}$$

For the sake of simplicity, we omit the energy consumption for acknowledgement frames in data or commands transmission. This kind of packets is much smaller than the data and commands packets.

4.2. Energy Consumption for Data Transmission Phase. After the completion of set-up phase, the steady-state phase for information collection and transmission begins. For one round data collection, let d_{BS} denote the distance from CH to sink that can be estimated by CH according to signal strength received from sink, the energy consumption for CH is given by

$$E_{\text{ch}} = \left(E_{\text{elec}} + \varepsilon_{\text{amp}} d_{\text{BS}}^{\alpha}\right) l + n E_{\text{DA}} l + (n-1) E_{\text{elec}} l. \tag{17}$$

Here, the value of radio dissipation coefficient $\alpha = 2$ or 4 is determined by the distance between the transmitter and receiver according to (1). Let d_{CH} denote the average distance from cluster members to CH which can be given by $M^2/2\kappa\pi$ [5], the energy consumption for all member nodes in cluster for one round data collection is calculated as follows:

$$E_{\text{non-ch}} = (n-1)\left(E_{\text{elec}} + \varepsilon_{\text{amp}} d_{\text{CH}}^{\alpha}\right) l. \tag{18}$$

The number of data collection rounds depends on the CHs rotation method. In time-driven CH rotation, the number of data transmission rounds is predetermined at the beginning of network operation. It is related to the properties of the network such as nodes density and energy status. We will provide the method to estimate it in the following section. We assume that there are k rounds of data collection and transmission, and then the energy consumption in one steady-state phase based on time-driven CH rotation is

$$E_{\text{Time-}D_r} = k\left(E_{\text{ch}} + E_{\text{non-ch}}\right). \tag{19}$$

However, in energy-driven CH rotation strategy, the number of data transmission rounds is a function of the remaining energy of CH node. The CH node i calculates a dynamic energy threshold λ_i based on its residual before it broadcasts its CH Candidacy. The threshold is defined by $\lambda_i = cE_{\text{res}_i}$, where $c \in [0, 1]$ is a predetermined constant and E_{res_i} is the residual energy of cluster head. When the residual energy of cluster head has dropped below this threshold, the steady-state phase will be over and new CHs rotation process is triggered. If we let $\varphi_{r,i}$ denote the number of data transmission rounds that node i severs as a CH, we can calculate the energy consumption in one steady-state phase for energy-driven cluster head rotation method as follows:

$$\begin{aligned} E_{\text{Energy-}D_r} &= \varphi_{r,i}\left(E_{\text{ch}} + E_{\text{non-ch}}\right) \\ &= \frac{(1-c)\,E_{ri} - E_{\text{choh}}}{E_{\text{ch}}} \\ &\quad \times \left(E_{\text{ch}} + E_{\text{non-ch}}\right). \end{aligned} \tag{20}$$

Here, E_{choh} denotes the energy consumption of CH node i in cluster set-up phase. When no conflict occurs, E_{choh} includes the energy consumption for broadcasting CH candidacy, receiving requests for joining from all of its members, and broadcasting TDMA schedule among its members. Therefore, its lower bound can be given by

$$E_{\text{choh}} = 2(n-1)\left(E_{\text{elec}}l + \varepsilon_{\text{amp}}d_{\text{CH}}^2 l\right) + (n-1)\,E_{\text{elec}}l. \tag{21}$$

5. Hybrid Energy- and Time-Driven Cluster Head Rotation Strategy

After analyzing the energy consumption in one CH round for both time-driven and energy-driven cluster head rotation strategies, we will deduce the hybrid cluster head rotation strategy which combines the advantages of both energy-driven and time-driven methods in this section. In the cluster-based WSNs, energy consumption can be divided into two categories. The first category is the energy consumption used for topology construction and maintenance. We call it additional energy cost E_a which is mainly consumed in cluster set-up phase. The second category is the energy consumption used for data transmission and reception. We call it efficiency energy cost E_e which is mainly consumed in steady-state phase. The main purpose of the WSNs is to collect as much useful information as possible in the monitoring area. So, the more energy is used for information collection, the better energy efficiency of cluster hierarchy protocol will be achieved. We define the energy efficiency η as the proportion of the effective energy cost in the total energy cost. Consider

$$\eta = \frac{E_e}{E_e + E_a}. \tag{22}$$

In time-driven CH rotation strategy, we get the energy efficiency according to (15), (16), and (19):

$$\begin{aligned} \eta_{\text{Time-}D_r} &= \frac{E_{\text{Time-}D_r}}{E_{\text{Time-}D_r} + E_{\text{ch-setup}}} \\ &= \frac{k\left(E_{\text{ch}} + E_{\text{non-ch}}\right)}{k\left(E_{\text{ch}} + E_{\text{non-ch}}\right) + E_{\text{ch-setup}}}. \end{aligned} \tag{23}$$

In energy-driven CH rotation strategy, according to (15), (16) and (20), we have the energy efficiency:

$$\begin{aligned} \eta_{\text{Energy-}D_r} &= \frac{E_{\text{Energy-}D_r}}{E_{\text{Energy-}D_r} + E_{\text{ch-setup}}} \\ &= \frac{\left[(1-c)\,E_{ri} - E_{\text{choh}}\right]\left(E_{\text{ch}} + E_{\text{non-ch}}\right)}{\left[(1-c)\,E_{ri} - E_{\text{choh}}\right]\left(E_{\text{ch}} + E_{\text{non-ch}}\right) + E_{\text{ch}}E_{\text{ch-setup}}}. \end{aligned} \tag{24}$$

In (23), E_{ch}, $E_{\text{non-ch}}$, and E_{choh} are only related to the cluster size and node density. $E_{\text{ch-setup}}$ will change little if cluster scale is under good control. That is to say, in a cluster hierarchy WSNs with nodes distribution, the parameters above can be nearly seen as almost constant. Normalizing $\eta_{\text{Time-}D_r}$, we have

$$\eta_{\text{Time-}D_r} = \frac{1}{1 + \left(\left(E_{\text{ch-setup}}\right)/\left(E_{\text{ch}} + E_{\text{non-ch}}\right)\right) * (1/k)}. \tag{25}$$

From (25), we can see that the energy efficiency for time-driven CH rotation is only related to the steady-state phase. Once the data transmission rounds k is predetermined, $\eta_{\text{Time-}D_r}$ will be fixed. This is a significant feature of the time-driven CH rotation method, which means that constant energy efficiency can be maintained regardless of how long the network runs. Furthermore, normalizing $\eta_{\text{Energy-}D_r}$, we have

$$\eta_{\text{Energy-}D_r} = 1 \times \left(1 + \frac{E_{\text{ch}}E_{\text{ch-setup}}}{E_{\text{ch}} + E_{\text{non-ch}}} * \frac{1}{(1-c)\,E_{ri} - E_{\text{choh}}}\right)^{-1}. \tag{26}$$

In (26), we have that the restriction of $(1-c)E_{ri} > E_{\text{choh}}$. c is a predetermined constant which depends on the topology of WSNs. The energy efficiency $\eta_{\text{Energy-}D_r}$ for energy-driven CH rotation is only related to the residual energy of cluster head node E_{ri}. The more residual energy of the cluster head has, the higher energy efficiency we can obtain. The residual energy E_{ri} will decrease gradually along with the operation of WSNs. It means that more energy will be used for frequent

cluster head election and cluster topology construction when $\eta_{\text{Energy-}D_r}$ becomes lower. Network can achieve good energy efficiency if the remaining energy of CHs is abundant and will get low energy efficiency if the residual energy of CHs is inadequate. This is the major drawback of energy-driven CH rotation strategy.

We set d_{CH} and d_{BS} as 30 meters and 87 meters, respectively. The distribution density of sensor node is 0.02 and each node is assigned with 0.5 J energy. For the threshold parameter c, without the loss of generality, we adopt the value $c = 0.644$ according to the research results in [22] and let $k = 5$. Advertisement set-up packets and data packets are arranged with 256 bits in length. Then, we can get the change in the energy efficiency with the network operation rounds for energy-driven CH rotation and time-driven CH rotation in clustering WSN as showed in Figure 3. We can see that high energy efficiency can be achieved at the beginning of WSNs if energy-driven CH rotation strategy is adopted. But as the operation time of WSNs increases, the residual energy of the selected CHs will decrease accordingly, which will result in decrease in energy efficiency. When the residual energy of the cluster head declines to a certain level, the energy efficiency of the energy-driven CH rotation strategy will be lower than that of the time-driven CH rotation strategy.

In order to collect as much useful information as possible in the monitoring area and keep as higher energy efficiency as possible, we can make use of the advantages of both the energy-driven strategy and time-driven CH rotation strategy in cluster hierarchy protocol. When the nodes have abundant energy, we can adopt energy-driven CH rotation strategy to achieve good energy efficiency and avoid excessive overhead frames in time-driven CH rotation policy. When WSNs' energy declines to the extent that the energy efficiency of energy-driven CH rotation strategy is lower than that of the time-driven CH rotation strategy, the CH rotation strategy will switch to time-driven method to maintain energy efficiency and data transmission efficiency. This adaptive switch can optimize the energy efficiency of the network. The analysis results above constitute the theoretical foundation of our hybrid energy- and time-driven cluster head rotation strategy in this paper.

According to the results above, when $\eta_{\text{Energy-}D_r} \geq \eta_{\text{Time-}D_r}$, we will adopt the energy-driven CH rotation strategy to obtain better energy efficiency. The time-driven CH rotation strategy is adopted when $\eta_{\text{Energy-}D_r} < \eta_{\text{Time-}D_r}$. Furthermore, we can simplify this criterion. As indicated in Figure 3, when the energy efficiency in energy-driven and time-driven CH rotation is equal to $\eta_{\text{Energy-}D_r} = \eta_{\text{Time-}D_r}$, we have

$$\frac{1}{1 + \left(\mathrm{E}_{\text{ch-setup}}/E_{\text{ch}} + E_{\text{non-ch}}\right) * (1/k)} = 1 \times \left(1 + \frac{E_{\text{ch}} E_{\text{ch-setup}}}{E_{\text{ch}} + E_{\text{non-ch}}} * \frac{1}{(1-c)\, E_{ri} - E_{\text{choh}}}\right)^{-1}. \tag{27}$$

From (27), we can calculate the energy of the cluster head node when CH rotation strategy needs to switch from energy-driven strategy to time-driven strategy. We define this energy of cluster head as critical energy value and it can be given by

$$E_{ri}^{*} = \frac{kE_{\text{ch}} + E_{\text{choh}}}{1 - c}. \tag{28}$$

From (28), we can see that the critical energy value of CH is only related to the intrinsic parameters of energy-driven and time-driven CH rotation. Moreover, this result proves that the hybrid CH rotation strategy does not depend on homogeneous or heterogeneous characteristics of the network and the exact cluster protocol. What is more is another important issue is to estimate the optimal data transmission rounds k for time-driven CH rotation method in (28). It is an important parameter that determines the timing for switching to the CH rotation strategy. For heterogeneous WSNs, we notice that the energy balance between sensor nodes has been greatly improved after the long time operation of the network using the energy-driven cluster head rotation strategy. It meets the network conditions for time-driven protocol to maintain high energy efficiency, and therefore, it is a good decision to switch to the time-driven cluster head rotation strategy.

In time-driven CH rotation cluster protocol, the steady-state phase should be as long as possible in order to increase energy efficiency. On the other hand, since each node's energy is limited, it will drain the energy of the CH node if the steady-state phase is too long. In [5, 18], authors have proved that if each node can be ensured to act as CH once and noncluster head node in the other rounds during network lifetime, a suboptimal network lifetime for the time-driven CH rotation cluster protocol can be achieved. Generally speaking, the sensor node with the least residual energy will die first. Let E_{min} denote the residual energy of the node with the minimum energy in WSNs when the time-driven rotation strategy is selected. To ensure that each node has enough

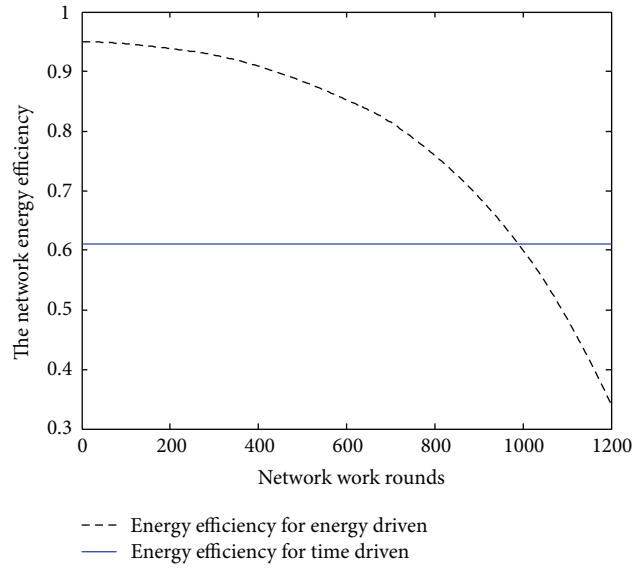


Figure 3: Energy efficiency in energy-driven and time-driven strategies.

energy to be a cluster head for one time and noncluster head sensor node for $(n-1)$ times, according to (17) and (18), we can get the following equation:

$$kE_{\text{ch}} + (n-1)\, k\frac{E_{\text{non-ch}}}{n-1} = E_{\min}. \tag{29}$$

Therefore, we can obtain the suboptimal number of data transmission rounds k when energy-driven CH rotation method is switched to time-driven method:

$$k = E_{\min} \times \left(\left[\left(E_{\text{elec}} + \varepsilon_{\text{amp}} d_{\text{BS}}^{\alpha}\right) l + nE_{\text{DA}} l + (n-1)\, E_{\text{elec}} l\right] + (n-1)\left(E_{\text{elec}} + \varepsilon_{\text{amp}} d_{\text{CH}}^{\alpha}\right) l\right)^{-1}. \tag{30}$$

We can see that the parameter k will change along with the gradual decrease of $E_{\min}$. From (28) and (30), we can calculate the corresponding k for each round of CH rotation and then get the critical energy value E_{ri}^{*}. When the residual energy of each CH $E_{ri} \geq E_{ri}^{*}$, the energy-driven CH rotation strategy will be adopted; otherwise, the time-driven CH rotation strategy will be selected.

The pseudo code of hybrid energy- and time-driven cluster head rotation strategy is presented in Algorithm 1. First, in cluster set-up phase, each node computes its P_{ri} using immediate neighbor information to run for cluster head. At the same time, the sink node computes and broadcasts parameter k to the whole network according to the *Res_Energy_List that* is collected and sent back by all CHs. After successful election, each CH calculates E_{ri}^{*} and compares it with its own residual energy. If CHs' residual energy is larger than E_{ri}^{*} or even equal, energy-driven CH rotation strategy is adopted; otherwise, time-driven CH rotation strategy will be used. When the network is worked based energy-driven CH rotation strategy, if the cluster head rotation is triggered, CHs will send the residual energy list *Res_Energy_List* of all cluster members back to the sink node. Once WSNs switch to time-driven CH rotation strategy, there is no longer any need to calculate the parameter k, and the network will operate based on time-driven CH rotation strategy until the network dies.

From Algorithm 1, we can see that the hybrid CH rotation strategy increases the message exchange of cluster protocol in cluster set-up phase. Since the cluster head selection process is message driven, we will discuss the message complexity of algorithm. We will validate that our hybrid rotation strategy does not add the complexity for all clustering protocol through the complexity analysis. When energy-driven rotation strategy is selected, at the beginning of the cluster head election phase, the sink node will broadcast system frame with parameter k included. Then κ tentative cluster heads are elected and each of them broadcasts election header frame *COMPETE_HEAD_MSG*. Later on, they make a decision to act as a final cluster head by broadcasting election frame *FINAL_HEAD_MSG*, or an ordinary node by broadcasting a quit election frame *QUIT_ELECTION_MSG*. They send out κ cluster heads to confirm acknowledgement *CH_ADV_MSGs*, and then $(N-\kappa)$ ordinary nodes transmit $(N-\kappa)$ join cluster frames *JOIN_CLUSTER_MSGs*. Thus, the messages add up to $N + \kappa + \kappa + \kappa + N - \kappa = 2(N + \kappa)$ in the cluster formation stage per round, that is, $O(N)$. When time-driven rotation strategy is selected, the network will not select cluster head rotation strategy any longer and the message complexity is the same as that of time-driven CH rotation strategy cluster protocol. Analysis above shows that the hybrid CH rotation strategy will not increase the complexity of cluster protocol compared with single energy-driven or time-driven CH rotation strategy.

6. Simulation Results

In this section, both the numerical results and the simulation results will be presented to evaluate the performance of the hybrid energy- and time-driven cluster head rotation strategy. The simulations are performed in Matlab 7.0, and two scenarios are chosen. We will firstly discuss the influence of the threshold parameter c on network and find a suitable value for our sensor networks model. Then, we will apply our hybrid CH rotation strategy, time-driven and energy-driven CH rotation methods, respectively, to reform LEACH, which is a famous and successful cluster hierarchy protocol recently. By comparing the lifetime and valid data transmission efficiency for the three CH rotation strategies, we will prove that our hybrid CH rotation strategy truly improves the energy efficiency as well as prolongs the network lifetime in both homogeneous and heterogeneous networks. In order to conduct the experiments, proper parameters for both the sensor nodes and the network should be defined. The simulation parameters for our proposed mechanism are given in Table 1.

In the following simulation, we evaluate the three CH rotation strategies in both homogeneous and heterogeneous network. In homogeneous WSNs, 200 nodes each with 0.5 J energy are randomly dispersed in a 100×100 region with BS located at (50, 50). In heterogeneous WSNs, 200 nodes each with 0.3 J to 0.8 J energy (randomly assigned) are randomly dispersed in a 100×100 region with BS located at (50, 50).

6.1. The Analysis of Energy Threshold. As we have explained in the previous section, the energy efficiency is the standard to judge whether energy-driven and time-driven CH rotation mechanism should be selected in our hybrid CH rotation strategy. When energy-driven CH rotation is selected, the residual energy of each existing CH will determine when a network calls for a new CH selection phase. The selection phase will be triggered once the residual energy of any of CHs falls below a predetermined threshold value. This energy threshold influences the energy efficiency of energy-driven rotation method and decides when the cluster head selection policy should turn from energy-driven to time-driven mechanism. Therefore, it is vital to find out the suitable energy threshold parameter c for energy-driven CH rotation strategy based on the above network model and parameters before performance evaluation.

Figure 4 illustrates the network lifetime of LEACH based on our hybrid energy- and time-driven CH rotation strategy in homogeneous WSNs with the energy threshold parameter c increasing from 0.4 to 0.8. Figure 5 shows these results in heterogeneous WSNs. From these two figures, we can see that network lifetime will change with different threshold

Algorithm: Hybrid cluster head rotation mechanism

1: *Setup* ()
2: **if** Initial round then
3: Compute P_{ri} using immediate neighbor information
4: **end if**
5: Sink computer k and broadcast CH Selection Command
6: CHs *Selection (COMPETE_HEAD_MSG, FINAL_HEAD_MSG, QUIT_ELECTION_MSG)*
7: CHs *Construction (CH_ADV_MSGs, JOIN_CLUSTER_MSGs)*
8: CHs compute E^*_{ri}
9: **if** $E_{ri} \geq E^*_{ri}$
10: {**while** $(E_{\text{res}_i} \geq cE_{ri})$
11: {*data_collection* ()
12: *data_transmission* ()}
13: CH Request BS for a CH Change
14: CHs Sent *Res_Energy_List* to Sink
15: **goto** 2
16: }
17: **else goto 18**
18: Sink broadcast Switch to Time-driven CH Rotation
19: **if** Initial round then
20: Compute P_{ri} using immediate neighbor information
21: **end if**
22: CHs *Selection* () and *Construction* ()
23 *Data_Round* = k
24: **while** (*Data_Round* > 0)
25: {*data_collection* () and *data_transmission* ()
26: *Data_Round* = *Data_Round*-1}
27: **else goto 19**

Algorithm 1: Hybrid energy- and time-driven cluster head rotation strategy.

Table 1: Parameters and characteristics of the network.

Parameter	Value
Network size (square)	100 × 100 meter
Sink location	(50, 50)
Command frame and data packet size	256 bits
Number of nodes	200
Cluster head probability	0.05
Aggregation ratio	0.1

parameters. When energy threshold is small, nodes die evenly as time passes by. When energy threshold is larger, we can see that the lifetime curve trends to right angle. Most of nodes will have a longer life and die almost simultaneously within a short time interval. When c is small, the elected cluster head node will consume most of its energy on cluster head position. After rotation, it has little residual energy and will die quickly. When c is larger, the cluster heads can still reserve some energy to act as regular nodes after fulfilling their role as cluster heads. Since noncluster head nodes consume less energy than cluster head nodes, the reserved energy can help it to last for a longer time. Nodes die almost simultaneously because all the nodes have little energy left after rotating for several times. Meanwhile, if c is larger, the CH will rotate frequently and waste lots of energy compared with sleep-listen model in steady-state phase. This result can be seen from Figures 4 and 5. It shows that network lifetime is longer when $c = 0.6$ rather than $c = 0.7$.

In [7], the authors have proved that the optimal value c mainly varies with the distance from nodes to their CHs and with the distance from CH to sink node in a single energy-driven CH rotation. In other words, it is mainly determined by network topology structure. The network will switch to time-driven CH rotation model in the late stage of network in accordance with our hybrid rotation method. It can be seen in Figures 4 and 5 that the energy threshold parameter acquires almost the same value when the network gains the optimal lifetime no matter whether in homogeneous or heterogeneous WSNs, so long as they have the same network topology. From the simulation and analysis results above, we find that the network will achieve better lifetime when $c = 0.6$. We will adopt this threshold value in the following part. For single energy-driven CH rotation, we use the optimal value $c = 0.644$ according to the research results in [22].

6.2. Performance Comparison in Network Lifetime and Energy Efficiency. In this part, we verify the network lifetime and energy efficiency of our hybrid CH rotation strategy by LEACH cluster protocol. To estimate the lifetime of the WSNs, the metrics of First Node Dies (FNDs), Percent of the Nodes Alive (PNA), and Last Node Dies (LNDs) are always used [9, 20]. FND is useful in sparsely deployed WSNs. However, PNA is more suitable to measure the network

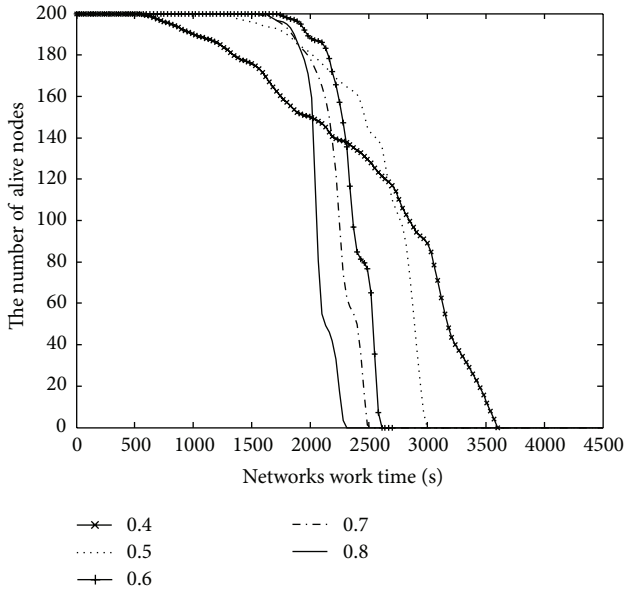


Figure 4: Comparison of lifetime for threshold parameter c in homogeneous WSNs.

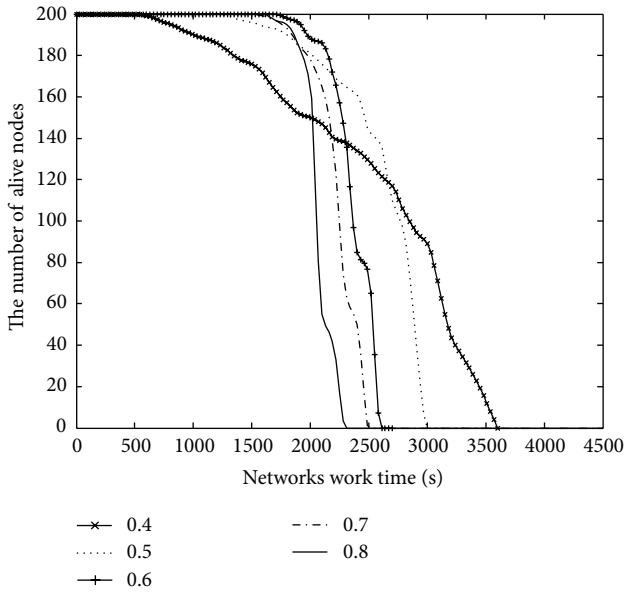


Figure 5: Comparison of lifetime for threshold parameter c in heterogeneous WSNs.

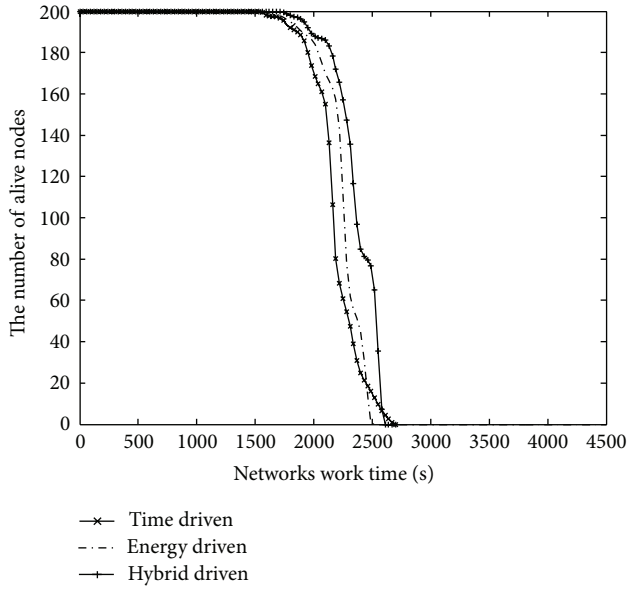


Figure 6: Lifetime in homogeneous WSNs for LEACH based on three rotation strategies.

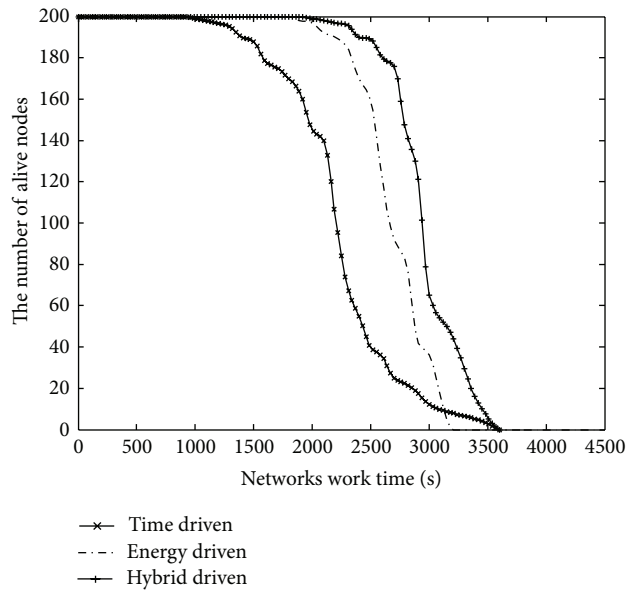


Figure 7: Lifetime in heterogeneous WSNs for LEACH based on three rotation strategies.

lifetime in densely deployed WSNs. LND is not suitable for practical application. Hence, in this paper, we use the FND and PNA metrics to measure performance of our CHs rotation strategy. In the case of PNA, we have assumed 95% of nodes alive in the network.

Figures 6 and 7 present the comparison of network lifetime for LEACH based on three CH rotation strategies both in homogeneous and heterogeneous WSNs. We can see that our hybrid energy- and time-driven CH rotation strategy is better than either the single energy-driven or time-driven CH rotation method both in FND and PNA. In homogeneous network, the initial energy of each node is equal, so the time-driven cluster head rotation strategy can also achieve a better balance on energy consumption among nodes. The advantage for hybrid CH rotation strategy in the network lifetime is not obvious in FND, but there is still a lot of improvement in PNA. However, in heterogeneous network, our hybrid CH rotation approach is remarkably better than both the time-driven CH rotation strategy and the energy-driven CH rotation strategy in terms of FND and PNA. For the time-driven CH rotation strategy, the initial energy is not equal in heterogeneous WSNs and it cannot achieve the energy consumption balance

among sensor nodes. On the other hand, for the energy-driven CH rotation strategy, the frequent rotation of cluster head may make all nodes always in active stage and lead to premature death in the final stage of WSNs, thus reducing the network lifetime.

The mechanism for prolonging the network lifetime by hybrid CH rotation strategy can be explained as follows. After clusters have been built successfully, the cluster head creates a TDMA schedule and tells each member node the exact time for transmission data. This allows sensor nodes to remain in the sleep model in other nodes' time slot in steady-state phase. If the node is in the sleep state, the radio and processor modules are turned off which can save considerable energy. But in cluster set-up phase, all nodes must keep active with all modules which will consume much energy especially for the radio communication module. If the energy-driven CH rotation strategy is adopted, when the nodes have abundant energy, the steady-state phase for data transmission is longer than that of time-driven CH rotation strategy, which means less cluster head frames and longer sleeps state. When the energy of nodes is scarce, if we continue to use energy-driven CH rotation method, frequent cluster head selection will happen, which will result in lots of energy waste. However, if we switch to the time-driven CH rotation strategy, frequent cluster head selection can be avoided and the network lifetime can be prolonged.

When the energy level of network is low and energy of nodes is balanced, time-driven CH rotation method can achieve higher energy efficiency than energy-driven CH rotation method. Table 2 shows the energy distribution between sensors when we switch from energy-driven CH rotation to time-driven CH rotation in heterogeneous network for LEACH. It shows that the energy of sensor nodes is quite evenly distributed among them. So, switching to time-driven CH rotation is a good choice when $\eta_{\text{Energy-}D_r} < \eta_{\text{Time-}D_r}$ and it also proves the rationality of the hybrid energy- and time-driven CH rotation strategy indirectly.

In addition to network lifetime, energy efficiency is another important performance indicator of WSNs. As replacing or recharging batteries is difficult or even impossible for sensor nodes in many cases, the cluster protocol needs to maintain high energy efficiency of networks in order to collect as much useful information as possible. According to the algorithmic details of our hybrid energy-time driven CH rotation, energy efficiency of the network is used as the criterion to select CH rotation strategy. In the next simulation test, we use the data transmission rounds that 95% of nodes are alive in the sensor network to measure and compare the energy efficiency.

Figures 8 and 9 illustrate the energy efficiency comparison among the three CH rotation strategies in both homogeneous and heterogeneous networks for LEACH. It can be seen that even though there is no big difference in network lifetime in homogeneous sensor network, our hybrid energy- and time-driven CH rotation strategy is about 1.3 times and 1.1 times more efficient in data transmission than time-driven CH rotation and energy-driven CH rotation, respectively. But in heterogeneous WSNs, the hybrid energy- and time-driven CH rotation strategy can achieve 2 times and 1.2 times more efficient in data transmission than time-driven CH rotation strategy- and energy-driven CH rotation strategy, respectively. Such results are mainly attributable to the fact that time-driven CH rotation strategy does not consider the energy imbalance of nodes. Under such a circumstance, cluster head selection is triggered after a constant number of data transmission rounds no matter whether the node's energy is abundant or not, thus resulting in unnecessary energy waste. At the same time, for energy-driven CH rotation strategy, the energy efficiency will decrease sharply when the energy of network reduce to some degree. The hybrid CH rotation strategy takes full account of such a situation and can achieve good energy efficiency by selecting either energy-driven or time-driven CH rotation according to the residual energy of network in real time.

Table 2: Node energy distribution between sensors in heterogeneous network.

Nodes Energy Distribution (J)	When energy-driven CH rotation begins	When time-driven CH rotation begins
0-0.1	0	8
0.1-0.2	0	147
0.2-0.3	0	41
0.3-0.4	42	4
0.4-0.5	40	0
0.5-0.6	43	0
0.6-0.7	39	0
0.7-0.8	36	0

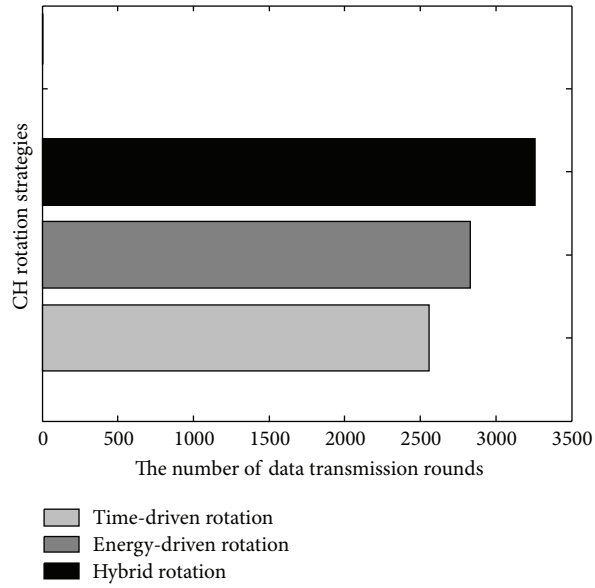


Figure 8: Data transmission rounds with PND for LEACH in homogeneous networks.

7. Conclusion

In this paper, we have proposed a hybrid energy- and time-driven CH rotation strategy for distributed wireless sensor

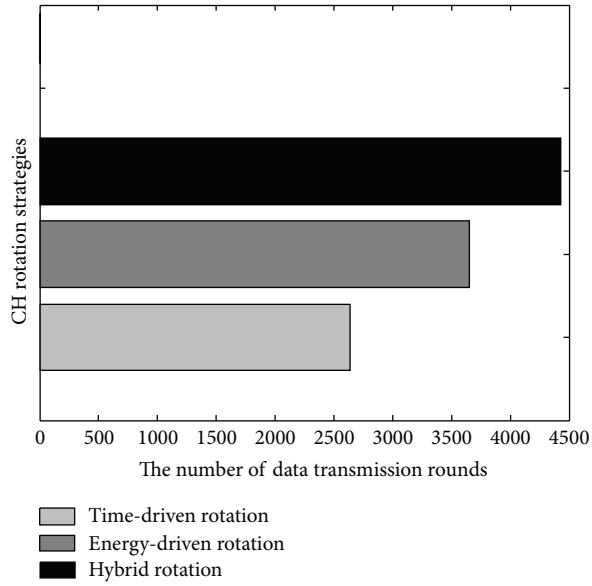


Figure 9: Data transmission rounds with PND for LEACH in heterogeneous networks.

networks. The main objective of our algorithm is to improve the energy efficiency and prolong the lifetime of WSNs by optimizing the CH rotation strategy in cluster hierarchy protocol. By analyzing energy consumption ratio between data gathering and cluster head selection for the time-driven cluster head rotation and energy-driven cluster head rotation, we propose the hybrid cluster head rotation strategy for WSNs, in which the CHs rotation mechanism based on time or energy will be selected according to the change of energy efficiency.

By comparing the simulation results, we find that our rotation strategy integrates the advantages of both the energy-driven CH rotation strategy and the time-driven CH rotation strategy in cluster hierarchy protocol and can enhance the energy efficiency and prolong network lifetime in both homogeneous and heterogeneous networks. As an important technology of cluster topology maintenance, our rotation strategy can be used in any cluster protocol for the topology reconstruction of the hierarchical cluster network.

The energy threshold parameter c plays an important role in our hybrid CH rotation strategy and it is mainly decided by network topology structure. In this paper, we only obtain a suitable value by simulation analysis. How to find out its optimal value in a different network topology structure for WSNs is an issue of great importance for our hybrid CH rotation strategy. We will try to find a solution that could determine the optimal value of these parameters according to different cluster topology structures in our future work.

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] G. Chen, C. Li, M. Ye, and J. Wu, "An unequal cluster-based routing protocol in wireless sensor networks," *Wireless Networks*, vol. 15, no. 2, pp. 193–207, 2009.

[3] D. Culler and W. Hong, "Wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 30–33, 2004.

[4] U. C. Kozat, G. Kondylis, B. Ryu, and M. K. Marina, "Virtual dynamic backbone for mobile ad hoc networks," in *Proceedings of the International Conference on Communications (ICC '01)*, pp. 250–255, Helsinki, Finland, June 2001.

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[6] Y. Wang, Q. Zhao, and D. Zheng, "Energy-driven adaptive clustering data collection protocol in wireless sensor networks," in *Proceedings of the International Conference on Intelligent Mechatronics and Automation*, pp. 599–604, Chengdu, China, August 2004.

[7] S. Gamwarige and E. Kulasekere, "An algorithm for energy driven cluster head rotation in a distributed wireless sensor network," in *Proceedings of the International Conference on Information and Automation (ICIA '05)*, pp. 354–359, Colombo, Sri Lanka, December 2005.

[8] Q. Zhang, D. Shao, L. Sun, and H. Huang, "A cluster head rotation based on energy consumption for wireless sensor network," *International Journal of Advancements in Computing Technology*, vol. 4, no. 11, pp. 239–247, 2012.

[9] G. Smaragdakis, I. Matta, and A. Bestavros, "SEP: a stable election protocol for clustered heterogenous wireless sensor networks," in *Proceedings of the International Workshop on SANPA*, Boston, Mass, USA, 2004.

[10] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[11] J. Kamimura, N. Wakamiya, and M. Murata, "Energy-efficient clustering method for data gathering in sensor networks," in *Proceedings of the 1st Workshop on Broadband Advanced Sensor Networks*, October 2004.

[12] H. Chan and A. Perrig, "ACE: an emergent algorithm for highly uniform cluster formation," in *Proceedings of the 1st European Workshop on Sensor Networks (EWSN '04)*, pp. 154–171, January 2004.

[13] C. F. Chiasserini, I. Chlamtac, P. Monti, and A. Nucci, "An energy-efficient method for nodes assignment in cluster-based ad hoc networks," *Wireless Networks*, vol. 10, no. 3, pp. 223–231, 2004.

[14] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the 22nd IEEE Societies Annual Joint Conference of the IEEE Computer and Communications (INFOCOM '03)*, vol. 3, pp. 1713–1723, 2003.

[15] S. Bandyopadhyay and E. J. Coyle, "Minimizing communication costs in hierarchically-clustered networks of wireless sensors," *Computer Networks*, vol. 44, no. 1, pp. 1–16, 2004.

[16] T. Moscibroda and R. Wattenhofer, "Maximizing the lifetime of dominating sets," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, April 2005.

[17] X. Fan and Y. Song, "Improvement on LEACH protocol of wireless sensor network," in *Proceedings of the International Conference on Sensor Technologies and Applications (SENSORCOMM '07)*, pp. 260–264, Valencia, España, October 2007.

[18] R. Madan and S. Lall, "Distributed algorithms for maximum lifetime routing in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 8, pp. 2185–2193, 2006.

[19] F. M. Hu, Y. H. Kim, K. T. Kim, H. Y. Youn, and C. W. Park, "Energy-based selective cluster-head rotation in wireless sensor networks," in *Proceedings of the International Conference on Advanced Infocomm Technology (ICAIT '08)*, July 2008.

[20] S. Soro and W. B. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 955–972, 2009.

[21] Y. Wu, Z. Chen, Q. Jing, and Y. C. Wang, "LENO: LEast rotation near-optimal cluster head rotation strategy in wireless sensor networks," in *Proceedings of the 21st International Conference on Advanced Information Networking and Applications (AINA '07)*, pp. 195–201, Niagara Falls, Canada, May 2007.

[22] S. Gamwarige and C. Kulasekere, "Performance analysis of the EDCR algorithm in a distributed wireless sensor network," in *Proceedings of the IFIP International Conference on Wireless and Optical Communications Networks*, Bangalore, India, April 2006.

[23] T. Rappaport, *Wireless Communications: Principles & Practice*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.

[24] K. Pahlavan and A. Levesque, *Wireless Information Networks*, Wiley, New York, NY, USA, 1995.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 718297, 12 pages
http://dx.doi.org/10.1155/2013/718297

Research Article

An Efficient Data Evacuation Strategy for Sensor Networks in Postdisaster Applications

Ming Liu,[1] Bang Liu,[1] and Yonggang Wen[2]

[1] *School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*
[2] *School of Computer Engineering, Nanyang Technological University, Singapore 639798*

Correspondence should be addressed to Ming Liu; csmliu@uestc.edu.cn

Received 10 October 2012; Accepted 26 November 2012

Academic Editor: Nianbo Liu

Disasters often result in a tremendous cost to our society. Previously, wireless sensor networks have been proposed to provide information for decision making in postdisaster relief operations. The existing WSN solutions for postdisaster operations normally assume that the deployed sensor network can tolerate the damage caused by disasters and maintain its connectivity and coverage, even though a significant portion of nodes have been physically destroyed. Inspired by the "blackbox" technique, we propose that preserving "the last snapshot" of the whole network and transferring those data to a safe zone would be the most logical approach to provide necessary information for rescuing lives and control damages. In this paper, we introduce data evacuation (DE), an original idea that takes advantage of the survival time of the WSN, that is, the gap from the time when the disaster hits and the time when the WSN is paralyzed, to transmit critical data to sensor nodes in the safe zone. Numerical investigations reveal the effectiveness of our proposed DE algorithm.

1. Introduction

While disasters could result in a tremendous cost to our society, access to environment information in the affected area, such as, damage level and life signals, has been proven crucial for relief operations. Hundreds of disasters in various scales, including earthquakes, flooding, tornadoes, oil spilling, and mining accidents, happen around the world each year. Not only do they bring in huge economic lost by destroying assets, but also they can take lives in large quantities. According to a disaster statistic report [1], the average number of people affected by disasters is more than two hundred million per year from 1991 to 2005, and thousands of them lost their lives. When disasters hit, relief operations often focus on saving lives and reducing property damages. Given the chaos in the affected areas, effective relief operations highly depend on timely access to environment information. For example, the life vitals of survivors would be extremely helpful for rescue workers to determine where to dig a tunnel to the spot. Previously, wireless sensor networks [2–4] have been proposed to gather useful information in disasters such as earthquake, volcano eruption, and mining accidents.

However, even with sensor networks, gathering crucial information in postdisaster relief operations turns out unpredictably challenging. When a disaster strikes, the communication facilities, power units, and roads will usually be destroyed, which, along with some concomitant accidents, for example, building collapse, fires, and gas explosions, and so forth, may disrupt the normal functionalities of sensor networks. For example, sensor nodes could be damaged in the event of a fire and communication channels are thus disconnected. Previous researches [5–10] tend to overlook at this possibility and thus result in relief solutions that are inherently impractical. As a result, the decision-making process could be paralyzed with incomplete information.

In this paper, inspired by the "blackbox" solution in flight industry, we propose data evacuation (DE): an original idea which utilizes the surviving time interval of sensor nodes, namely, the duration in which WSNs still function after the disaster, to transmit vital data to the sensor nodes in the safe

zone. Our idea relies on the following observation. It is quite possible that the buildings or local resources do not get damaged or destroyed at the beginning of most disasters. As a result, the deployed sensor network can keep working for a while before it becomes completely paralyzed. This grace period can be used to transit vital data gathered by the WSN.

Our proposed data evacuation works under extreme situations, thus requiring different design metrics from normal wireless sensor networks. From an engineering perspective, one would like to gather as much information as possible, preferably within short period as much as possible. The former metric corresponds to the evacuation ratio, defined as the amount of successfully rescued information in respect to the whole amount of information gathered by the network; the latter corresponds to the evacuation time, defined as the amount of time spent in rescuing all the information. However, as with any engineering design problem, these two metrics are competing with each other. Since maximizing evacuation ratio and minimizing evacuation time cannot be achieved simultaneously in any design of DE, it is our responsibility to judicially balance a tradeoff between the two metrics in a realistic solution of DE.

In this research, we first reveal a mathematical structure of our problem, and then our main focus turns to develop and evaluate scalable distributed algorithms for our proposed DE strategy. If one would trace the path of each bit of data transits in the network, this problem can be modeled as a non-linear programming problem with multiple minimums in its support. Rather than seeking the analytical solution for such a formulation, we take a pragmatic approach to design distributed protocols to route the vital data to safe zones in an affected region. We will propose two distributed data-rescuing protocols, namely, a gradient-based (GRAD-DE) one and a gravitation-based (GRAV-DE) one. The former is related to Newton's method [11] for non-linear programming and the latter is related to Newton's law of physics [12]. In addition, we will evaluate their efficacy under the aforementioned design metrics with extensive simulation. Evaluation shows the significant effectiveness of DE strategies for postdisaster applications. The major contributions of this works are as follows.

(i) To the best of our knowledge, we are the first to propose the idea of *data evacuation* for postdisaster applications. The basic operation of DE is to send sensitive data from the whole network to the nodes in the safe zone; in that case, the relief efforts of rescue group will benefit a lot from the reproduction of "the last shot" of the monitoring region based on the saved sensitive data.

(ii) Building the mathematical structure of our problem, we propose two distributed data-rescuing algorithms. Our algorithms are mathematic avatars of Newton's method on non-linear optimization and Newton's law of physics.

(iii) Extensive simulation has been conducted to verify the efficacy of GRAD-DE and GRAV-DE and illustrate the fundamental tradeoff between the two design metrics: evacuation time and evacuation ratio.

The remainder of this paper is organized as follows. Section 2 discusses the related work. Section 3 gives the definitions and assumptions about disaster scenario and network model. Section 4 presents the detailed design of GRAD-DE and GRAV-DE, followed by their evaluations in Section 5. Section 6 concludes the paper.

2. Related Work

One of the critical tasks for postdisaster relief is to collect urgent information quickly and safely to rescue lives and control damages. There have been a lot of research works on data collection with wireless sensor networks. However, research on vital data collection in disaster circumstances has been rare.

Some previous research employ wireless sensor network to gather useful data in a hostile environment like earthquake or volcano [2–5]. Suzuki et al. present a high-density earthquake monitoring system in [2]. The raw data about earthquake is gathered by a sink node and can be used for further analysis after earthquake. But the collected data is just about earthquake rather than survivors. To estimate the individual damage in personal area, a WSN system is proposed in [3] to provide useful information to predict the individual damage, which is not accurate to serve for rescuing. In [5], Cayirci and Coplu presents a wireless sensor network (i.e., SENDROM), in which nodes are randomly deployed before disaster occurs, for disaster relief operations management. In postdisaster relief, rescue teams use mobile central nodes to gather information such as survivor's location by querying the sensor nodes.

To collect data more efficiently, some works have studied hybrid networks for data collection in disaster situations [6–8]. These systems employ cellular systems (or wires systems) and sensor networks in parallel to achieve a superior performance, such as, high speed, high capacity, and wide area coverage. A hybrid network model in [9] collects damage assessment information from a large number of nodes, and its connectivity is maintained by an alternative route in the event of disasters. Fujiwara et al. employ a hybrid of sensor and cellular networks in [10]. They present a data collection system to detect damage in a disaster and to transmit the data to an emergency operation center. It applies the network scheme to a versatile data collection system using sensor networks for damage assessment and for victim detection beneath the rubble of collapsed buildings. However, in the hybrid network, the cellular network could be paralyzed by disasters quickly or congested by the sudden high load even if it survives so that the data collection system breaks down.

Among these works, they did not consider the possibility that some base stations of cellular networks or the sensor nodes might be collapsed or unreachable during or after disasters. In [13], authors presented a data collection framework which employs Ad hoc Relay Stations (ARSs). It can convey data from the collapsed area by sending them to the nearest ARSs. However, it is built on cellular networks, which would be destroyed immediately during disasters. Li and Liu present SASA [14], a Structure-Aware Self-Adaptive

wireless sensor network, for underground monitoring in coal mines. By regulating the mesh sensor network deployment and formulating a collaborative mechanism based on the regular beacon strategy, SASA is able to rapidly detect structural variations caused by underground collapses. The collapse holes can be located and outlined and the data can be transferred outside of the collapsed region. However, the stationary mesh network could be ruined and become unreliable when a collapse occurs.

To the best of our knowledge, this paper is the first one that considers a wireless sensor network under stress and evacuates the critical data to the safe zone for postdisaster relief operations.

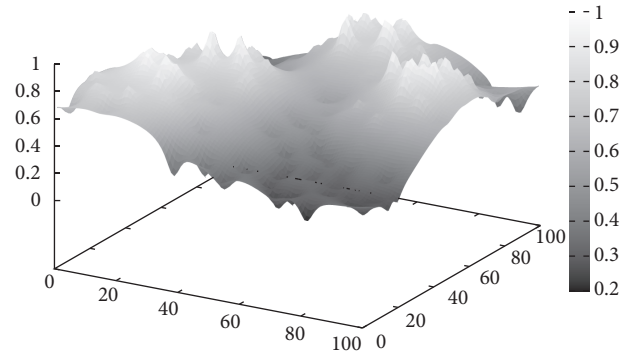


Figure 1: Devastating event intensity distribution in disaster.

3. System Models and Problem Description

3.1. Network Model. In this paper, we assume that N sensor nodes are randomly uniformly deployed in an $M \times M$ square area A, and the communication radius of sensor node is r. The network can be modeled as an undirected graph $G = (V, L)$, where V is the set of sensor nodes in the network, $|V|$ is the number of sensor nodes, and L is the set of links between sensor nodes in the network. For any two nodes v_i and v_j, if $\text{dist}(v_i, v_j) < r$, v_i and v_j are neighbors and there is a link $l(v_i, v_j)$ between them. For any node v_i, its neighbor node set is $\text{neighbor}(v_i)$.

In the event of a disaster, the capability of sensor nodes is assumed to be as follows.

(i) It can sense some meaningful event around. For example, sensor node can sense human vital signs through sound, infrared rays, temperature, image, and vibration sensors.

(ii) Sensor node can sense and measure the surrounding physical intensity (like the intensity of earthquake shock, temperature and smoke density in the fire, gas density before gas explosion, etc.) variation caused by a disaster.

(iii) Sensor node can rank itself as safe, critical, or dangerous, according to a predefined algorithm using the physical intensity variation it senses as inputs.

We do not assume the existence of a sink node that gathers all the data and routes to the relief center. When a serious disaster occurs, original communication infrastructure may be destroyed; even if some of them survive, they usually cannot provide effective service for disaster relief applications. In our approach, data evacuation is accomplished by collaborative efforts of every sensor node in the network to route the critical information to a few safe zones in the affected region.

3.2. Disaster Model. In this subsection, summarizing a set of common characteristics in most disasters, we construct a simplified disaster model, as follows.

Definition 1 (devastating event). We use a Quaternion $(C_i, I_i, T_i, \text{ and } A_i)$ to represent a devastating event E_i, where C_i is the centre point of the zone where the devastating event occurs, given by the coordinate (x_i, y_i); I_i is the intensity of the devastating event; T_i is the attenuation coefficient of disaster propagation; A_i is the region that the devastating event affects.

Definition 2 (disaster). Disaster is a set of devastating events and could be denoted by $D = \{E_i \mid 0 \le i \le d-1, d \in N\}$.

Let us look at an example. When a coal-mine accident (disaster) occurs, it probably consists of several gas explosions and water leak accidents, each of which corresponds to a devastating event. Each devastating event could be described by four elements: the position of event occurrence, the intensity of this event, the attenuation coefficient of this event, and the region affected by this event. The intensity I_i is the highest in the centre point of a devastating event and weakens as it gets further away from the center point. Usually T_i reflects the change of I_i in the region where disaster affects. There is no common attenuation coefficient for disasters. For simplicity, we assume a linear attenuation coefficient denoted as T_i. Under the impact of T_i, a devastating event can be depicted as a subarea of which the intensity is linearly descending from a centre point. As an example, Figure 1 illustrates a typical intensity distribution of a disaster with four devastating events, and the intensity is collected by sensors in the affected region. The centers of the four devastating events are (15, 25), (25, 40), (55, 85), and (85, 60). It can be seen that the intensity function has multiple sets of minimum points in its support (i.e., the affected region).

In this paper, according to the data that the sensor nodes collect, we define an algorithm to classify the state of the sensor node into three categories: safe, critical, and dangerous. Let $\text{intens}(v_i)$ be the intensity that the node v_i senses; I_s, I_d $(I_s < I_d)$ are two thresholds which are predefined according to the disaster scene. Then, we have

$$\text{rank}(v_i) = \begin{cases} \text{safe}, & \text{intens}(v_i) < I_S, \\ \text{critical}, & I_S \le \text{intens}(v_i) < I_D, \\ \text{dangerous}, & I_D \le \text{intens}(v_i). \end{cases} \quad (1)$$

Let V_S, V_C, and V_D represent the set of safe nodes, critical nodes, and dangerous nodes, respectively.

When a disaster occurs in a certain place, the disaster usually only affects a limited area near the center, and similar disaster damage often shares the same zone. According to

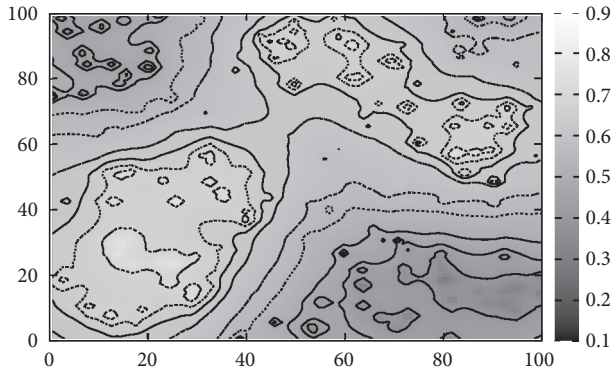


Figure 2: Vertical view of three zones distribution.

this, the three sets V_S, V_C, and V_D will have their own zones geographically, and since, after a disaster happens, there exists a short period of time when the sensor nodes collect the intensity data and rank themselves, the disaster area will be divided into several zones, which could be safe, or critical, or dangerous.

Definition 3 (zone). The sensor nodes in a zone has the same rank level, and they are connected. Consider the zone to be a connected subgraph $G_i^s = (V_i, L_i)$, $V_i \subseteq V$, $L_i \subseteq L$, $0 \leq i \leq g-1$, g is a positive integer and it represents the total number of zones in the whole area. For arbitrary two zones G_i, G_j, there are $V_i \cap V_j = \varphi$, $L_i \cap L_j = \varphi$, $1 \leq i < N$, $1 \leq j \leq N$, and $\bigcup_{i=1}^{g} V_i = V$. According to the rank level of each zone, we call it safe zone, or critical zone, or dangerous zone.

Figure 2 gives a vertical view for the disaster shown in Figure 1. Without the loss of generality, we adopt a normalized threshold of 0.5 for nonsafe zone in this paper and the threshold can be any value that manifests the physical meaning of a specific disaster (e.g., the Richter magnitude in earthquakes). In Figure 2, most area is covered by dangerous zone and critical zone (intens$(v_i) \geq 0.5$, for all $v_i \in V_C \cup V_D$), due to the devastating event's influence, and only a small area is covered by safe zone (intens$(v_i) < 0.5$, for all $v_i \in V_S$).

Sensor nodes in different ranks have varying surviving time, resulting in different roles in our data evacuation strategy. Sensor nodes in the dangerous zone have the shortest life time, only several seconds or dozens of seconds. Sensor nodes in the critical zone live longer, usually minutes or hours, because of less damage the devastating event causes in this zone, but the continuous damage will make the sensor nodes in critical zone ultimately destroyed. Sensor nodes in safe zone can live much longer, usually hours or days or longer, because of the long distance from the dangerous zone and the least damage the devastating event causes, so the sensor nodes in this zone are suitable to store valuable data for assisting personnel rescue and disaster analysis.

In our scheme, if one follows a piece of information, it normally traverses from dangerous zones, with possible route via critical zones, to two alternative destinies. It either arrives at some safe zone, or is trapped in dangerous/critical zones (lost in the end). For the former case, we adopt a definition for the path through which the information traverses, as follows.

Definition 4 (effective evacuation path). If a path $P = v_0' v_1' \cdots v_{i-1}' v_i'$, $1 \leq i \leq n$, $v_0' \in V_C \cup V_D$, and $v_i' \in V_S$ and such link $l(v_j', v_{j+1}')$, $0 \leq j \leq i-1$, $v_j' \in V_S \cup V_C$, and $v_{j+1}' \in V_D$ do not exist in path P, then P is the effective evacuation path.

3.3. Problem Formulation and Its Mathematical Structure. The end goal of our proposed data-rescuing strategy is to route the critical data sensed in the dangerous zone and critical zone to the safe zone for disaster relief and disaster analysis. The process of data evacuation can be expressed like this: for every sensitive data in any sensor node v, $v \in V_C \cup V_D$, data evacuation is to find an effective path and transmit the data to the safe zone. Any solution in this domain should have at least two desirable features. First, it should route as much information as possible. Second, data evacuation should be fast; otherwise the sensor nodes in the dangerous zone and critical zone could lose their data, or the sensor nodes in the effective evacuation path could be inactive.

As a manifest of the aforementioned features, we will focus on the following two performance metrics: (a) evacuation time: the time to complete the data evacuation process and (b) evacuation ratio: the percentage of whole sensitive data preserved in safe zone after finishing the data evacuation process. Data evacuation should be quick; otherwise the sensor nodes in the dangerous zone and critical zone could be damaged. As a result, some effective evacuation paths could fail to send the sensitive data to the safe zone. The data evacuation protocols need to guarantee that the amount of preserved sensitive data can provide enough useful information for postdisaster applications.

This formulation renders itself an elegant mathematical polymorphism. For each piece of information, it should strive to follow a path to any safe zone as fast as possible. If one considers the disaster intensity map as a two-dimensional function and any safe zone as a set of points with a minimum value, the data evacuation problem is equivalent to a non-linear programming problem with multiple (usually unknown) minimums in its support. This structural polymorphism with non-linear optimization will inspire the development of two efficient data-rescuing algorithms, both of which will be elaborated in the next section and are distributed in nature.

4. Data Evacuation Protocols

In this section, we present two alternatives DE protocols, each of which is a greedy algorithm seeking to optimize one design metric. The first protocol routes the critical information through effective evacuation paths following the highest gradient in the disaster intensity map, and thus denoted as GRAD-DE. The second protocol routes the critical information by effective evacuation paths leading to the closest safe zone with the largest storage capacity. Intuitively,

the information is attracted by the gravitation (proximity and capacity) of the safe zone, and thus the protocol is denoted as GRAV-DE. The GRAD-DE protocol performs better in minimizing the evacuation time, while the GRAV-DE protocol excels in maximizing the evacuation ratio.

4.1. GRAD-DE Protocol

4.1.1. Detailed Design of GRAD-DE Protocol. The GRAD-DE protocol stems from the Newton's method (gradient based) for non-linear programming problems. One of the potential issues with Newton's method is that it could converge to local minimums. In our protocol, we allow a few steps to route the information to nodes with higher intensity, so that the critical message will not be trapped. Here is how the protocol works.

First, each sensor node obtains the intensity and the rank level of all its neighbors through a round of hello-message exchange. In the event of any disaster, a sensor node first senses the intensity of devastating event and determines its rank level based on the predefined I_s and I_d. After that, it will broadcast a hello message, including its sensed intensity and self-determined rank level, to all neighbors.

Second, as water always flows downwards, in the GRAD-DE protocol, each sensor node forwards the sensitive data sensed locally or received from other nodes to its neighbor with the minimum sensed intensity. Obviously, in most cases, it is reasonable to send the sensitive data to the node with lower sensed intensity because it is the most logic step toward the safe zone (also suggested by the Newton's method). In order to avoid collision and reduce the communication cost, we adopt a single-copy forwarding strategy in the design.

This simple gradient-based forwarding strategy, however, could result in data trapped in stressed zones. For example, if a disaster consists of several devastating events, the sensed intensity ($>I_s$) of nodes in a certain region could be lower than the sensed intensity of any other nodes that surround this region. In this situation, the sensitive data of surrounding nodes may be forwarded to the "Highland Basin" as shown in Figure 3, and all sensitive data in this region will be trapped.

In order to avoid this problem (equivalently, the local minimums in non-linear programming problems), the GRAD-DE algorithm consists of three correction steps as follows.

Step 1. Upon receiving all the hello messages from its neighbors, each node marks itself if its intensity is lower than that of any other neighbor and broadcasts a warning message to prevent all its neighbors from sending sensitive data to it.

Step 2. When a node receives a warning message, if its all neighbors with smaller intensity have send warning messages to it, it will mark itself and broadcast a warning message to inform that it cannot play a relay role in an effective path. Otherwise, it just drops the received warning message.

Step 3. When a node has sensitive data to forward, it will check whether it has been marked. (a) If yes, it will send the sensitive data to the unmarked neighboring node with the lowest intensity. If it cannot find any unmarked neighboring node, it will send the sensitive data to the neighboring node with the highest intensity, with the hope that the data will escape from the trapped region from the trapped region as soon as possible. (b) If not, it means that this node is not located in a "Highland Basin" region, so the sensitive data can be sent to the unmarked neighboring node with the lowest intensity.

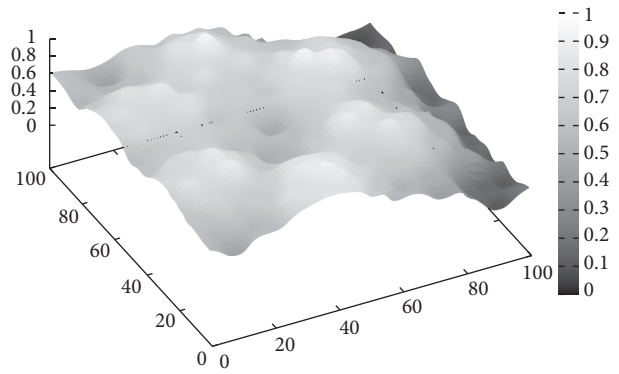


Figure 3: "Highland Basin" phenomenon caused by the coactions of devastating events.

4.1.2. Pros and Cons of GRAD-DE Algorithm. In this subsection, we will discuss the advantages and disadvantages of the GRAD-DE protocol, respectively.

On one hand, the GRAD-DE protocol comes with a few desirable characteristics. First, the control-message overhead for the GRAD-DE protocol is limited and upper bounded by two times of the total number of sensor nodes. In most cases, each node broadcasts a one-hop hello message to all its neighbors. Only when a "Highland Basin" problem appears will the nodes in this region broadcast an extra one-hop warning message to prevent sensitive data being transmitted to this region. As a result, even in the worst case, the number of control messages sent by one node is 2. Second, the GRAD-DE protocol does not rely on detailed information of the network topology. Specifically, each node simply sends sensitive data to its neighbor with the minimum intensity. As a result, the *evacuation time* will not be too long since we do not incur additional delay in topology discovery. Third, the GRAD-DE protocol is a scalable and distributed algorithm for data rescuing under stress, with some resemblance to the famous Newton's method in non-linear programming domain.

On the other hand, the GRAD-DE protocol has several drawbacks. For example, any effective evacuation path is predetermined by the intensity distribution in the affected region. If a relay node is damaged by devastating events, sensitive data transmission cannot be adapted to a new path. Although such an issue can be avoided by periodically sending hello messages, the control-message overhead would increase. Collision is another issue, which is caused by no topology control for the GRAD-DE protocol and cannot be solved thoroughly by relying on the IEEE 802.15.4 MAC protocol. Adjusting the time interval for data sending could

Table 1: Concepts mapping.

Concepts in physics	Concepts in data evacuation	Description
Object 1	Single sensitive message of $v_i(D_i^j)$	Single sensitive message can been seen as an object with unit mass
Object 2	Space of certain safe zone (space(G_j^s))	The mass of G_j^s is the storage size (messages) of this safe zone
$\|D\|$	Hops from v_i to G_j^s	Distance can be expressed as hops of evacuation path

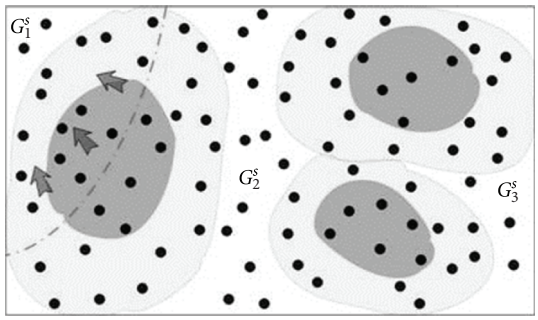


Figure 4: Buffer overflow due to blindly sending.

be a way to avoid collisions; however, such a strategy would pay the penalty of prolonging *evacuation time*. Buffer overflow is also a problem for the GRAD-DE protocol. As shown in Figure 4 where G_1^s, G_2^s, and G_3^s are three safe zones, the number of member nodes of these three safe zones are 2, 11, and 3, respectively. Unfortunately, the GRAD-DE protocol does not provide any information about safe zones, such as the number of safe zones and the storage capacity of safe zones. As a result of the blind data forwarding, Figure 4 illustrates that too many nodes blindly send their sensitive data to G_1^s and G_3^s, although the storage capacity of these two zones is limited.

4.2. GRAV-DE Protocol

4.2.1. General Principle. The GRAV-DE protocol is proposed to avoid the buffer overflow problem in the GRAD-DE protocol, which happens because each sensor node blindly forwards its data to a random safe zone. Indeed, we believe that the protocol would better off if the data is forwarded to the closest safe zone with the maximum storage capacity. Such a principle is similar to Newton's theory of gravitation. Intuitively, the movement of every single sensitive message to a certain safe zone can be regarded as it is attracted by the safe zone and moves to this zone along the direction of gravitational force between them if this gravitational force is bigger than that of any other force caused by different safe zones. As a result of this parallelism, one can see a natural mapping between concepts in data evacuation and those in physics, as summarized in Table 1, where D_i^j denotes the jth sensitive message of node v_i, and space(G_j^s) denotes the storage size of certain safe zone G_j^s.

The key for the GRAV-DE protocol is for each node in dangerous or critical zones to discover the information and distribution of safe zones, and then it can make a decision to send their sensitive data to an appropriate zone. For this object, the GRAV-DE protocol follows a three-step procedure.

(1) *Safe zone organization*: in our design, each connected component or isolated node with maximum sensed intensity $<I_s$ can be seen as a safe zone.

(2) *Safe-zone/evacuation-path broadcast*: after safe zones have been identified, they will advertise their existence by broadcasting announcement messages to all nodes in critical or dangerous zones.

(3) *Evacuation path decision*: nodes under stressed zones can determine the effective evacuation path for data evacuation, similar to the Newton's law of gravitation.

In next three subsections, we investigate the detailed implementation for the three steps in our proposed GRAV-DE protocol.

4.2.2. Safe Zone Organization. In this phase, the major task is to identify all connected components with the maximum sensed intensity $<I_s$. Each component will be organized as a cluster with one head, and the head node has the knowledge of the storage size of its cluster. This problem exhibits a strong resemblance to the gossip problem in [15].

Leveraging the rich set of results from the gossip protocols, we propose the following distributed algorithm for safe zone organization. At the initiation phase, each node v_i with intens(v_i) $< I_s$ sets its ID i as its component ID and broadcasts it to its all neighbors. Upon receiving other component IDs broadcasted by its neighbors, it will set its component ID to the minimum of its ID and all of its neighbors. Finally, the component ID of every node in the same connected subgraph will converge to a fixed ID. Algorithm 1 illustrates the safe zone organizing algorithm. The convergence and the correctness of this algorithm is omitted in this paper due to space limitation.

4.2.3. Safe-Zone/Evacuation-Path Broadcast. The next logical step is for each sensor node to obtain an evacuation path to all valid safe zones. Intuitively, this problem is equivalent to finding the set of the shortest paths from each sensor node in stressed zones to the list of safe zones. A rich set of research existed for this problem [16–18], among which the famous Dijkstra's algorithm [19] inspires us the following distributed implementation for the evacuation-path broadcast protocol.

The protocol works as follows. After the safe-zone organizing phase is completed, the head node will broadcast an announcement message, including its unique component ID, its storage size, and hops with initial value 0, to all nodes in critical or dangerous zones. When a node receives an announcement message, it will check whether it receives

(1) **if** (intens(v_i) < I_s) **then**
(2) component_id = i and broadcasts M_c^i message
(3) **while** (v_i receives M_c^j from its neighbors) **do**
(4) **if** (component_id > j) **then**
(5) component_id = j and broadcasts M_c^j message
(6) **else** drop M_c^j message
(7) **endwhile**
(8) **if** ($i \neq$ component_id) **then**
(9) send a join_msg to node with id = component_id
(10) **else** reveives join_msg from all members

Algorithm 1: Safe zone organizing algorithm.

a message from the same safe zone. If yes, it will compare the value of hops in this new message with that of former messages and record the announcement message with lower value and the forwarding neighbor; otherwise, it will record the message and the forwarding neighbor. After this step, it will rebroadcast its own record of safe zones. At last, every node in danger area will have the knowledge of evacuation paths to all safe zones.

4.2.4. Evacuation Path Decision. This step addresses how each node should choose its evacuation path to one of the safe zones, by adopting a similar criterion as in the Newton's theory of gravitation. When a node v_i with intens(v_i) > I_s receives announcement messages from all safe zones (for all G_i^s, $1 \leq j \leq m$), the gravitational force between one sensitive message D_i^j and G_j^s can be calculated using the following equation:

$$F\left(D_i^j, G_j^s\right) = \frac{G \cdot \text{space}\left(G_j^s\right)}{\left(\text{dist}\left(v_i, G_j^s\right)\right)^2}, \quad (2)$$

where G is a constant called the universal gravitation constant; space(G_j^s) denotes the storage size of G_j^s, and dist(v_i, G_j^s) denotes the hops from v_i to the corresponding border node of G_j^s. It then chooses the safe zone with maximum gravitational force as the destination for sensitive data evacuation of v_i. Note that there are other possible decision criteria, as long as it generates an index increasing with higher storage capacity and decreasing with longer distance. In our research, we focus on one representative criterion, derived from Newton's law of physics, but it is not necessarily optimal.

Notice that the location of head could affect the performance of the GRAV-DE protocol significantly. For example, as illustrated in Figure 5, the sensitive data of node A can choose an evacuation path from path 1 or path 2 to evacuate its sensitive data to a safe zone. If the head node is the destination of an effective evacuation path, the distances of A to G_1^s and G_2^s are 4 and 2 hops, respectively, and our criterion indicates that the sensitive data of A will be transmitted to G_2^s although the mass of G_1^s is 2.5 times the mass of G_2^s. A corrective measure we can apply is for the border nodes in a safe zone to reincarnate itself as the head one of its associated

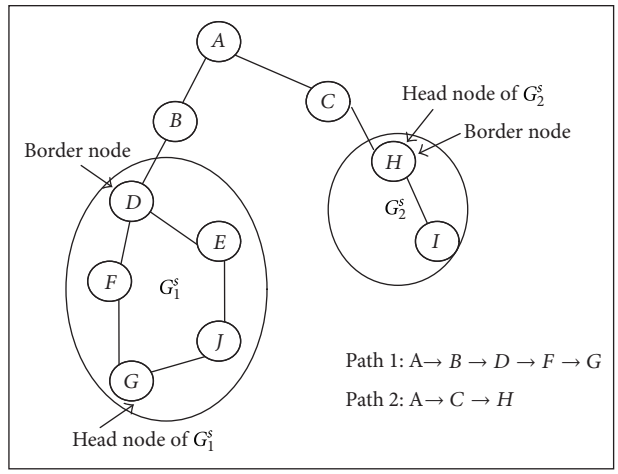


Figure 5: The impact of the location of head.

safe zone. In the same situation as in Figure 5, if we allow D, a border node, to be the destination of an evacuation path, the critical information will be transmitted along path 2.

5. Numerical Studies via Simulations

In our numerical study of data-evacuation strategies, we have implemented GRAD-DE and GRAV-DE protocols on an ns-2.33 simulation platform. We compare the performance of GRAD-DE and GRAV-DE protocols to a simple flooding approach in terms of *evacuation ratio* and *evacuation time*. In addition, we analyze the impacts of experimental parameters on the two proposed protocols.

5.1. Simulation Setup. In our simulations, the network size varies from 100 nodes to 900 nodes, and the area of monitoring region varies from 100 × 100 to 500 × 500. All sensor nodes have the same communication radius. Due to the limited bandwidth and the weakness of the collision avoidance mechanism of IEEE 802.15.4 MAC protocol, the sensitive message evacuation velocity of each sensor is assumed to follow a Poisson process with an average arriving interval of 1.5 s. To simulate the influence of disasters, we divide the whole network area into 2 × 3 small rectangles

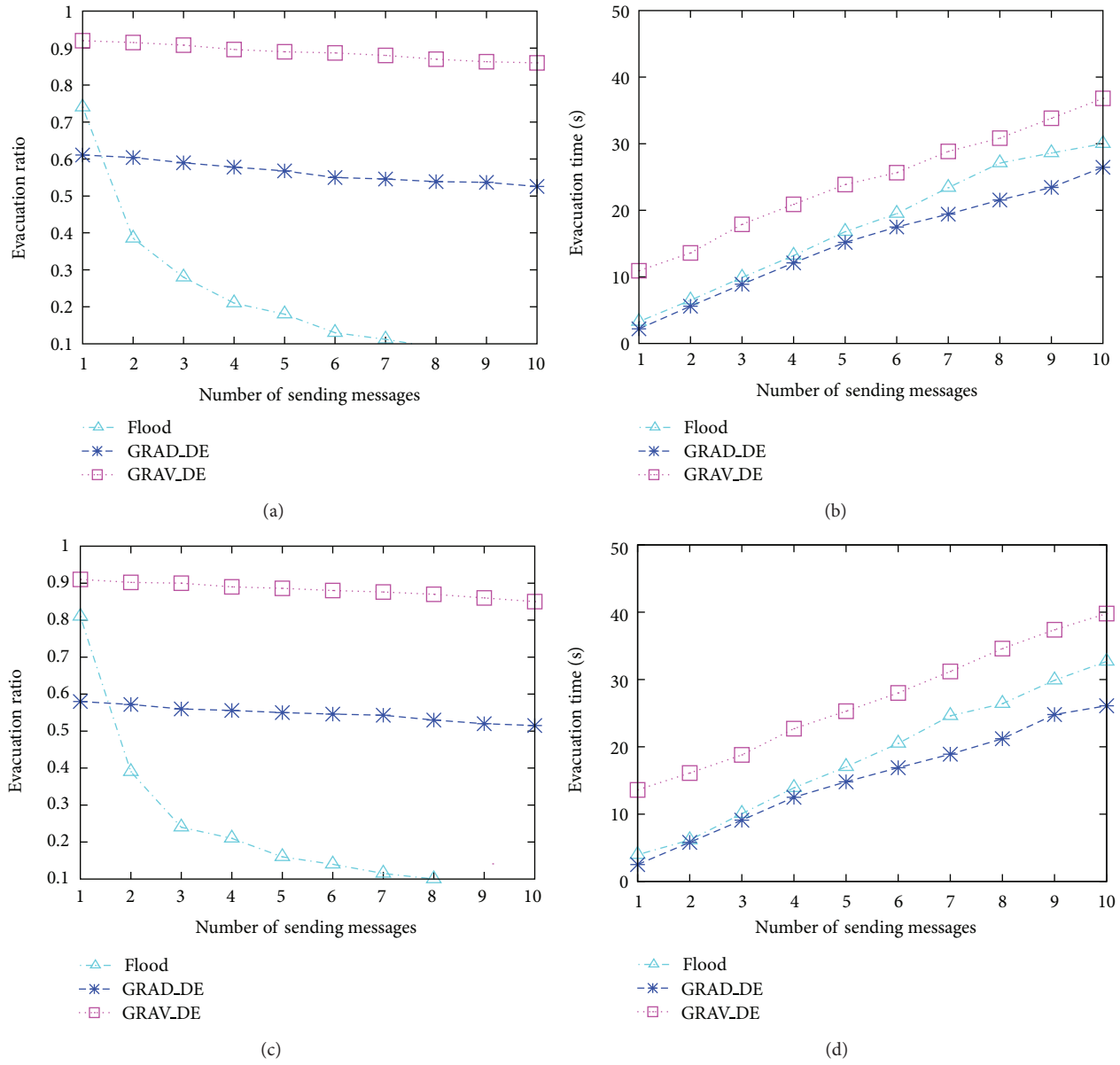


FIGURE 6: Impact of the number of sending sensitive messages on evacuation ratio and evacuation time with different network sizes: (a) average evacuation ratio (100 × 100), (b) average evacuation time (100 × 100), (c) average evacuation ratio (300 × 300), and (d) average evacuation time (300 × 300).

and put a devastating event in every small rectangle. The location of each devastating event is randomly chosen in the corresponding small rectangle. For simplicity, we presume that the intensity of the centre place of any devastating event is a real number between [0.8,1]. For any point $P(x, y)$ in network, the intensity of P caused can be calculated according to

$$\text{intens}(p) = \begin{cases} 1 - k\dfrac{\text{dist}}{M}, & \text{dist} < M, \\ 0, & \text{dist} \geq M, \end{cases} \tag{3}$$

where dist denotes the Euclidean distance between P and D; M is the longer side of the small rectangles.

5.2. Impact of the Number of Sensitive Messages. We first look at the performance of our proposed algorithms, with a rising number of sensitive data messages ranging from 1 to 10, for two network topologies (100 nodes, $100 \times 100\,\text{m}^2$ and 600 nodes, $300 \times 300\,\text{m}^2$). As a benchmark, we have also included a simple flooding protocol in our simulation.

Simulation results are summarized in Figure 6, which verifies our intuitions. First, we notice that the GRAV-DE protocol reaches a higher *evacuation ratio*, which outperforms both the GRAD-DE protocol and the flooding protocol. Specifically, the *evacuation ratio* for the GRAV-DE protocol is stabilized over 0.8, even in the worst case, as the number of message varies from 1 to 10. The reason

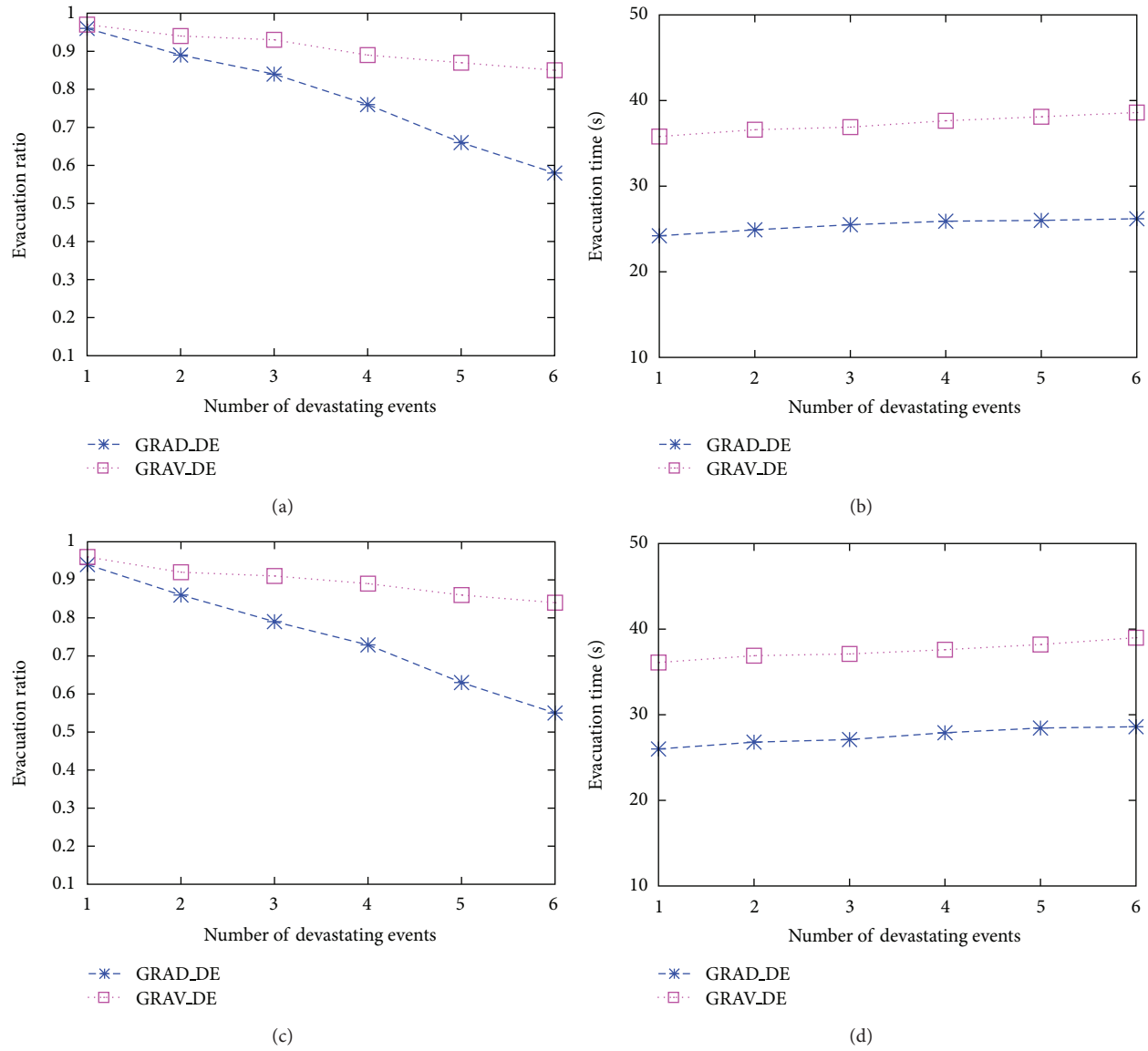


Figure 7: Impact of devastating event number on evacuation ratio and evacuation time with different network sizes: (a) average evacuation ratio (100×100), (b) average evacuation time (100×100), (c) average evacuation ratio (300×300), and (d) average evacuation time (300×300).

why the *evacuation ratio* of the GRAD-DE protocol is low is because too many messages could be forwarded to safe zones with smaller storage space. Second, the flooding algorithm has a higher evacuation ratio than the GRAD-DE protocol when the number of messages needed to be evacuated is very small. However, the *evacuation ratio* of the flooding approach drastically decreases and is lower than that of the GRAD-DE protocol as the number of messages increases. This observation can be traced back to two effects of the flooding algorithm. First, the chance of wireless collision is higher when flooding a lot of messages into the network; second, the storage space in safe zones will be occupied by replicated message soon. Third, as expected, the GRAV-DE protocol has a higher evacuation time than other two protocols, because it has to pay some time penalty in the two phases of safe-zone organization and evacuation-path broadcast.

5.3. Impact of the Number of Devastating Events. For simulating the different destruction degrees of disaster, we set a different number of devastating events on the network. Specifically, we randomly pick out n from 2×3 small rectangles and set a devastating event into each of the n small rectangle(s), where $1 \leq n \leq 6$.

From Figure 7, we see that both the GRAV-DE protocol and the GRAD-DE protocol obtain high *evacuation ratio* when the number of devastating events is small. As the number of devastating events increases, the evacuation ratio for the GRAD-DE protocol decreases faster than that of the GRAV-DE protocol. As a result, we argue that the GRAV-DE

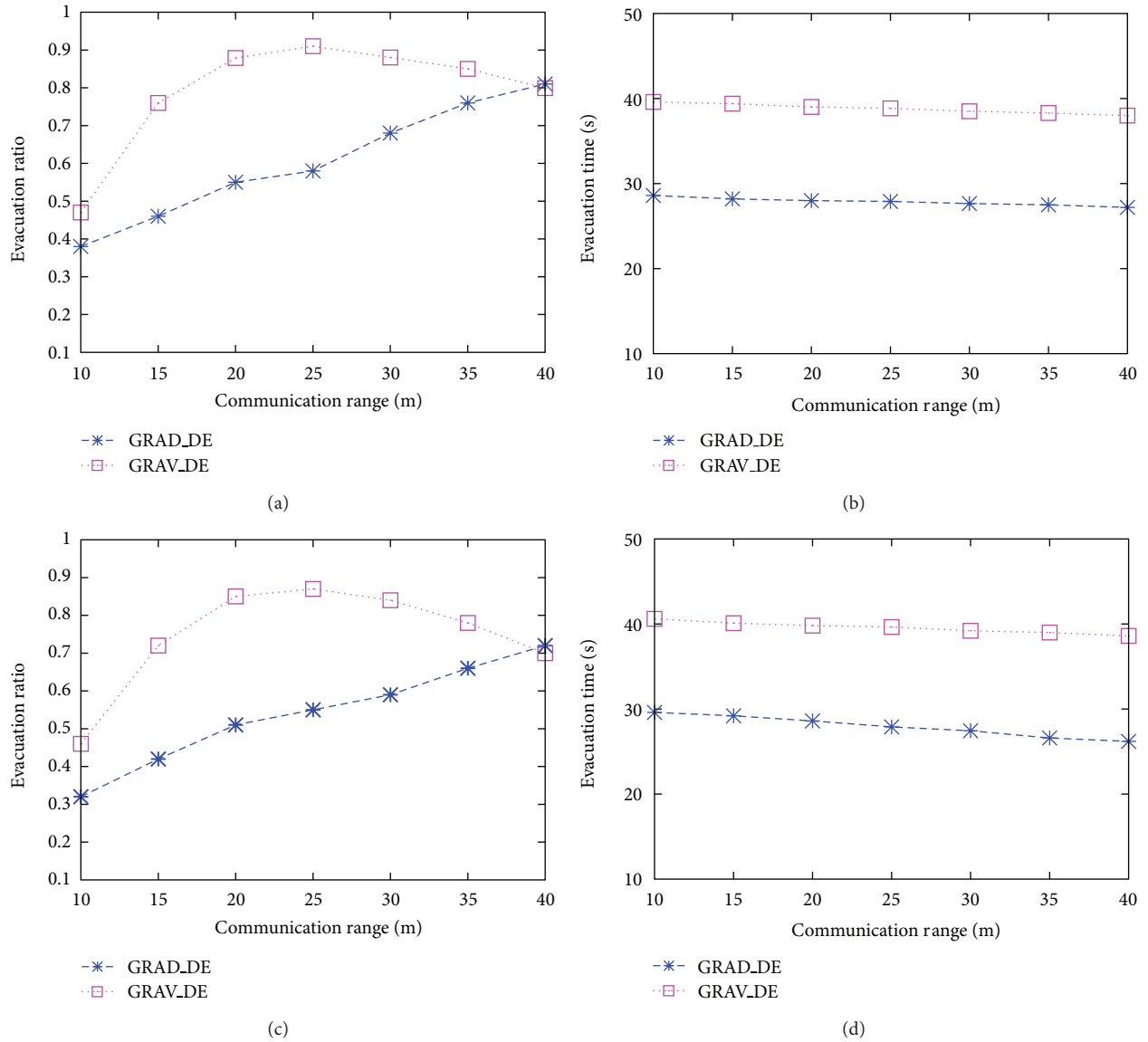


Figure 8: Impact of communication radius on evacuation ratio and evacuation time with different network sizes: (a) average evacuation ratio (100 × 100), (b) average evacuation time (100 × 100), (c) average evacuation ratio (300 × 300), and (d) average evacuation time (300 × 300).

protocol should be considered for efficient data evacuation in grievous disasters. We also notice that the increment of the number of devastating events brings about an increasing number of data messages need to be evacuated, and thus the completion time of data evacuation goes up slightly.

5.4. Impact of Communication Radius. The network connectivity is related to the communication radius of sensor nodes. In this subsection, we characterize the performance of the GRAV-DE algorithm and the GRAD-DE algorithm under different communication radius.

As shown in Figure 8, our proposed algorithms experience different performance trends as the communication radius increases. For the GRAD-DE algorithm the evacuation ratio increase monotonically as network connectivity improves. For the GRAV-DE algorithm the evacuation ratio first increases and then decreases as the communication radius increases. The main reason is that the rising communication radius increases the chance of wireless collision at the phase of organizing safe zones, which in turn results in a partial loss of the information of safe zones. From Figure 8, the *evacuation time* of both of the two proposed schemes descends slightly with the larger communication radius, since the average hop number from the node under stress to safe zones decreases.

5.5. Impact of Nodes' Survival Time. The performance of our proposed algorithm highly depends on the survival time of sensor nodes. In different types of disasters, nodes have different survival time in dangerous and critical zones. To evaluate the impact of nodes' survival time on the data evacuation performance, we vary the lifetime of nodes in

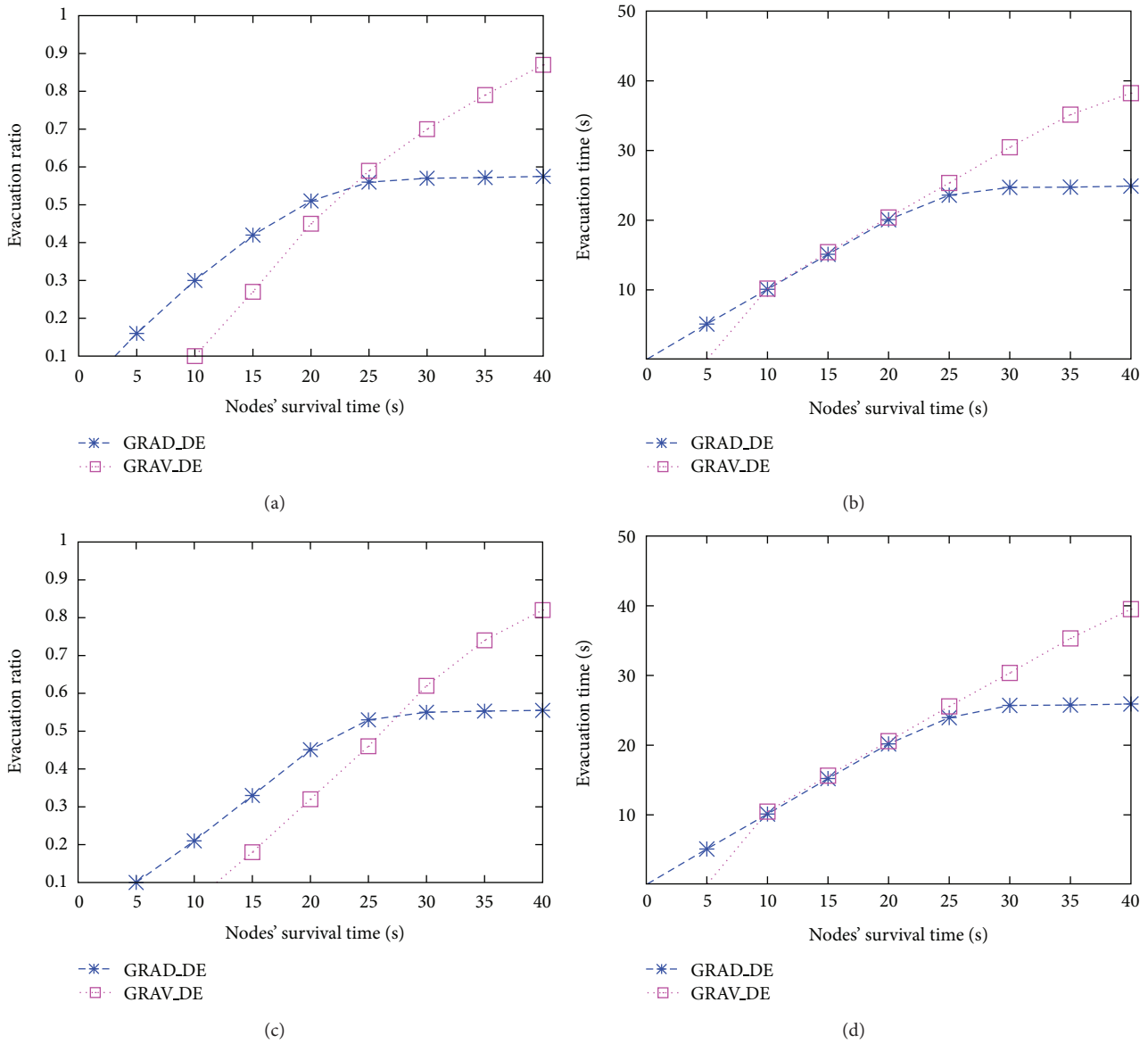


FIGURE 9: Impact of nodes' survival time on evacuation ratio and evacuation time with different network sizes: (a) average evacuation ratio (100 × 100), (b) average evacuation time (100 × 100), (c) average evacuation ratio (300 × 300), and (d) average evacuation time (300 × 300).

dangerous and critical zones from 10 s to 40 s. The results are shown in Figure 9.

When the nodes' survival time in dangerous and critical zones is too short, a large number of data messages cannot be evacuated to safe zones timely, so that both the GRAV-DE protocol and the GRAD-DE protocol have very low evacuation ratios. With the rising survival time, the data evacuation performance of both of the two schemes clearly improves. Because the evacuation time of the GRAD-DE protocol is much shorter than that of the GRAV-DE protocol, the evacuation ratio of the GRAD-DE protocol does not go up any more after the node's survival time reaches 25 s, whereas the evacuation ratio of the GRAV-DE protocol increases till the node's survival time rises to 40 s. As far as the *evacuation time* is concerned, the longer survival time of nodes in dangerous and critical zones means the larger number of data messages needed to be evacuated. Therefore, the *evacuation time* of both of the two schemes gently goes up as the nodes' survival time increases.

5.6. Blancing Raito-Time Tradeoff. As verified in Sections 5.2–5.5, the tradeoff between the evacuation ratio and the evacuation time can be balanced by judicially applying either the GRAD-DE algorithm or the GRAV-DE algorithm. The GRAD-DE algorithm outperforms the GRAV-DE algorithm in minimizing the evacuation time, while the GRAV-DE algorithm dwarfs the GRAD-DE algorithm in maximizing the evacuation ratio.

6. Conclusion

In this paper, motivated by the serious damages incurred by a few recent disasters around the globe, we have investigated on how to apply wireless sensor networks for postdisaster relief operations in a more realistic situation, where the sensor nodes could be paralyzed by the devastating events. Rather than relying on the sensor network for information gathering for a long time, we believe that a more relevant strategy would be to exploit the survival time of sensor nodes for transmitting critical information, for example, a snapshot of the affected region before the network is destroyed (similar to what blackbox preserves in flight accident), to safe zones in affected regions.

In this context, we formulate the data-evacuation problem with two competing design metrics: the evacuation ratio and the evacuation time. The former captures the amount of information rescued and the latter captures the time incurred in data rescue. Mathematically, this problem is similar to a non-linear programming problem with multiple minimums in its support. This structural parallelism inspired two alternative data-rescuing algorithms, both of which manifest some kind of principle derived by Newton. The GRAD-DE algorithm, named after its gradient-based approach, provides a superior time performance, but suffers from a throughput perspective, whiles the GRAV-DE algorithm, named after its resemblance to the law of gravitation, exhibits a higher throughput, but only takes much longer to rescue critical data. Our numerical study verifies the tradeoff between these two metrics. It is the field engineers' responsibility to judicially apply either algorithm in a realistic situation to rescue lives and/or control damages.

For future research, a direct extension of this work would be to compare different criteria to decide which evacuation paths to take. Another possible topic would be to make it possible to multiply evacuation paths for each sensor node and evaluate the associated tradeoff between the evacuation time and the evacuation ratio.

Acknowledgments

This work is supported by National Science Foundation under Grant numbers 61170256, 61103226, 60903158, 61173172, 61003229, and 61103227 and the Fundamental Research Funds for the Central Universities under Grant number ZYGX2010J074.

References

[1] http://www.unisdr.org/disaster-statistics/pdf/isdr-disaster-statistics-impact.pdf.

[2] M. Suzuki, S. Saruwatari, N. Kurata, and H. Morikawa, "A high-density earthquake monitoring system using wireless sensor betworks," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pp. 373–374, Sydney, Australia, November 2007.

[3] H. Miura, Y. Shimazaki, N. Matusa, F. Uchio, K. Tsukada, and H. Taki, "Ubiquitous earthquake observation system using wireless sensor devices," in *Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems (KES '08)*, vol. 5179 of *Lecture Notes in Computer Science*, 2008.

[4] G. Werner-Allen, K. Lorincz, M. Welsh et al., "Deploying a wireless sensor network on an active volcano," *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.

[5] E. Cayirci and T. Coplu, "SENDROM: sensor networks for disaster relief operations management," *Wireless Networks*, vol. 13, no. 3, pp. 409–423, 2007.

[6] W. Yang and Y. Huang, "Wireless sensor network based coal mine wireless and integrated security monitoring information system," in *Proceedings of the 6th International Conference on Networking (ICN '07)*, April 2007.

[7] Y. Yamao, T. Otsu, A. Fujiwara, H. Murata, and S. Yoshida, "Multi-hop radio access cellular concept for fourth-generation mobile communications system," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '02)*, pp. 59–63, Lisbon, Portugal, September 2002.

[8] H. Wu, C. Qiao, S. De, and O. Tonguz, "Integrated cellular and ad hoc relaying systems: iCAR," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2105–2115, 2001.

[9] T. Fujiwara, S. Nakayama, N. Iida, and T. Watanabe, "A wireless network scheme enhanced with ad-hoc networking for emergency communications," in *Proceedings of the 3rd IASTED International Conference on Wireless and Optical Communications*, pp. 604–609, Banff, Canada, July 2003.

[10] T. Fujiwara, H. Makie, and T. Watanabe, "A framework for data collection system with sensor networks in disaster circumstances," in *Proceedings of the International Workshop on Wireless Ad-Hoc Networks*, pp. 94–98, June 2004.

[11] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, Springer, 2008.

[12] I. Newton, *The Principia: Mathematical Principles of Natural Philosophy*, University of California Press, 1999.

[13] S. Suman and M. Mitsuji, "A Framework for data collection and wireless sensor network protocol for disaster management," in *Proceedings of the Communication Technology (ICCT '06)*, November 2006.

[14] M. Li and Y. Liu, "Underground coal mine monitoring with wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 2, article 10, 2009.

[15] N. J. T. Bailey, *The Mathematical Theory of Epidemics*, Griffen Press, 1957.

[16] B. V. Cherkassky, A. V. Goldberg, and T. Radzik, "Shortest paths algorithms: theory and experimental evaluation," *Mathematical Programming B*, vol. 73, no. 2, pp. 129–174, 1996.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, and All-Pairs Shortest Paths, *Introduction to Algorithms*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2009.

[18] D. Z. Chen, *Developing Algorithms and Software for Geometric Path Planning Problems*, ACM Computing Surveys, 1996.

[19] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 145727, 16 pages
http://dx.doi.org/10.1155/2013/145727

Review Article

Study on Routing Protocols for Delay Tolerant Mobile Networks

Haigang Gong and Lingfei Yu

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Haigang Gong; hggong@uestc.edu.cn

Received 11 October 2012; Accepted 6 December 2012

Academic Editor: Nianbo Liu

Delay tolerant mobile networks feature with intermittent connectivity, huge transmission delay, nodal mobility, and so forth. There is usually no end-to-end path in the networks and it poses great challenges for routing in DTMNs. In this paper, the architecture of DTMNs is introduced at first, including the characteristics of DTMNs, routing challenges, and metric and mobility models. And then, the state-of-the-art routing protocols for DTMNs are discussed and analyzed. Routing strategies are classified into three categories: nonknowledge-based approach, knowledge-based approach, and social-based approach. Finally, some research issues about DTMNs are presented.

1. Introduction

With the rapid development of low-power wireless communication technology and integrated circuit technology, there emerge a large number of low-cost, portable wireless devices. These devices are organized into a wireless ad hoc network and communicate with each other by multihop transmissions, which have great potential for many applications. For example, wireless sensor networks (WSNs) [1], composed of densely deployed low-power, low-cost sensor nodes, could be applied in scenarios such as military surveillance [2], disaster relief [3], health monitoring [4], environment monitoring [5], and smart home [6]. Another example is vehicular ad hoc networks (VANETs), in which vehicles equip with short range RF modules and exchange data when they meet, widely used in traffic safety [7], traffic efficiency [8], and information service [9].

Data gathering and routing is one of the fundamental functions of the low-power wireless ad hoc network and there have been lots of research works on routing issues [10–14]. However, authors assume that the network is full connected in these works, that is to say, there exists an end-to-end path between the source node and destination node, which is unreasonable in the real environment. In fact, if nodes are deployed randomly in the region, the density of nodes in some subregions would be higher than other subregions, leading to the phenomena of network partition, as shown in Figure 1. Once the network is partitioned, it is not fully connected any more. Secondly, the environment often has great impacts on the low-power communication. For instance, if there are electromagnetic fields or some obstacles, nodes will not communicate with each other even if they are within the transmission range, disconnecting the network. Thirdly, nodes are often powered by batteries, which is hard to rechargeable. When the energy of the battery exhausts, nodes cannot transmit data any more, degrading the network connectivity. Moreover, if nodes move with animals such as ZebraNet [15] and SWIM [16], data transmission only occurs when nodes meet each other. The mobility of nodes introduces opportunistic connectivity and there is not a stable end-to-end path in the network, leading to partially connected network.

Above all, the network is often not fully connected in the real environments and the network connectivity is intermittent and opportunistic, which is the characteristic of delay tolerant networks (DTNs) [17]. DTNs feature with sparse and intermittent connectivity, long and variable delay, high latency, high error rates, highly asymmetric data rate, and no stable end-to-end path. Obviously, traditional routing protocols are not well suitable for DTNs. For example, on-demand routing protocols such as AODV [18] and DSR [19] for MANET try to find an end-to-end path and table-driven routing protocols such as DSDV [20] and WRP [21] need to build route table. They are both hard to be adaptive to

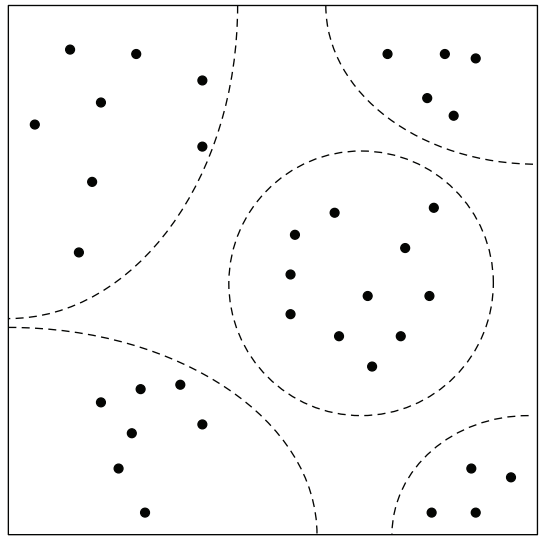


Figure 1: The phenomena of network partition.

the intermittent connectivity and dynamic network topology. New protocols must be designed for delay tolerant networks.

With the improvement of the portability of the wireless nodes, the mobility of nodes has been greatly improved. The enhanced mobility deteriorates the network connectivity further and challenges network routing. Harras et al. discuss the characteristics of delay tolerant mobile networks (DTMNs) and present routing issues of DTMNs in [22]. However, DTMNs are application specific and there are different types of DTMNs such as delay tolerant mobile sensor networks (DTMSNs) [23] composed of tiny sensor nodes, mobile social networks (MSNs) [24] when nodes attached to the human, and vehicular delay tolerant networks (VDTNs) [25]. Undoubtedly, there is not a universal routing protocol running in different types of DTMNs and routing protocols should be application specific, too.

The key issue of routing for DTMNs is to find an opportunistic connectivity between the nodes and transmit data to the nodes when they meet with each other if possible. Some methods have been proposed to achieve opportunistic communication in such challenged networks, trying to achieve the higher delivery ratio with the shorter delivery delay. Each of them has its own pros and cons and is just suitable in certain domains. Flooding is the simplest approach to transmit data to the destination but it wastes network resources extremely. In order to reduce the network overheads, some of them employ the history of contacts made by the nodes to route the data. Some other schemes try to forward messages to the neighbor node with the higher probability to communicate with the destination node. There are also some approaches that predict the behavior of the nodes and assist to route messages by the prediction knowledges. In addition, some other mechanisms are proposed, including infrastructure assisted method, that is, placement of stationary waypoint stores, using some mobile nodes to bridge the disconnection in the network, message replication, network coding, and leveraging prior knowledge of mobility patterns. Authors classify the routing protocols for delay tolerant networks into two categories: flooding-based approach and forwarding-based approach [26–28]. In [29], authors categorize the routing protocols into flooding-based method, history-based method, and special device-based method. In our opinions, the routing protocols should be divided into two categories: nonknowledge-based protocols and knowledge-based protocols. The former is to transmit messages to the next hop without any information indicating whether the next hop is an appropriate relay node. The latter relays messages with the assistance of the collected information about the network state and chooses a suitable next hop based on the knowledge. Moreover, social behavior analysis has been introduced to resolve the routing issues when the nodes are attached to the human and could achieve better performance by using social relationship or human mobility in real life environment, in which the routing schemes are called social-based protocols. In fact, the social-based protocols often utilize the knowledge of the social structure of the network and should be classified into the knowledge-based protocols. However, we discuss them separately from the former two categories in order to present routing schemes by using the social interaction of the nodes more clearly. In this paper, we study the existing routing protocols for DTMNs and give an analysis of them with respect to the important challenging issues and performance metrics.

The rest of the paper is organized as follows. Section 2 presents the architecture of DTMNs, including the characteristics of the DTMNs, routing challenges for the DTMNs, evaluation metrics of routing protocols for the DTMNs, and mobility model of the nodes. In Section 3, the states-of-the-arts of previous routing protocol for DTMNs are introduced and the existing problems are discussed. Section 4 presents some open issues about the DTMNs and Section 5 concludes the paper.

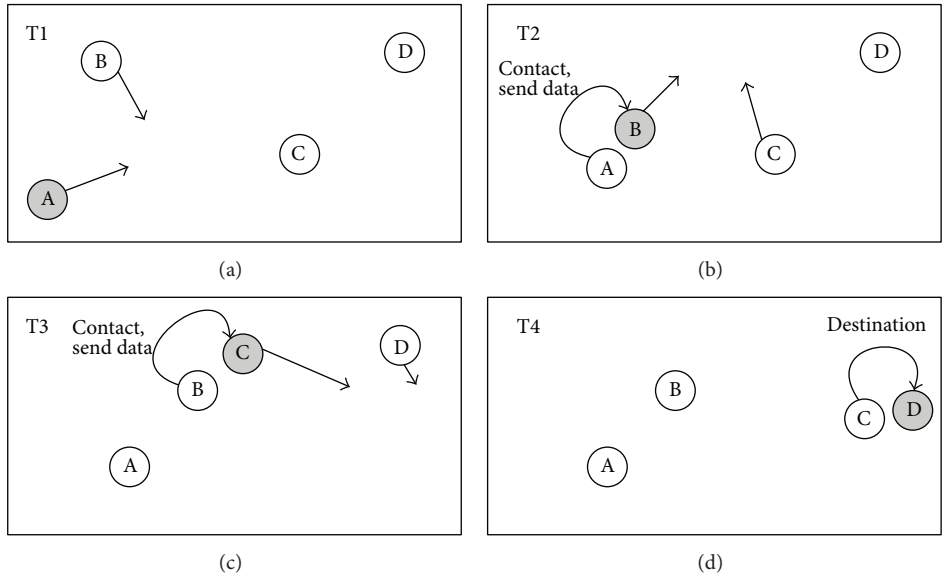


Figure 2: Data transmission in DTMNs.

2. Network Architecture

The concept of delay tolerant networking was initially proposed as an approach for the interplanetary Internet (IPN) [30]. Deep space communication suffers from very long latencies, low bandwidth, and intermittent scheduled connectivity. Fall proposes an overall architecture of DTN in [17], and it operates as an overlay above the transport layer to provide services such as in-network data storage and retransmission, and data forwarding. DTN technology has been introduced into wireless ad hoc network in the past few years. According to the mobility of the nodes, the network can be classified into two categories. (1) Network with some controllable nodes. In the network, most of the nodes are static and only a few movable nodes. The managed mobile nodes bridge the sparse disconnected network, store data from static nodes, and carry data to the destination. This method substantially saves the energy of the nodes as they only transmit over a short range. (2) Network with mobile nodes. In the second category, most of the nodes are movable and they have to transmit data occasionally when they contact with each other, introducing more challenges for routing messages. In this section, we firstly describe the characteristics of DTMNs, then analyze the routing challenges in DTMNs and metrics to evaluate the performance of the routing protocols, and finally discuss some mobility models, which have great influence on the network performance.

2.1. Characteristics of DTMNs. DTMN distinguishes itself from conventional networks by the following characteristics. (1) Intermittent connectivity. The connectivity of DTMNs is very poor. In most cases, it is impossible to have an end-to-end path. A node connects to other nodes only occasionally and the link is the scarcest resource in the network. (2) Delay tolerable. The end-to-end transmission latency is dominated by the queuing delay. Messages have to be stored in the message queue until the node meets a neighbor node. Obviously, opportunistic connection will lead to long latency so that applications have to tolerate the large transmission delay. (3) Sparse density. Node density is normally much lower in DTMNs compared with the traditional densely deployed networks, which further deteriorates network connectivity. (4) Node mobility. Since the nodes are attached to randomly moving objects, the network topology changes frequently. Besides, the buffer size of sensor nodes is usually limited. Since data messages may be stored in the buffer queue for quite a long time before being sent out, queue management is a challenge.

Clearly, a node only transmits its messages to the next hop when it meets other nodes and chooses an appropriate neighbor. As shown in Figure 2, node A wants to send message to node D at T1 but there is no connection between them. Node A has to store the messages and carries them while moving. Then node A contacts node B at T2 and node A will send the messages to node B because node B moves to node D. And then, node B meets node C at T3 and relays the messages to node C. Node C carries the messages and meets node D at T4, then the messages are sent to the destination node.

2.2. Routing Challenges. One of the main design goals of DTMNs is to exchange data between the nodes and employ the opportunistic links among the nodes for transmission. Clearly, the design of routing protocols in DTMNs is influenced by many challenging factors. In the following, we summarize some of the routing challenges that affect routing and forwarding in DTMNs.

2.2.1. Intermittent Connectivity. As mentioned before, intermittent connectivity is the inherent property of DTMNs. DTMN is a partially connected network because of node

mobility, sparse deployment, and poor communication quality. The network connectivity varies with time. Consequently, it is hard to find an end-to-end connection between the source node and the destination node so that routing techniques in conventional network are not well suitable for DTMNs. Intermittent connectivity means that the links between the nodes are opportunistic. How to get an opportunistic link and transmit a message is a challenging issue in DTMNs.

2.2.2. High Latency. High latency is also a fundamental property of DTMNs. In general, the transmission delay from a source node to a destination node is composed of four components: waiting time, queuing time, transmission delay, and propagation delay [31]. The waiting time is the interval that a message carried by node until it meets another node, depending on the contact time and the message arrival time. The queuing time is the time it waits for the higher priority messages to be sent out. This depends on the data rate and the traffics in the network. The transmission delay is the time it takes for all the bits of the message to be transmitted, which is determined by data rate and the length of message. The propagation delay is the time a bit takes to propagate across the connection, which depends on the distance between two nodes. Obviously, messages have to be buffered in the queue of the nodes due to intermittent connectivity, incurring more waiting time and queuing time. Moreover, the low data rate of DTMNs introduces more transmission delay. The design of routing protocols for DTMNs should reduce the delivery latency as shorter as possible.

2.2.3. Limited Resources. The nodes in DTMNs are often equipped with low-power RF module, limited buffer size, irreplaceable battery, and low computation capacity, that is to say, the resources of the nodes are limited. The scarce of resources degrades the performance of the routing protocols.

(1) Buffer Size. When a message is generated, the message is buffered in the message queue of the node. Once the node contacts other nodes, it chooses the next hop and delivers the messages in its queue. However, the node usually waits a long periods of time until it meets another node so that the messages have to be buffered in the queue. If the queue is full, some messages would be dropped off, which decreases the delivery ratio. Routing strategies might need to consider the limited buffer space when making routing decisions. In addition, there must be a scheme to manage buffer.

(2) Energy Efficiency. Nodes in delay tolerant mobile networks are usually powered by the battery, which cannot be replaced easily. Lots of energy will be consumed for sending, receiving, and computing. While researchers have investigated general techniques for saving power in delay tolerant networks [32], none of the routing strategies has incorporated energy-aware optimizations. In fact, most of the previous routing techniques do not consider the energy efficiency. In these works, the RF module of the nodes has to work all the time so as to find the possible links (opportunistic connectivity) to their potential neighbors. Then, the nodes will drain off their battery quickly and cannot contribute any more for routing, while degrading the performance of DTMNs. Therefore, there is a tradeoff between the energy consumption and network connectivity. How to maintain an acceptable connectivity while keeping the energy consumption slowly is a challenging routing issue for DTMNs.

(3) Process Capability. The nodes in DTMNs may be very small and have small processing capability, in terms of CPU and memory. These nodes will not be capable of running complex routing protocols. To design routing protocols for DTMSN, we must consider the computing capability of the nodes.

2.2.4. Replication Management. Since the connectivity between mobile nodes is poor, it is difficult to form a well-connected network for data transmission. The nodes deliver the message to their neighbors opportunistically when they contact. In order to achieve certain success delivery ratio in such an opportunistic network, data replication is necessary [33]. However, multiple copies of messages will increase transmission overhead, which is a substantial disadvantage for energy limited sensor networks. Replication management mechanism is necessary to control the number of message copies in order to reduce the overhead caused by redundant copies.

2.2.5. Network Topology. Due to nodal mobility or link quality, the network topology of DTMNs may change dynamically and randomly. It is impossible to maintain a stable end-to-end path in the networks, and routing in DTMNs is often on demand. Routing strategies designed for delay tolerant networks must be adaptive to the frequent change of network topology.

2.3. Routing Metrics. To evaluate the performance of the routing protocols for DTMNs, there are two main metrics: data delivery ratio and data delivery delay. Moreover, there are some other metrics to evaluate the performance of the routing strategies for application of specific DTMNs such as energy consumption, the number of replications, and network overhead.

2.3.1. Delivery Ratio. The most important performance metric is the data delivery ratio. Delivery ratio is defined as the fraction of all generated messages that are successfully transmitted to the destination within a specific time interval. In DTMNs, there are two factors to cause data loss. One is that the TTL of message exceeds the tolerable delay of the application. The network is unable to deliver messages within an acceptable amount of time. The second factor is that the queue of the node is full and some messages have to be dropped. If there are no any other copies of the dropped messages, these messages will not arrive at the destination forever. Routing protocols should achieve higher data delivery ratio.

2.3.2. Delivery Delay. Data delivery delay is another metric to evaluate the performance of routing strategies of DTMNs,

which is the time interval between when data is generated by the source node and when it is received by the destination. Due to the intermittent connectivity, the delivery delay of DTMNs is much longer than that of the conventional networks. Though applications in DTMNs can tolerate high latency, they can benefit from a short delivery delay. Some applications also have some time window where the data is useful. For example, if a DTN is used to deliver e-mail to a mobile user, the messages must be delivered before the user moves out of the network.

2.3.3. Energy Consumption. As described before, most of the existing routing protocols for DTMNs do not consider the energy efficiency. The RF module of node works all the time to search the possible links to other nodes, which exhausts the battery energy quickly. Once the battery is exhausted, the node is dead and cannot deliver any more data. However, for some data-centric applications, they want to gather data from the network as much as possible. That is to say, energy consumption of the node must be considered to design routing protocols in order to achieve longer network lifetime. The longer the network lifetime is, the more data the network collects. Routing techniques should make a tradeoff between the energy consumption and data delivery ratio.

2.3.4. Number of Replications. In order to get higher data delivery ratio, some routing protocols employ replication strategies and they transit more messages than others. The intuition is that having more copies of the message increases the probability that one of them will find its way to the destination and decreases the average time for one to be delivered. Unfortunately, the redundant replications waste a number of network resources such as buffer, bandwidth, and energy. The more the number of replications is, the more the wasted resources are. Routing protocols should achieve higher data delivery ratio with less replications.

2.3.5. Network Overhead. Usually, there are some control messages to assist in forwarding messages efficiently. These messages are network overheads. A good routing protocol should create little network overhead and it makes tradeoff between the data delivery ratio/delay and the delivery overhead.

2.4. Mobility Model. The mobility model is designed to describe the movement pattern of mobile users, and how their location, velocity, and acceleration change over time. Since mobility patterns may play a significant role in determining the performance of the routing protocols, it is desirable for mobility models to emulate the movement pattern of targeted real life applications in a reasonable way [34]. We classify the mobility models into four categories: random-based mobility model, social mobility model, map-based mobility model, and real dataset-based mobility model.

2.4.1. Random-Based Mobility Model. In random-based mobility models, the nodes move randomly without any restrictions. More specifically, the nodes choose their destination, speed, and direction randomly and independently of other nodes.

The simplest mobility model is the random walk mobility model [35], also called Brownian motion; it is a widely used model to represent purely random movements of the entities of a system in various disciplines from physics to meteorology. However, it cannot be considered as a suitable model to simulate wireless environments, since human movements do not present the continuous changes of direction that characterize this mobility model.

Another example of random mobility model is the random waypoint mobility model [36]. This can be considered as an extension of the random walk mobility model, with the addition of pauses between changes in direction or speed. When the simulation begins, each node randomly chooses a location in the field as the destination. It then moves towards the destination with constant velocity chosen randomly from $[0, V]$. The velocity and direction of the nodes are chosen independently of each other. On arriving at the destination, the node stops for a period of time and then chooses another random destination in the simulation field and moves towards it, as shown in Figure 3. The whole process is repeated again and again until the simulation ends.

The random waypoint model and its variants are designed to emulate the movement of mobile nodes in a simplified way. They are widely used due to their simplicity. However, they may not adequately capture certain mobility characteristics of some realistic scenarios, including temporal dependency, spatial dependency, and geographic restriction.

2.4.2. Social Mobility Model. In some types of DTMNs such as mobile social networks, the nodes are usually attached to the humans and carried by them. Apparently, the mobility of the nodes is determined by human decisions and social behavior. In order to emulate the social behavior, researchers propose social mobility model which is dependent on the structure of the relationships among people carrying the node.

Musolesi and Mascolo propose the community-based mobility model based on social network theory [37]. They think that a network consists of several communities, and the nodes are grouped into one community according to their social relationships among the individuals. The mobility of the nodes is also based on the social relationships. The model also allows for the definition of different types of relationships during a certain period of time (i.e., a day or a week). For instance, it might be important to be able to describe that in the morning and in the afternoon of weekdays, relationships at the workplace are more important than friendships and family ones, whereas the opposite is true during the evenings and weekends.

The idea of using communities to represent group movements in an infrastructure-based WiFi network has also been exploited in [38] and in its time-variant extension is presented in [39]. More specifically, this model preserves two fundamental characteristics, the skewed location visiting preferences and the periodical reappearance of nodes in the same location.

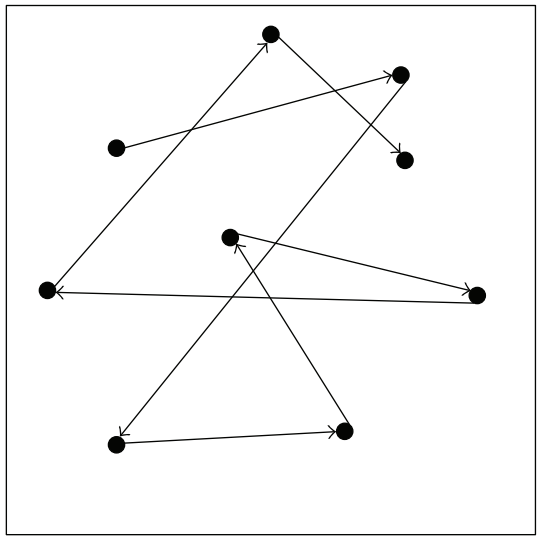


Figure 3: Random waypoint mobility model.

An agenda-based mobility model is proposed in [40], in which authors predict the movement of humans based on the trace of the people in the city. Then routing decision is made according to these information.

2.4.3. Map-Based Mobility Model. Map-based mobility model is designed for a specific type of DTMNs, vehicular delay tolerant networks. Different from the random-based mobility model, the movement of the nodes in vehicular delay tolerant networks is not random. The mobility is restricted by the road in the map and the nodes move regularly.

Freeway mobility model is proposed in [41] to emulate the motion behavior of mobile nodes on a freeway. It can be used in exchanging traffic status or tracking a vehicle on a freeway. The freeway mobility pattern is expected to have spatial dependence and high temporal dependence. It also imposes strict geographic restrictions on the node movement by not allowing a node to change its lane.

Manhattan mobility model is also introduced in [41] to emulate the movement pattern of mobile nodes on streets defined by maps in the city. The map is composed of a number of horizontal and vertical streets. Each street has two lanes for each direction. The mobile node is allowed to move along the grid of horizontal and vertical streets on the map. At an intersection of a horizontal and a vertical street, the mobile node can turn left, right, or go straight. The Manhattan mobility model is also expected to have high spatial dependence and high temporal dependence, but differs from the freeway model in giving a node some freedom to change its direction.

The obstacle mobility Model [42] takes a different approach in the objective to obtain a realistic urban network in presence of building constellations. Instead of extracting data from TIGER files, the simulator uses random building corners and Voronoi tessellations in order to define movement paths between buildings. It also includes a radio propagation model based on the constellation of obstacles. According to this model, movements are restricted to paths defined by the Voronoi graph.

2.4.4. Real Dataset-Based Mobility Model. In order to reflect the node behavior in real environment, some institutes try to collect a large number of real data reflecting the mobility of the nodes and their behavior. Based on these data, a real life mobility model could be built. For example, the reality mining project proposed by MIT [43] builds a system for sensing complex social systems with data collected from 100 mobile phones over the duration of 9 months. The collected data can be used to recognize social patterns in daily user activity, infer relationships, identify socially significant locations, and model organizational rhythms.

Haggle project proposed by Cambridge University [44] is an innovative paradigm for autonomic opportunistic communication. Students carry a tiny iMote with Bluetooth to record the contact history. Similarly, Hui and Crowcroft [45] created a human mobility experiment during IEEE Infocom 2006, with the participants labelled according to their academic affiliations. After collecting 4 days of data during the conference period, they replay traces using an emulator and discover that a small label indicating affiliation can indeed effectively reduce the delivery cost, without trading off much against delivery ratio. The intuition that simply identifying community can improve message delivery turns out to be true even during a conference where the people from different subcommunities tend to mix together.

3. Nonknowledge-Based Routing Protocols

In the nonknowledge-based routing approach, it tries to relay messages to the neighboring node without any information about the next hop. For example, the node does not know the likelihood that the next hop meets with the destination node, and the node chooses the next hop randomly or broadcasts.

Flooding is a mechanism which needs relay nodes to store and forward message copies independently through creating multiple duplications of a message in the network. This method could dramatically enhance delivery ratio and reduce average delivery delays at the cost of huge network resource consumption. Numerous optimization approaches have been presented based on flooding striving for reasonable resource consumption.

Direct transmission [46] is a typical nonknowledge-based routing technique. When source node generates messages, it carries the messages moving in the field. Once it contacts with the destination node, the messages are directly sent to the destination. Direct transmission is very simple and there is only one message copy and one transmission. However, the scheme does not employ the opportunistic links and suffers long delivery ratio due to the long waiting time in the buffer, especially when the source node is hard to meet the destination node.

In two-hop relay mechanism [27], the source node will send a message copy to the first n nodes it contacts. Then there is $n + 1$ node carrying the message and moving on. If any node holding the message encounters the destination node, the message will be delivered to the destination. Obviously, this method consumes more network resources, but it achieves better performance than direct transmission since it has better chance to communicate with the destination. For example, assuming that each node has an independent probability P to contact with the destination, then two-hop relay mechanism will deliver the message to the destination node with the probability $1 - (1 - P)^{n+1}$, which is far more than the probability of direct transmission when P is small. Moreover, it can choose the number of copies to control the resources consumption. However, two-hop relay mechanism has the same disadvantage as direct transmission, that is, if all the $n + 1$ nodes cannot encounter the destination node, the message cannot be transmitted.

Tree-based flooding method [27] improves two-hop relay by distributing the task of making copies to other nodes. When a message copy is transferred to a relay node, it will tell the relay node the number of copies it will generate. Because the relay nodes form a tree rooted at the source, the method is called tree-based routing. There are many ways to decide the number of copies the relay node will make. A simple scheme is to allow each node to make unlimited copies, but to restrict the message to travel a maximum of n hops from the source. Tree-based flooding can deliver messages to destinations that are multiple hops away, unlike direct contact or two-hop relay. However, tuning the parameters is a challenging problem.

Vahdat and Becker present epidemic routing in [47]. Epidemic routing works as follows. When a message is sent, it is still in the buffer with a unique ID. Once two nodes contact with each other, they exchange a summary vector including the list of all the messages IDs they have in their buffers. Then they exchange the message they do not have. Though Epidemic Routing uses the knowledge of summary vector, the knowledge does not indicate whether the next hop is the appropriate relay node. So we classify it into nonknowledge-based category. Epidemic Routing relies upon carriers coming into contact with another connected portion of the network through node mobility. At this point, the message spreads to an additional island of nodes. Through such transitive transmission of data, messages have a high probability of eventually reaching their destination. If the buffer size is large enough, the message will be distributed over the network like epidemic viruses until it arrives at the destination node. Epidemic Routing is relatively simple because it requires no knowledge about the network. Similarly to flooding, the disadvantage of Epidemic Routing is that a great amount of resources are consumed due to the large number of copies and requires large amount of buffer space, bandwidth, and energy.

Authors introduce the idea of immunity to improve the basic Epidemic Routing strategy in [48]. Each node maintains a list of delivered messages, called the immunity list. When two nodes contact with each other, they exchange their immunity lists at first, and then those messages in the immunity lists will not exchange in the future. It is expected to increase the number of delivered messages due to improved buffer and network utilization. Simulation shows statistically significant performance improvement both in delivery ratio and delay for immunity-based epidemic as compared to the basic epidemic protocol.

PREP is another improvement of Epidemic Routing proposed in [49]. The key idea of PREP is to impose a partial priority on the messages for transmission and dropping. The priority calculation is based upon four inputs: the current cost to destination, current cost from source, expiry time and generation time. Each link's average availability is epidemically disseminated to all nodes. As a result of this priority scheme, PREP maintains a gradient of replication density that roughly decreases with increasing distance from the destination. PREP is derived from the recognition that Epidemic routing is unbeatable from the point of view of successful delivery as long as the load does not stress the resources (bandwidth, storage).

Gossip [50] is also nonknowledge-based routing technique. Compared with flooding, Gossip tries to reduce network resources consumption by randomly choosing the relay node rather than delivering message to all nodes it meets. Clearly, the number of message copies is controlled and the resource consumptions decrease. However, randomly selected next hop might not be a suitable relay node and would make negative influence on the performance.

To significantly reduce the overhead of flooding-based schemes, Spyropoulos et al. propose spray and wait (SW) [51], which "sprays" a number of copies into the network at first, and then "waits" till one of these nodes meets the destination. SW routing strategy consists of two phases: spray phase and wait phase. In the spray phase, once a message is generated at source node, the number of message copies is confined by L. L message copies are forwarded by the source node and other relay nodes. If the L nodes with the message copies do not encounter the destination, then enter in to the wait phase. In the wait phase, the L nodes with the message copies performs direct transmission, that is to say, the L nodes carry the message copy till one of them contacts with the

destination. SW combines the speed of epidemic routing with the simplicity and thriftiness of direct transmission. At first, it spreads message copies in a manner similar to epidemic routing. When there are enough copies that at least one of them will find the destination quickly with high probability, it stops flooding and performs direct transmission.

Besides flooding-based routing techniques, there are other two types of nonknowledge-based routing strategies: special node-based approach such as SWIM [16] and data MULE [52], and coding based approach [53, 54].

The Shared Wireless Infostation Model (SWIM) architecture proposed by Small and Haas [16] employs the special node called Infostations at various locations. The Infostations are static, and the nodes are attached to moveable whales. Each Infostation is considered as a destination and they are connected. The mobile nodes forward data to the Infostations when they contact with any of the Infostations. So in effect, SWIM is similar to the epidemic scheme, except that in the SWIM, each Infostation is a destination.

Shah et al. [52] present a system called Data MULE. The special node in the system is called MULEs. The MULEs are mobile nodes and move around the sensor area randomly. The MULES try to collect data from the static sensor nodes and carry data back to the base station. Furthermore, there are some other routing schemes using special node to assist in forwarding message [55–61].

There are two types of coding-based strategies: network coding [53] and erasure coding [62]. The former embeds the decoding algorithms into the coded message blocks and the latter adds redundancy into the message blocks. In network coding-based strategies [53, 54], fragmentation and network coding taken are used to reduce resource consumption. In these strategies, each message is partitioned into K fragment packets when it is originated. Those fragments are flooded in the network, and relay nodes firstly combine the fragments and encode them into a new packet then forwarding. At last, when the destination obtains coded packets which collect all the K fragments, it attempts to decode the K source packets and the message is delivered. This method reduces the buffer and transmission consumption at the cost of long time waiting for the destination to receive a sufficient number of coded packets.

Chen et al. [62] apply erasure coding, but combine it with some replication techniques. Liao et al. [63] also propose a method where the message is erasure coded and then routed using estimation-based routing. The same authors further improve this approach in [64] by utilizing the knowledge of the mobility pattern of the network to route the erasure coded blocks.

4. Knowledge-Based Routing Protocols

Nonknowledge-based routing strategies relay message blindly and consume huge network resources. To forward messages efficiently, knowledge about the network could be used to optimize routing strategies and improve the performance. Knowledge about the network include link metric, history contact, mobility pattern, and network topology. According to the knowledge, a node can select the next hop which has the highest likelihood to communicate with the destination node.

4.1. Link Metric-Based Approach. Similarly to traditional networks, DTMNs can be considered as a graph and each link is assigned a weight. Then the shortest path algorithm such as Dijsktra's algorithm is run to get a best route. Link weights are based on some performance metric: the highest bandwidth, lowest latency, and the highest delivery ratio. In DTMNs, the most important metric is the delivery ratio, since the network must be able to reliably deliver data. A secondary metric is the delivery latency. Thus, the challenge is to determine a system for assigning link metrics that maximize the delivery ratio and minimize the delivery latency. Some metrics may also attempt to minimize resource consumption, such as buffer space or power.

Jain et al. utilize link metrics for routing in delay tolerant networks in [65]. Their object is to minimize the end-to-end delivery latency. The intuition is that this minimizes the amount of time that a message consumes to buffer space, and thus it should also maximize the delivery ratio since there is more space available for other messages. Their work uses a metric, that is, the time it will take for a message to be sent over each link. Since this value may depend on the time a message arrives at a node, the authors present a time-varying version of Dijkstra's shortest path algorithm.

Feng et al. propose minimum expected delay-based routing (MEDR) [66] protocols for delay tolerant mobile sensor networks. In MEDR, each sensor maintains two important parameters: minimum expected delay (MED) and its expiration time. According to MED, messages will be delivered to the sensor that has at least a connected path with their hosting nodes and has the shortest expected delay to communicate directly with the sink node. Because of the changing network topology, the path is fragile and volatile, so MEDR uses the expiration time of MED to indicate the time of the path and avoid wrong transmissions.

Jones et al. present a metric called the minimum estimated expected delay (MEED), where the weights are based purely on the history contact record [67]. MEED estimates the transmission delay to the next hop and assumes that the future delay will be similar to the past. The delay metrics are distributed over the network by an epidemic protocol. The node computes the shortest path based on all received link states of the network. MEED maintains a single message copy and selects the next hop with the shortest delay to the destination. Compared to direct transmission, MEED reduces the delivery delay efficiently. But MEED introduces more network overheads when distributing the link states over the network, especially when the network topology changes frequently.

Tan et al. [68] present a shortest expected path routing (SEPR) for DTN scenario. The forwarding probability of the link is calculated from the history of encounters. Based on this, the shortest expected path is calculated. The meet and visit routing (MV routing) proposed by Burns et al. [69] improves SEPR by using only the frequency of node contacts. It uses the frequency of the past contacts of nodes and also the visit to certain regions.

Moreover, Wang and Song propose a distributed real-time data traffic statistics assisted routing protocol (DRTAR) [70] for vehicular ad hoc network. In DRTAR, each vehicle estimates the state of the partitioned network of each road by the real-time statistics of records of neighbors. Based on the estimated delay of all roads, each vehicle can compute the appropriate routing path for message forwarding.

4.2. Prediction-Based Approach. To improve routing performance in opportunistic scenarios, prediction-based approaches have been designed for DTMNs. These approaches calculate and predict the state of network (i.e., message delivery probability, nodes' contact schedule, etc.) based on history information.

ZebraNet [71] is one of the earliest schemes to make routing decisions by the history of encounters. The object of the project is to monitor zebra movement in their habitat and wireless nodes are attached to the zebras. Each mobile node has a hierarchy level, which is calculated from the frequency of its contact with the base station. The hierarchy level of each node varies with time, depending on its frequency of contact with the base station. When a node encounters other nodes, it transmits the messages to another node with higher hierarchy level. In this way, the history of the node's encounter with the base station becomes the metric for data forwarding.

PRoPHET is probabilistic routing protocol proposed by Lindgren et al. [72]. PRoPHET uses the history of encounters to compute the delivery predictability of the nodes. The delivery predictability indicates the likelihood to meet the destination node. Each node maintains the delivery predictability of every other node for all known destinations. When nodes meet each other, they exchange the information of delivery predictability. Moreover, it also incorporates transitivity information to decide the next hop. PRoPHET has a higher delivery ratio than epidemic, with much lower communication overhead.

Spray and focus (SF) proposed in [73] improves spray and wait by substituting wait phase for focus phrase. The works of SF in the spray phase are the same as that of SW. In the focus phase, message carriers would select appropriate relay node based on predicted utility and then forward it. Spray and focus are demonstrated to achieve both good latency and low bandwidth overhead, thereby significantly reducing resource consumption in flooding routing.

PER proposed by Yuan et al. [74] predicts messages' delivery on the ground of probability distribution of future contact schedules and chooses a suitable next hop in order to improve the end-to-end delivery probability. In PER, a model based on a time-homogeneous semi-Markov process is designed to predict the probability distribution of the time of contact and the probability that the two nodes encounters in the future. When making decision, there are three metric functions for nodes in PER, which means nodes could select one of them to choose relay nodes.

Wang and Wu [75] present a replication-based efficient data delivery called RED, which consists of two components for data delivery and message management. Firstly, data delivery uses a history-based method like ZebraNet to calculate the delivery probabilities of sensor nodes. Secondly, the message management algorithm decides the optimal erasure coding parameters based on sensor's current delivery probability to improve the data delivery ratio. However, the optimization of erasure coding parameters used in [75] is usually inaccurate, especially when the source is very far away from the sinks. They also propose a FAD protocol in [76] to increase the data delivery ratio in DTMSNs. Besides using the same delivery probability calculation method as RED, FAD further discusses how to constrain the number of data replications in the sensor network by using a fault tolerance value associated with each data message. However, that protocol still has a quite high transmission overhead.

Xu et al. present a novel data gathering method named relative distance-aware data delivery scheme (RDAD) in [77]. RDAD introduces a simple non-GPS method with small overhead to gain the relative distance from a node to sink and then to calculate the node delivery probability which gives a guidance to message transmission. RDAD also employs the message survival time and message maximal replication to decide message's transmission and dropping for minimizing transmission overhead. Simulation results have shown that RDAD does not only achieve a relatively long network lifetime but also gets the higher message delivery ratio with lower transmission overhead and data delivery delay than FAD approach.

Similarly, a distance-aware replica adaptive data gathering protocol (DRADG) is proposed in [78]. DRADG economizes network resource consumption through making use of a self-adapting algorithm to cut down the number of redundant replicas of messages and achieves a good network performance by leveraging the delivery probabilities of the mobile sensors as main routing metrics.

So far, the routing techniques we discussed do not consider the energy efficiency of the network. However, for some data-centric applications, they want to gather data from the network as much as possible. That is to say, energy consumption of the node must be considered to design routing protocols in order to achieve longer network lifetime. The longer the network lifetime is, the more data the network collects. Routing techniques should make a tradeoff between the energy consumption and data delivery ratio.

Wang et al. develop a cross-layer data delivery protocol for DFT-MSN in [79]. They think that there is a tradeoff between link utilization and energy efficiency. The goal is to make efficient use of the transmission opportunities whenever they are available, while keeping the energy consumption at the lowest possible level. But the sleeping period of sensor nodes is determined by their working cycles and their buffered message. If a node moves around the sink and its sleeping period is too long according to [79], it will not deliver any data to the sink.

To make tradeoff between opportunistic connectivity and energy consumption, a data delivery protocol with periodic sleep (DPS) tailored for DTMSN is proposed in [80]. Based on their delivery probability and their distance to the sink, sensor nodes choose their sleep schedule to save the energy. The higher the delivery probability and the shorter the distance to the sink, the less the time they sleep in order to improve the connectivity around the sink. Simulation results

show that DPS achieves acceptable delivery ratio and delay with a very long network lifetime. In the long lifetime, the network can gather more data from sensor nodes than other approaches.

4.3. Context-Aware Approach. Some other protocols use the context information to aid in data forwarding. Musolesi et al. propose a context-aware adaptive routing (CAR) in [81], in which some context information such as the energy, moving speed, location, and communication probability are used to calculate utility. The node chooses the next hop that has the highest utility to transfer the messages. Based on CAR, Mascolo et al. present SCAR (sensor context-aware routing) [82], a routing approach which uses the context of the sensor node (history neighbors, battery level, etc.) to foresee which of the neighbors are the best relay nodes for data forwarding. In addition, SCAR controls the number of message copies like spray and wait.

Leguay et al. propose MobySpace [83], which utilizes the mobility pattern of nodes as context information. A MobySpace consists of Mobypoints. Each Mobypoint summarizes some characteristics of a node's mobility pattern. Nodes with similar mobility patterns are close in MobySpace. They are the optimum carriers of messages. The same concept on multicopy routing schemes is presented in [84].

Opportunistic routing with window-aware replication (ORWAR) is a resource-efficient protocol for opportunistic routing in delay-tolerant networks presented by Sandulescu and Nadjm-Tehrani [85]. ORWAR exploits the context of mobile nodes (speed, direction of movement, and radio range) to estimate the size of a contact window. This knowledge is exploited to make better forwarding decisions and to minimize the probability of partially transmitted messages. As well as optimizing the use of bandwidth during overloads, it helps to reduce energy consumption since partially transmitted messages are useless and waste transmission power. Another feature of the algorithm is the use of a differentiation mechanism based on message utility. This allows allocating more resources for high utility messages. More precisely, messages are replicated in the order of the highest utility first and removed from the buffers in the reverse order.

Grossglauser and Vetterli [86] propose another algorithm that was based on context information. Here the context information was the time lag between the last encounter with the destination. The main purpose of the work is to show that node mobility can be exploited to disseminate destination location information without incurring any communication overhead. To achieve this, each node maintains a local database of the time and location of its last encounter with every other node in the network. The database is consulted by packets to obtain estimates of their destination's current location. As a packet travels towards its destination, it is able to successively refine an estimate of the destination's precise location, because node mobility has "diffused" estimates of that location.

4.4. Position-Based Approach. Position-based routing (also called geographic routing) is a routing principle that relies on geographic position information, which is based on the idea that the source sends a message to the geographic location of the destination instead of using the network address. Position-based routing requires that each node can determine its own location and that the source is aware of the location of the destination. With this information, a message can be routed to the destination without knowledge of the network topology or a prior route discovery.

Greedy perimeter stateless routing (GPSR) presented by Karp and Kung [87] is a typical routing protocol for wireless ad hoc networks that uses the positions of routers and a packet's destination to make packet forwarding decisions. GPSR makes greedy forwarding decisions using only information about a router's immediate neighbors in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region.

Geographic source routing (GSR) [88] combines position-based routing with topological knowledge, as a promising routing strategy for vehicular ad hoc networks in city environments. Greedy perimeter coordinator routing (GPCR) [89] is a position-based routing protocol. The main idea of GPCR is to take advantage of the fact that streets and junctions form a natural planar graph, without using any global or external information such as a static street map. GPCR consists of two parts: a restricted greedy forwarding procedure and a repair strategy which is based on the topology of real-world streets and junctions and hence does not require a graph planarization algorithm.

5. Social-Based Routing Protocols

In the recent years, social structures have been used to help forwarding in intermittently connected networks. Social behavior analysis has been introduced to resolve the routing issues when the nodes are attached to the human and could achieve better performance by using social relationship or human behavior in real-life environment.

5.1. Social Relationship-Based Approach. In society, there are inherent social relationships between people such as relatives, friends, colleagues, and schoolmates. The relationships usually remain stable in a long period of time. Based on the social relationships, message could be forwarded efficiently.

Hui and Crowcroft have proposed a routing algorithm called LABEL which takes advantage of communities for routing messages [45]. LABEL partitions nodes into communities based on only affiliation information. Then ach node in the network has a label telling others about its affiliation. A node only chooses to forward messages to destinations, or to the next-hop nodes belonging to the same group (same label) as the destinations. LABEL significantly improves forwarding efficiency over oblivious forwarding using their dataset, but it lacks a mechanism to move messages away from the source when the destinations are socially far away.

BUBBLE combines knowledge of the community structure with knowledge of node centrality to make forwarding decisions [90]. Centrality in BUBBLE is equivalent to popularity in real life, which is defined as how frequently a node interacts with other nodes. People have different

popularities in the real life so that the nodes have different centralities in the network. Moreover, people belong to small communities like in LABEL. When two nodes encounter, the node forwards the message up to the node with higher centrality (more popular node) in the community until it reaches the same level of centrality as the destination node. Then, the message can be forwarded to the destination community at the same ranking (centrality) level. BUBBLE reduces the resource consumption compared to epidemic and PRoPHET. However, this reduction may not be large since the ranking process creates significant communication overhead. In addition, this protocol still uses multicopy forwarding which means that it is not efficient in terms of resource consumption.

SimBet presented in [91] makes routing decisions by centrality (betweenness) and similarity of nodes. Centrality means popularity as in BUBBLE. More specifically, the centrality value captures how often a node connects nodes that are themselves not directly connected [7]. Similarity is calculated based on the number of common neighbors of each node. SimBet routing exchanges the preestimated centrality and locally determined similarity of each node in order to make a forwarding decision. The forwarding decision is taken based on the similarity utility function (SimUtil) and betweenness utility unction (BetUtil). When the nodes contact with each other, the node selects the relay node with higher SimBet utility for a given destination.

SimBetAge [92] improves SimBet by introducing a new parameter, freshness. Routing decision is made based on freshness, betweenness, and similarity. Betweenness and similarity are the same as in SimBet and they are proportional to the freshness in SimBetAge. SimBetAge employs a weighted time-dependent graph, in which the weight of an edge is called the edge freshness, where $w(e,t) = 0$, $e = (\mathrm{A}, \mathrm{B})$ means that nodes A and B have not been connected from the initial time t_0 to time t and $w(e,t) = 1$ represents a permanent connection between A and B. The similarity of two nodes in SimBetAge is proportional to the freshness of a common neighbor between the two nodes. In order to have a more accurate calculation of betweenness compared to SimBet, SimBetAge takes all possible paths in a network into account, whereas SimBet only uses the shortest path between nodes.

LocalCom proposed by Li and Wu [93] is a community-based epidemic forwarding scheme in disruption tolerant network. LocalCom detects the community structure using limited local information and improves the forwarding efficiency based on the community structure. It defines similarity metrics according to nodes' encounter history to depict the neighboring relationship between each pair of nodes. A distributed algorithm, which only utilizes local information, is then applied to detect communities and the formed communities have strong intracommunity connections.

In social greedy [94], forwarding decision is made by the closeness and social distance. Closeness is calculated by the common attributes (address, affiliation, school, major, city, country, etc.) of the two nodes. The more common the attributes, the closer the two nodes. Social greedy forwards a message to the next node if it is socially closer to the destination. Social greedy outperforms the LABEL protocol. However, the delivery ratio of Epidemic and BUBBLE is better than social greedy.

PeopleRank approach [95] uses a tunable weighted social information to rank the nodes. PeopleRank is inspired by the PageRank [96] algorithm employed by Google to rank web pages. By crawling the entire web, the algorithm measures the relative importance of a page within a graph (web). Similar to the PageRank idea, PeopleRank gives higher weight to nodes if they are socially connected to other important nodes of the network. With the emergence of Online Social Network platforms and applications such as Facebook, Orkut, or MySpace, information about the social interaction of users has become readily available. Moreover, while opportunistic contact information is changing constantly, the links and nodes in a social network remain rather stable. The idea of PeopleRank is to use this more stable social information to augment available partial contact information in order to provide efficient data routing in opportunistic networks.

5.2. Human Behavior-Based Approach. Another social-based routing strategy employs the regularity of human behavior to aid in routing decision.

Liu and Wu present a cyclic MobiSpace [97], which is a MobiSapce where the mobility of the node exhibits a regular cyclic pattern as there exists a common motion cycle for all nodes. In a cyclic MobiSpace, if two nodes were often in contact at a particular time in previous cycles, then the probability that they will be in contact around the same time in the next cycle is high. Cyclic MobiSpace is common in the real world: (1) most objects' motions exhibit regularity as they are repetitive, time sensitive, and location related; (2) a common motion cycle usually exists because most objects' motions are based on human-defined or natural cycles of time such as hour, day, and week nodes. Based on this phenomenon, routing in cyclic MobiSpace (RCM) scheme is proposed. Routing decision is made by the expected minimum delay (EMD), which is the expected time that an optimal forwarding scheme takes to deliver a message at a specific time from a source to a destination, in a network with cyclic and uncertain connectivity. When nodes contact, messages would be relayed to the next hop with minimum EMD.

Liu et al. consider that there are preference locations that people visit frequently and they propose preference location-based routing strategy (PLBR) [98]. Firstly, PLBR provides the approach of acquiring one's preference locations and then calculates the closeness metric which is used to measure the degree of proximity of any two nodes proposed. On the basis of that, the data forwarding algorithm is presented. The closeness is defined to indicate the similarity of the preference locations that the two nodes visit. The higher the closeness of the two nodes, the more the common preference locations. If the closeness of the two nodes is high, the probability of the two nodes to contact is high. The messages would be forwarded to the next hop with the highest closeness. However, the calculation of the closeness requires the preference locations of the destination node, introducing large network overheads.

An expected shortest path routing (ESPR) [99] scheme improves PLBR by utilizing the stable property of human

that they have preference locations in their mobility traces, and the direct distance between node pairs can be calculated according to the similarity of their location visiting preferences. Then an expected shortest path length (ESPL) can be achieved by Dijkstra algorithm. Messages are forwarded to nodes which are closer to the destination than the previous nodes in the message delivery history. In addition, ESPR also employs the priority of message in the queue management.

CSI [100] is a behavior-oriented service as a new paradigm of communication in mobile human networks, which is motivated by the tight user-network coupling in future mobile societies. In such a scenario, messages are sent to the inferred behavioral profiles, instead of explicit IDs. At first, user behavioral profiles are constructed based on traces collected from two large wireless networks, and their spatiotemporal stability is analyzed. The implicit relationship discovered between mobile users could be utilized to provide a service for message delivery and discovery in various network environments. CSI shows that user behavioral profiles are surprisingly stable. Leveraging such stability in user behaviors, the CSI service achieves delivery rate very close to the delay-optimal strategy with minimal overhead.

Hot area-based routing protocol (HARP) scheme presented in [101] is based on the observation that there are some hot areas with higher nodal density and the node in the hot area has higher delivery probability to the destination. In HARP, the delivery probability is determined by the transmission ranking and the popular degree. Transmission ranking indicates the likelihood that sensor nodes communicate with the sink nodes. And popular degree reflects the popularity of sensor nodes. In the real world, some nodes may be more popular and interact with sink nodes more often than others in the network. The more hot areas a sensor node visits, the higher its popular degree is.

6. Open Issues

6.1. Energy Efficiency. Energy efficiency is an important issue for wireless ad hoc networks and there are lots of researches on energy efficiency in traditional wireless ad hoc networks such as WSNs. However, the existing routing strategies for delay tolerant networks seldom consider the energy consumptions of the nodes, shortening the network life time. In the previous works for DTMNs, the RF module of nodes has to work all the time so as to find the possible links (opportunistic connectivity) to their potential neighbors. Then, the nodes will drain off their battery quickly and cannot contribute any more for data gathering. Therefore, routing protocol should make a tradeoff between the energy consumption and data delivery ratio. In DTMNs with intermittent connectivity, it is helpful to find the link to keep the radio working all the time at the cost of rapidly exhausted battery. On the contrary, periodically working of the RF module saves energy but leads to lower connectivity. How to maintain an acceptable connectivity while keeping the energy consumption slowly is a challenging issue for DTMNs.

6.2. Security Routing. Existing routing protocols for DTMNs focus on improving the delivery ratio and reducing the delivery delay, but do not consider security issue. To our knowledge, there is little study on the security of data delivery in DTMNs. In DTMNs with intermittently connectivity, it is argued that the issue of security and privacy is not so important. In fact, DTMNs face all the security threats that a traditional network faces. Just like PC, the nodes in DTMNs are often controlled by people. There might be some malicious attackers using the nodes to transmit bad data in the network. So, security routing is a promising issue for DTMNs.

6.3. Selfish Routing. In the previous routing techniques, there is a common assumption that all nodes in the network are unselfish and coordinated. Each node is willing to receive and relay the messages sent by other nodes. In fact, there would be some selfish nodes, which want to preserve their own resources while using the services of others and consuming the resources of others, especially in social network. In the real world, most people are socially selfish, that is, they are willing to forward packets for nodes with whom they have social ties but not others, and such willingness varies with the strength of the social tie. Social selfishness will affect node behaviors. As a forwarding service provider, a node will not forward packets received from those with whom it has no social ties, and it gives preference to packets received from nodes with stronger ties when the resource is limited. Thus, a DTMNs routing algorithm should take the social selfishness into consideration.

6.4. Social Routing. Utilizing the social behavior and relationship of humans is still a very promising research area. In fact, the handheld devices are more and more popular today so that most of people carry some handheld devices. Knowledge of the social structure of people can enable these devices to be the bridge between the disconnectedness and to forward message more efficiently.

6.5. Cross-Layer Design. Generally speaking, cross-layer design refers to protocol design done by actively exploiting the dependence between protocol layers to obtain performance gains. This is unlike layering, where the protocols at the different layers are designed independently. Knowledge has to be shared between layers to obtain the highest possible adaptivity. So, how to use the information of other layers to assist in routing decision and optimizing forwarding is an interesting topic.

7. Conclusion

In this paper, we introduce delay tolerant mobile networks and discuss the characteristics of the DTMNs. The intermittent connectivity of DTMNs influences the routing performance significantly. Then routing challenges and routing metrics are analyzed. Moreover, mobility models are also discussed because they affect the forwarding efficiency directly. Then we study the existing routing protocols for delay tolerant mobile networks in depth. The routing strategies for DTMNs are categorized into nonknowledge based, knowledge based, and social based. In fact, it is not possible to classify each

of the schemes into exactly one of the many classes. More and more routing techniques are hybrid in nature and may be categorized into more than one category. Finally, we give some research issues about routing for DTMNs, including energy efficiency, security routing, selfish routing, social routing, and cross-layer design.

Acknowledgments

This work is supported by National Science Foundation under Grant no. 60903158, 61003229, 61103226, 61170256, 61173172 and the Fundamental Research Funds for the Central Universities under Grant no. ZYGX2010J074 and ZYGX2011J073.

References

[1] 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.

[2] G. Simon, M. Maroti, A. Ledeczi et al., "Sensor network-based countersniper system," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 1–12, Baltimore, Md, USA, November 2004.

[3] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for smart dust," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '99)*, pp. 271–278, 1999.

[4] T. Gao, D. Greenspan, M. Welsh, R. R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS '05)*, pp. 102–105, September 2005.

[5] M. A. Batalin, M. Rahimi, Y. Yu et al., "Call and response: experiments in sampling the environment," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 25–38, November 2004.

[6] N. Noury and T. Herve, "Monitoring behavior in home using a smart fall sensor," in *Proceedings of 1st Annual International Conference on Microtechnologies in Medicine and Biology*, pp. 607–610, 2000.

[7] Q. Xu, T. Mak, J. Ko, and R. Sengupta, "Vehicle-to-vehicle safety messaging in DSRC," in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks (VANET '04)*, pp. 19–28, October 2004.

[8] J. Eriksson, H. Balakrishnan, and S. Madden, "Cabernet: vehicular content delivery using WiFi," in *Proceedings of the 14th Annual International Conference on Mobile Computing and Networking (MobiCom '08)*, pp. 199–210, September 2008.

[9] R. Panayappan, J. M. Trivedi, A. Studer, and A. Perrig, "VANET-based approach for parking space availability," in *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks (VANET '07)*, pp. 75–76, September 2007.

[10] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks," *Wireless Networks*, vol. 8, no. 2-3, pp. 169–185, 2002.

[11] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.

[12] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '00)*, pp. 3005–3014, IEEE Computer Society, Maui, Hawaii, USA, January 2000.

[13] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2–16, 2003.

[14] A. Manjeshwar and D. P. Agrawal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings of the 15th International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, pp. 2009–2015, IEEE Computer Society, San Francisco, Calif, USA, April 2000.

[15] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 96–107, October 2002.

[16] T. Small and Z. J. Haas, "The shared wireless infostation model: a new ad hoc networking paradigm," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pp. 233–244, June 2003.

[17] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)*, pp. 27–34, August 2003.

[18] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, February 1999.

[19] D. Johnson and D. Maltz, "Dynamic source routing in ad-doc wireless networks," *Mobile Computing*, vol. 36, pp. 153–181, 1996.

[20] C. E. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers," *Computer Communication Review*, vol. 24, no. 4, pp. 234–244, 1994.

[21] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *Mobile Networks and Applications*, vol. 1, no. 2, pp. 183–197, 1996.

[22] K. A. Harras, K. C. Almeroth, and E. M. Belding-Royer, "Delay tolerant mobile networks (DTMNs): controlled flooding in sparse mobile 9 networks," in *Proceeding of 4th IFIP International Conference on Networking Technologies, Services, and Protocols*, pp. 1180–1192, 2005.

[23] Y. Wang, F. Lin, and H. Wu, "Poster: efficient data transmission in delay fault tolerant mobile sensor networks," in *Proceeding of IEEE International Conference on Network Protocols*, pp. 1021–1034, 2005.

[24] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN '05)*, pp. 244–251, ACM, Philadelphia, Pa, USA, August 2005.

[25] P. Pereira, A. Casaca, J. Rodrigues, V. Soares, J. Triay, and C. Cervello-Pastor, "From delay-tolerant networks to vehicular delay-tolerant networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1166–1182, 2011.

[26] J. Shen, S. Moh, and I. Chung, "Routing protocols in delay tolerant networks: a comparative survey," in *Proceedings of the 23rd International Technical Conference on Circuits/Systems, Computers and Communications*, p. 1577, 2008.

[27] E. P. Jones and P. A. Ward, "Routing strategies for delay tolerant networks," *Computer Communication Review*. In press.

[28] M. Liu, Y. Yang, and Z. Qin, "A survey of routing protocols and simulations in delay-tolerant networks," in *Proceedings of the 6th International Conference on Wireless Algorithms, Systems, and Applications (WASA '11)*, vol. 6843 of *Lecture Notes in Computer Science*, pp. 243–253, 2011.

[29] R. J. D'Souza and J. Jose, "Routing approaches in delay tolerant networks: a survey," *International Journal of Computer Applications*, vol. 1, no. 17, pp. 8–14, 2010.

[30] S. Burleigh, A. Hooke, L. Torgerson et al., "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine*, vol. 41, no. 6, pp. 128–136, 2003.

[31] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '04)*, vol. 34, pp. 145–158, ACM Press, October 2004.

[32] H. Jun, M. H. Ammar, and E. W. Zegura, "Power management in delay tolerant networks: a framework and knowledge-based mechanisms," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '05)*, pp. 418–429, September 2005.

[33] H. Wu, Y. Wang, H. Dang, and F. Lin, "Analytic, simulation, and empirical evaluation of delay/fault-tolerant mobile sensor networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 9, pp. 3287–3296, 2007.

[34] F. Bai and A. Helmy, "A survey of mobility modeling and analysis in wireless adhoc networks," in *Wireless Ad Hoc and Sensor Networks*, Springer, 2006.

[35] C. Bettstetter, "Mobility modeling in wireless networks: categorization, smooth movement, and border effects," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 3, pp. 55–66, 2001.

[36] C. Bettstetter, H. Hartenstein, and X. Pérez-Costa, "Stochastic properties of the random waypoint mobility model," *Wireless Networks*, vol. 10, no. 5, pp. 555–567, 2004.

[37] M. Musolesi and C. Mascolo, "A community based mobility model for ad hoc network research," in *Proceedings of the 2nd International Workshop on Multi-Hop Ad Hoc Networks: From Theory to Reality (REALMAN '06)*, pp. 31–38, ACM Press, May 2006.

[38] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '06)*, pp. 49–60, ACM, New York, NY, USA, 2006.

[39] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 758–766, May 2007.

[40] Q. Zheng, X. Hong, and J. Liu, "An Agenda Based Mobility Model," in *Proceeding of 39th ANnual Symposium on Simulation (ANSS 2006)*, pp. 188–195, 2006.

[41] F. Bai, N. Sadagopan, and A. Helmy, "Important: a frame work to systematically analyze the impact of mobility on performance o frouting protocols fo rad hoc networks," in *Proceeding of the 22th IEEE Annual Joint Conference on Computer Communications and Networking (INFOCOM '03)*, pp. 825–835, 2003.

[42] The Obstacle Mobility Model moment, http://moment.cs.ucsb.edu/mobility/.

[43] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, 2006.

[44] C. Diot, M. Martin, and N. Erik, Haggle project[EB/OL], 2004, http://www.haggleproject.org.

[45] P. Hui and J. Crowcroft, "How small labels create big improvements," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW '07)*, pp. 65–70, March 2007.

[46] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[47] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Tech. Rep., Duke University, Durham, NC, USA, 2000.

[48] P. Mundur, M. Seligman, and N. L. Jin, "Immunity-based epidemic routing in intermittent networks," in *Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '08)*, pp. 609–611, June 2008.

[49] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in *Proceedings of the 1st International MobiSys Workshop on Mobile Opportunistic Networking (MobiOpp '07)*, pp. 62–66, June 2007.

[50] X. Zhang, G. Neglia, and J. Kurose, "Performance modeling of epidemic routing," in *Proceeding of the International Federation for Information Processing Networking*, pp. 535–546, 2006.

[51] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN '05)*, pp. 252–259, August 2005.

[52] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the 1st International Workshop on Sensor Network Protocols and Applications*, pp. 30–41, IEEE Computer Society Press, Anchorage, Alaska, USA, 2003.

[53] Y. Jiang, Y. Li, L. Zhou, D. Jin, L. Su, and L. Zeng, "Optimal opportunistic forwarding with energy constraint for DTN," in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM '10)*, pp. 1–2, March 2010.

[54] Y. F. Lin, B. C. Li, and B. Liang, "Stochastic analysis of network coding in epidemic routing," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 794–808, 2008.

[55] F. Farahmand, I. Cerutti, A. N. Patel, Q. Zhang, and J. P. Jue, "Relay node placement in vehicular delay-tolerant networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, vol. 27, no. 1, pp. 2514–2518, November 2008.

[56] W. Zhao, Y. Chen, M. Ammar, M. D. Corner, B. N. Levine, and E. Zegura, "Capacity enhancement using throwboxes in DTNs," in *Proceedings of IEEE International Conference on Mobile Ad Hoc and Sensor Sysetems (MASS '06)*, pp. 31–40, October 2006.

[57] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in

Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '06), pp. 37–48, May 2006.

[58] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in *Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS '04)*, pp. 296–305, December 2004.

[59] Y. Gu, D. Bozdağ, and E. Ekici, "Mobile element based differentiated message delivery in wireless sensor networks," in *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM '06)*, pp. 83–92, 2006.

[60] Y. Gu, D. Bozdağ, E. Ekici, F. Özgüner, and C.-G. Lee, "Partitioning based mobile element scheduling in wireless sensor networks," in *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '05)*, pp. 386–395, September 2005.

[61] M. Shin, S. Hong, and I. Rhee, "DTN routing strategies using optimal search patterns," in *Proceedings of the 3rd ACM Workshop on Challenged Networks (CHANTS '08)*, pp. 27–32, September 2008.

[62] L. J. Chen, C. H. Yu, T. Sun, Y. C. Chen, and H. H. Chu, "A hybrid routing approach for opportunistic networks," in *Proceedings of the ACM SIGCOMM workshop on Challenged Networks (CHANTS '06)*, pp. 213–220, September 2006.

[63] Y. Liao, K. Tan, Z. Zhang, and L. Gao, "Combining erasure-coding and relay node evaluation in delay tolerant network routing," Microsoft Technical Report MR-TR, 2006.

[64] Y. Liao, K. Tan, Z. Zhang, and L. Gao, "Estimation based erasure-coding routing in delay tolerant networks," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '06)*, pp. 557–562, July 2006.

[65] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '04)*, vol. 34, pp. 145–158, ACM Press, October 2004.

[66] Y. Feng, M. Liu, X. Wang, and H. Gong, "Minimum expected delay-based routing protocol (MEDR) for delay tolerant mobile sensor networks," *Sensors*, vol. 10, no. 9, pp. 8348–8362, 2010.

[67] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical routing in delay-tolerant networks," in *Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN '05)*, pp. 237–243, August 2005.

[68] K. Tan, Q. Zhang, and W. Zhu, "Shortest path routing in partially connected ad hoc networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '03)*, pp. 1038–1042, December 2003.

[69] B. Burns, O. Brock, and B. N. Levine, "MV routing and capacity building in disruption tolerant networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 1, pp. 398–408, March 2005.

[70] X. Wang and C. Song, "Distributed real-time data traffic statistics assisted routing protocol for vehicular networks," in *Proceedings of the 16th IEEE International Conference on Parallel and Distributed Systems (ICPADS '10)*, pp. 863–867, December 2010.

[71] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems*, vol. 37, pp. 96–107, October 2002.

[72] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.

[73] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW '07)*, pp. 79–85, March 2007.

[74] Q. Yuan, I. Cardei, and J. Wu, "Predict and relay: an efficient routing in disruption-tolerant networks," in *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, pp. 95–104, May 2009.

[75] Y. Wang and H. Y. Wu, "Replication-based efficient data delivery scheme (RED) for delay/fault-tolerant mobile sensor network (DFT-MSN)," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW '06)*, pp. 485–489, March 2006.

[76] Y. Wang and H. Y. Wu, "Delay/fault-tolerant mobile sensor network (DFT-MSN): a new paradigm for pervasive information gathering," *IEEE Transactions on Mobile Computing*, vol. 6, no. 9, pp. 1021–1034, 2007.

[77] F. L. Xu, M. Liu, H. G. Gong, G. H. Chen, J. P. Li, and J. Q. Zhu, "Relative distance-aware data delivery scheme for delay tolerant mobile sensor networks," *Journal of Software*, vol. 21, no. 3, pp. 490–504, 2010.

[78] Y. Feng, H. Gong, M. Fan, M. Liu, and X. Wang, "A distance-aware replica adaptive data gathering protocol for delay tolerant mobile sensor networks," *Sensors*, vol. 11, no. 4, pp. 4104–4117, 2011.

[79] Y. Wang, H. Y. Wu, F. Lin, and N. F. Tzeng, "Cross-layer protocol design and optimization for delay/fault-tolerant mobile sensor networks (DFT-MSN's)," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 809–819, 2008.

[80] H. Gong, L. Yu, and F. Xu, "A data delivery protocol with periodic sleep for delay tolerant mobile sensor networks," *Journal of Convergence Information Technology*, vol. 7, no. 16, pp. 36–43, 2012.

[81] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile ad hoc networks," in *Proceeding of the 6th IEEE International Symposium on World of Wireless Mobile and Multimedia Networks (WoWMoM '05)*, pp. 183–189, June 2005.

[82] C. Mascolo, M. Musolesi, and B. Pásztor, "Opportunistic mobile sensor data collection with SCAR," in *Proceeding of the 4th International Conference on Embedded Networked Sensor Systems*, pp. 343–344, 2006.

[83] J. Leguay, T. Friedman, and V. Conan, "Evaluating MobySpace Based Routing Strategies in Delay Tolerant Networks," Université Pierre et Marie CURIE, Laboratoire LiP6-CNRS, Thales Communications.

[84] J. Leguay, T. Friedman, and V. Conan, "Evaluating mobility pattern space routing for DTNs," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, April 2006.

[85] G. Sandulescu and S. Nadjm-Tehrani, "Opportunistic DTN routing with window-aware adaptive replication," in *Proceedings of the 4th Asian Internet Engineering Conference (AINTEC '08)*, pp. 103–112, November 2008.

[86] M. Grossglauser and M. Vetterli, "Locating nodes with EASE: last encounter routing in ad hoc networks through mobility diffusion," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, pp. 1954–1964, April 2003.

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

[88] C. Lochert and H. Hartenstein, "A routing strategy for vehicular ad hoc networks in city environments," in *Proceeding of IEEE Intelligent Vehicles Symposium*, pp. 156–161, June 2003.

[89] C. Lochert and M. Mauve, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, pp. 69–72, 2005.

[90] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE rap: social-based forwarding in delay tolerant networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 241–250, May 2008.

[91] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pp. 32–40, September 2007.

[92] J. Ágila Bitsch Link, N. Viol, A. Goliath, and K. Wehrle, "SimBetAge: utilizing temporal changes in social networks for pocket switched networks," in *Proceedings of the 1st ACM Workshop on User-Provided Networking: Challenges and Opportunities*, pp. 13–18, December 2009.

[93] F. Li and J. Wu, "LocalCom: a community-based epidemic forwarding scheme in disruption-tolerant networks," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '09)*, pp. 1–6, 2009.

[94] K. Jahanbakhsh, G. C. Shoja, and V. King, "Social-greedy: a socially-based greedy routing algorithm for delay tolerant networks," in *Proceedings of the 2nd International Workshop on Mobile Opportunistic Networking (MobiOpp '10)*, pp. 159–162, February 2010.

[95] A. Mtibaa, M. May, C. Diot, and M. Ammar, "PeopleRank: social opportunistic forwarding," in *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 1–5, March 2010.

[96] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.

[97] C. Liu and J. Wu, "Routing in a cyclic mobispace," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 351–360, May 2008.

[98] J. Liu, H. Gong, and J. Zeng, "Preference location-based routing in delay tolerant networks," *International Journal of Digital Content Technology and Its Applications*, vol. 5, no. 12, pp. 468–474, 2011.

[99] J. Liu, M. Liu, and H. Gong, "Expected shortest path routing for social-oriented intermittently connected mobile network," *Journal of Convergence Information Technology*, vol. 7, no. 1, pp. 94–101, 2012.

[100] W. J. Hsu, D. Dutta, and A. Helmy, "CSI: a paradigm for behavior-oriented delivery services in mobile human networks," *Ad Hoc Networks*, vol. 6, no. 4, pp. 13–24, 2011.

[101] H. Gong, X. Wang, L. Yu, L. Wu, and C. Song, "Hot area based routing protocol for delay tolerant mobile sensor network," *Journal of Convergence Information Technology*, vol. 4, no. 15, pp. 450–457, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 398460, 10 pages
doi:10.1155/2012/398460

Research Article

Power Control in Distributed Wireless Sensor Networks Based on Noncooperative Game Theory

Juan Luo,[1] Chen Pan,[1] Renfa Li,[1] and Fei Ge[2]

[1] *School of Information Science and Engineering, Hunan University, Changsha 410082, China*
[2] *Central China Normal University, Wuhan 430079, China*

Correspondence should be addressed to Chen Pan, panchen@hnu.edu.cn

Received 29 September 2012; Revised 6 December 2012; Accepted 10 December 2012

Academic Editor: Nianbo Liu

A game theoretic method was proposed to adaptively maintain the energy efficiency in distributed wireless sensor networks. Based on a widely used transmission paradigm, the utility function was formulated under a proposed noncooperative framework and then the existence of Nash Equilibrium (NE) has been proved to guarantee system stability. To pursuit NE, an NPC algorithm was proposed to regulate heterogeneous nodes with various communication demands given the definition of urgency level. Results from both simulation and real testbed presented the robustness and rapid convergence of NPC algorithm. Furthermore, the network performance can remain in a promising state while the energy consumption is greatly decreased.

1. Introduction

Power control is one of the critical issues in wireless sensor networks (WSNs), especially when node is battery-powered. In wireless network, both throughput and bit error rate (BER) depend on the signal to interference and noise ratio (SINR) on receiver side, which will result in the transmission dilemma in wireless sensor networks. If transmitter raises its transmission power p_t to increase SINR, it will inevitably also act as noise to other nodes which are on the same channel. Therefore, power control in WSNs has been targeting to find certain appropriate strategies to alleviate the effect.

Most of solutions focus on regulating transmission power to increase the network capacity and prolong the battery life. To better manipulate transmitter power, Yates proposed an analytic method for power iteration, which is based on the satisfaction of signal to interference ratio (SIR) requirement [1]. A SIR balancing algorithm was developed by Zander that each and every terminal, by using this algorithm, would periodically adjust their power to converge to the corresponding SIR equilibrium [2].

As wireless sensor network has been evolving as the popular platform for large-scale applications, an alternative approach to the power control problem based on the game theory has been discussed. For example, in military and emergency scenarios, wireless sensor nodes under the same authority tend to work with each other in a fully cooperative way. Wu et al. proposed a fill-fledged cross-layer optimization design, which operated in a bandwidth-limited regime and in an energy-limited regime. The significant performance could be achieved by making a tradeoff between throughput and energy efficiency [3]. Wu and Bertsekas pointed out that generally power levels are assigned from a discrete set, and each mobile node holds its own interest so that the acceptable signal quality would individually not be the same. Eventually, the optimal solution could be found in a finite number of iterations [4]. To effectively communicate in energy-constrained network, Zhou et al. investigated the minimum energy relay selection mechanism jointly with transmission power control [5].

Recently, the applications of wireless sensor networks tend to focus on civilian usage that lacks of authority for any single node. In this situation nodes can not fully cooperate with each other, therefore noncooperative frameworks for solving the power control problem have been proposed. Long et al. featured the network that each individual held its own independent decision for the power selection. Based on the theory of stochastic fictitious play, a pure Nash

Equilibrium was realized with QoS requirement [6]. Altman et al. taken SINR as objective function and characterized both cooperative scenario and noncooperative one, while in noncooperative scenario, the system is modeled in a Hawk-Dove game form and each individual can choose either conciliation or conflict fighting for shared subcarriers [7]. Considered both SINR and network capacity, Sun et al. presented a distributed noncooperative game algorithm for the system to reach the proved unique NE [8]. Shi et al. consider the problem of power control for two independent relay-assisted wireless systems that that once both systems act noncooperatively to optimize their own rate, they can always reach a unique Nash equilibrium [9]. Tsiropoulou et al. studied the distributed power control problem via convex pricing of nodes' transmission power in the uplink of CDMA wireless networks and proved that their formulated MSUPC-CP game had a unique Pareto optimal Nash equilibrium. Finally a distributed iterative algorithm is proposed to compute the game's equilibrium [10]. Kesselheim analyzed the SINR capacity maximization problem, the proposed algorithm, under the SINR constraints, can maximize the number of simultaneous communications [11]. Lu et al. emphasized on noncooperative distributed power control in Gaussian interference channel and provide two types of power control schemes: gradient projection type and nonlinear type, both of which, however, were on the same propose of utility maximization. Convergence requirements were finally studied to supplement the utility function [12].

Those previous works, however, either did they not systematically analyze the convergence or did they not propose a reasonable solution to attain that convergence. For that matter, we have been fundamentally concerned about constructing a noncooperative game model for wireless sensor networks. Based on the importance levels of various messages, nodes in the game can define their own utility function individually. Because there is no central controller or infrastructure, the information of each node cannot be knowledgeable by others. Thereby, even if Nash Equilibrium (NE) exists, it may not be achieved directly through the Best Response (BR) choice. In this case a convergence algorithm is proposed so that nodes can be guided and quickly converge to the NE point with a stable network performance.

The remainder of the paper is organized as follows. Section 2 builds up system model and defines the utility function. Convergence analysis and detailed description of NPC algorithm are given in Section 3. Experiments from both the simulation and real testbed are evaluated in Section 4. Finally, Section 5 concludes the paper and discusses future work.

2. System Model and Utility Function

2.1. Preliminaries. In wireless communication, Energy is consumed by both receiver and transmitter denoted as p_{rx} and p_{tx}, respectively. During communication, the transmission power p_t also plays as noise to other nodes that share the same channel. Hence, the utility function can be defined as $U_i(p_i, P_{-i})$, where p_i denotes power usage on link i which is comprised of both p_{tx} and p_{rx}, and $P_{-i} = \sum_{j \neq i} p_{jt} h_{ji}$ denotes the interference power from other transmission links, where p_{jt} is the link j's transmission power at time t and h_{ji} represents the link gain from link j's transmitter to link i's receiver. To ensure power convergence of this noncooperative system, the existence of NE should first be guaranteed. The condition for the existence of NE points in noncooperative game was proposed in [13]. Let $G = [N, \{P_i\}, \{U_i(^*)\}]$ denote power control game for every $i \in N$. If the power strategy tuple $P^* = (p_1^*, p_2^*, p_3^* \cdots p_n^*)$ is NE of game G, the following condition should be satisfied for every $i \in N$ and $p_i', p_i^* \in P_i$:

$$U_i(p_i^*, P_{-i}^*) \geq U_i(p_i', P_{-i}^*). \tag{1}$$

Nash Equilibrium is a fixed point of best response power strategy profile that everyone chooses its BR power based on the choices of others which builds up internal connection for everyone.

Lemma 1. *An NE point exists in the game G if*

(1) *Power strategy set P_i is a nonempty, convex, and compact subset of some Euclidean space R_n;*

(2) *$U_i(^*)$ is continuous in p and quasiconcave in p_i.*

In order to be compatible with the NE requirements, power strategy should first be quantified. The smallest unit is defined as $(p_{i\max} - p_{i\min})/E_i$ where E_i denotes the degree of quantification which should not be infinite.

p_i and P_{-i} are variables of Utility U_i. In order to achieve the maximum U_i, its partial derivative with respect to p_i should first be made, then let $\partial U_i/\partial p_i = 0$ to calculate the extreme value strategies $\{p_E\}$. After that we bring those extreme values into the second-order partial derivative of U_i with respect to p_i and ground on the requirements of $\partial^2 U_i/\partial^2 p_i|_{p_i = p_E} < 0$ to decide which one of them can make the max value for U_i. And, if the max value does exist, we denote this extreme value as p_{EM}. Because of possible irrational players and unexpected stimulus, power stability of the game model G should be taken into consideration.

Theorem 2. *Define F as the interference power, then game G will be power stabilized, if and only if $\partial p_{iEM}/\partial F \leq 0$ and $-1 < \sum_{j \neq i}(\partial p_{jEM}/\partial F) * h_{ji} \leq 0$, for every $i \in N$.*

Proof. Consider a scenario that each transmitter has already reached to its balance point, and because of some unknown stimulus, there is an increment of ρ to the background noise of every node. If the system is power stabilized, there must be another balance point for each and every transmitter to assign its power strategy. Now let us define p_{jEMk} as link j's extreme power value at time k, and each transmitter refreshes its p_{iEM} synchronously based on its interference power F that is composed by P_{-i} and background noise Δ. Because of this, we take it as a Markov Process with memory only to the situation one step before.

Let $K(F)$ denote $\partial p_{iEM}/\partial F$, then we can initiate p_{iEM}, p_{-i} as

$$p_{iEM0} = \int_0^{P_{-i0}+\Delta} K(x)dx + p_{iEM}\,|_{p_{-i},\,\Delta=0},$$
$$p_{-i0} = \sum_{j\neq i} p_{jEM0}h_{ji}. \tag{2}$$

After ρ was added into the system, during the first iteration we have

$$p_{iEM1} = \int_0^{P_{-i0}+\Delta+\rho} K(x)dx + p_{iEM}\,|_{p_{-i},\,\Delta=0},$$
$$p_{-i1} = \sum_{j\neq i} p_{jEM1}h_{ji}. \tag{3}$$

Firstly, $K(F) \leq 0$ should be the prerequisite of the following deduction. According to the Mean Value Theorem for Integrals, p_{iEM1} can be expressed as

$$p_{iEM1} = p_{iEM0} + \rho K(A_i), \tag{4}$$

where $A_i \in (p_{-i0} + \Delta, p_{-i0} + \Delta + \rho)$. If $K(A_i) > 0$, p_{iEM1} will be larger than p_{iEM0} after first iteration. For the following iterations, because each link does the same process, $p_{iEMk}|_{k\geq 2}$ would continue boosting up to its new balance level as shown in Figure 1. However, this new level may be beyond $p_{i\max}$; therefore this power strategy will not be adopted for the energy saving matter. For the situation when $K(F) \leq 0$, p_{iEM} is inversely proportional to the interference F. We need to make sure that the iterations sequence are convergent other than divergent, therefore, two more cases need to be considered. One is when $p_{-i0} - \rho < p_{-i1} < p_{-i0}$ as shown in Figure 2, which means $F_0 < F_1 < F_0 + \rho$. Because of this, we have $p_{iEM1} < p_{iEM2} < p_{iEM0}$. For the rest of iteration, as a Markov Process, $p_{iEMk}|_{k>2}$ should be somewhere between p_{iEM0} and p_{iEM1}. As for another case when $p_{-i1} < p_{-i0} - \rho$ as shown in Figure 3, which means $F_1 < F_0$. In order to be convergent, $p_{-i2} < p_{-i0}$ should be guaranteed based on the Markov Process. $p_{iEMk}|_{k>2}$ would gradually be stabilized at the following iterations, otherwise the iteration sequence is divergent. Also because we have $p_{-i2} = \sum_{j\neq i} p_{jEM2}h_{ji}$, the condition $p_{iEM2} < p_{iEM0}$ should first be satisfied, and then we have following deduction.

For

$$p_{-i1} = \sum_{j\neq i} p_{jEM1}h_{ji} = p_{-i0} + \rho \sum_{j\neq i} K\left(A_j\right)h_{ji}, \tag{5}$$

we have

$$p_{iEM2} = \int_0^{P_{-i1}+\Delta+\rho} K(x)dx + p_{iEM}\,|_{p_{-i},\,\Delta=0}$$
$$= p_{iEM0} + \rho\left(1 + \sum_{j\neq i} K\left(A_j\right)h_{ji}\right)K(B_i). \tag{6}$$

Because $P_{iEM2} < P_{iEM0}$, we derive $\rho(1 + \sum_{j\neq i} K(A_j)h_{ji})$ $K(B_i) < 0$ in which

$$B_i \in \left(p_{-i0} + \Delta, p_{-i0} + \Delta + \rho\left(1 + \sum_{j\neq i} K\left(A_j\right)h_{ji}\right)\right). \tag{7}$$

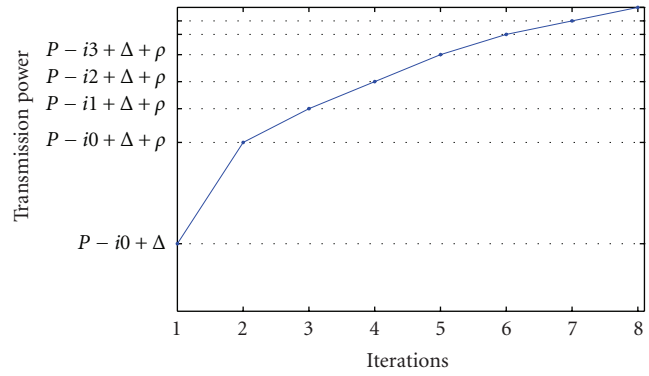


Figure 1: The escalation of interference.

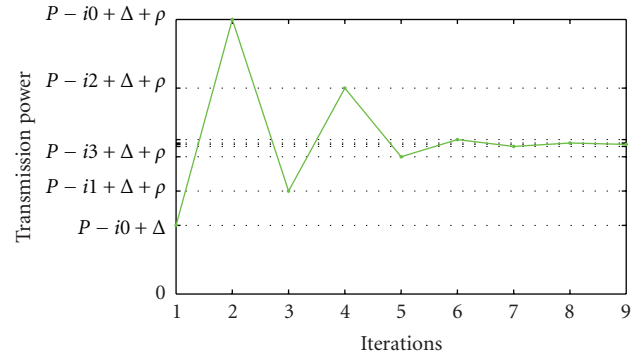


Figure 2: System converged with low magnitude of fluctuation.

Thus for $K(B_i) \leq 0$, we have

$$1 + \sum_{j\neq i} K\left(A_j\right)h_{ji} > 0 \Longrightarrow \sum_{j\neq i} K\left(A_j\right)h_{ji} > -1. \tag{8}$$

This completes the proof of theorem. □

Based on the Theorem 2 one can verify whether the game model will be power stabilized.

2.2. System Model. In order to formulate a noncooperative framework, the utility function should first be made. According to the Shannon Theorem, channel capacity can be expressed as $C = W\log_2(1+S/N)$, which reasonably indicates the benefits that a link can achieve during communication. However, Shannon Capacity is more easily calculated than realized. In addition to different types of modulations, the maxima capacities approaching to the Shannon Capacity are numerically quite distinct. We choose the capacity model presented in [14] so that the achievable rate can be well approximated by $F(\gamma) = W\log_2(1+\gamma/\Gamma)$, where Γ represents the gap to the Shannon capacity and γ denotes SINR.

During transmission, energy consumed by transmitter and receiver makes up the link consumption p_i. According to the data revealed in [15, 16], transmitting and receiving have so comparable power consumption that it is acceptable to assume $p_{tx} = p_{rx}$ approximately. p_{tx} can simply be expressed as $p_{tx} = \alpha p_t + \chi$, where α and χ are constants depending on

process topology and architecture. Thereafter we have $p_i = \eta p_t + \omega$, where η, ω are constants depending on hardware property.

2.3. Utility Function. Power consumption is taken as the cost for the game model, and the utility function is proposed as follows:

$$U_i(p_i, P_{-i}) = W \log_2\left(1 + \frac{\gamma(p_i, P_{-i})}{\Gamma}\right) - a_i p_i \gamma \geq \gamma_{\min}, \quad (9)$$

where a_i is urgency index which represents the urgency level of the transferred information, $\gamma(p_i, P_{-i})$ is the SINR on link i's receiver, and $\gamma_{\min}$ is the minimum SINR that transmission can tolerate. In our system, we choose the path-loss model used in [17] with free space propagation model and Omnidirection antenna, the path-loss function can be described as $p_r = p_t K[d_0/d]^{\beta} = p_t h_i$, where d_0 is the reference distance for the antenna, β denotes the path-loss exponent determined by the environments, and h_i is path gain from link i's transmitter to receiver. When $d > d_0$, K can empirically approximate as $KdB = 20\log_{10}(\lambda/4\pi d_0)$, where λ is the signal wavelength.

Supposing that the background noise is AWGN with the same thermal noise power σ for every receiver, (9) could be rewritten as

$$U_i(p_{it}, P_{-i}) = W\log_2\left(1 + \frac{p_{it}h_i}{(P_{-i} + \sigma)\Gamma}\right) - a_i(\eta p_{it} + \omega) \quad \gamma \geq \gamma_{\min}. \quad (10)$$

In order to find out whether (10) is qualified for non-cooperative power control game; the existence of NE should firstly be proved [13]. Since we already quantified the power strategy set $\{P_i\}$ and the given conditions shows that $\{P_i\}$ is nonempty, convex, and compact set of some Euclidean space R_n, the first condition of Lemma 1 is satisfied. Additionally from (10) we can see that U_i is continuous in P, so it remains to show the quasiconcaveness of U_i in p_i. Firstly let us define the quasiconcaveness.

Definition 3. A function $f : A \to \Re$ defined on a convex subset A of a real vector space is quasiconcave if it satisfies (11) where every $x, y \in A$ and $\mu \in [0, 1]$:

$$f[\mu x + (1 - \mu)y] \geq \min[f(x), f(y)]. \quad (11)$$

According to [13] we can prove the quasiconcaveness of U_i by demonstrating that the local maximum of U_i is, at the same time, the global maximum.

The local maximum can be calculated from $\partial U_i/\partial p_i = 0$. Here we only modify p_i to be p_{it}, thus

$$\frac{\partial U_i}{\partial p_{it}} = \frac{Wh_i}{[(P_{-i} + \sigma)\Gamma + p_{it}h_i]\ln 2} - a_i\eta = 0 \implies p_{iE} = \frac{W}{a_i\eta \ln 2} - \frac{\Gamma(P_{-i} + \sigma)}{h_i}, \quad (12)$$

where p_{iE} is the best response (BR) power strategy that is either global maximum or minimum. However,

$$\frac{\partial^2 U_i}{\partial p_{it}^2} = \frac{-W{h_i}^2}{[(P_{-i} + \sigma)\Gamma + p_i h_i]^2 \ln 2} < 0. \quad (13)$$

Therefore p_{iE} is global maximum, and Lemma 1 is satisfied. We conclude that NE exists in this noncooperative power control game G.

Theorem 4. *NE in game G is unique.*

Proof. Since $P_{-i} = \sum_{j \neq i} p_{jt}h_{ji}$, where h_{ji} denotes the path gain form link j's transmitter to link i's receiver, the power vector $p = \langle p_{1E}, p_{2E}, p_{3E}, \ldots, p_{nE}\rangle^T$ can be written in matrix notation that

$$p' = QMp + S. \quad (14)$$

In which p' is the BR power strategies of p of the next time step, Q equals to $-\Gamma$, M is a $N \times N$ matrix of path-gain expressed as follows:

$$M = \begin{pmatrix} 0 & \frac{h_{21}}{h_1} & \frac{h_{31}}{h_1} & \cdots & \frac{h_{n1}}{h_1} \\ \frac{h_{12}}{h_2} & 0 & \frac{h_{32}}{h_2} & \cdots & \frac{h_{n2}}{h_2} \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ \frac{h_{1n}}{h_n} & \frac{h_{2n}}{h_n} & \frac{h_{3n}}{h_n} & \cdots & 0 \end{pmatrix}. \quad (15)$$

The above matrix makes up a seamless connection for each node within the network. And

$$S = (S_1, S_2, \ldots, S_n)^T, \quad \text{where } S_i = \frac{W}{a_i\eta \ln 2} - \frac{\sigma\Gamma}{h_i}. \quad (16)$$

Since the existence of NE has been proved in game G, we have $p = QMp + S$ at NE points. Thus, we derive (17) where E is an identity matrix of size N:

$$p = (E - QM)^{-1}S. \quad (17)$$

Since E, Q, M, and S are constants, the solution of (17) is unique, and NE is unique in game G.

Based on the two conditions in Theorem 2, the first condition $\partial p_{iEM}/\partial F = -\Gamma/h_i < 0$ is satisfied and the second condition can be expressed as $0 < \sum_{j \neq i}(h_{ji}/h_j) < 1/\Gamma$, which, depending on the real situations, cannot be proved directly. However, it can act as a constraint for node density scale. If it is not satisfied, the transmitter will not finally be stabilized. Our experiment results in latter section show that if the link satisfies the condition of $\gamma > \gamma_{\min}$, system will eventually be stabilized. □

3. Convergence Analysis and NPC Algorithm

3.1. Urgency Index. Game model formulated in (10) can be applied individually. Generally the tradeoff making during the transmission is not merely about energy consumption and network capacity, but the urgency of information should also be taken into consideration. Here we define a parameter a_i to indicate the urgency level of data for different individuals.

According to (12) P_{-i} ranges from maximum $\{P_{-i}\}$ to 0, thus, we have

$$\begin{aligned} &\frac{W}{a_i\eta\ln 2} - \frac{\Gamma(\max\{P_{-i}\}+\sigma)}{h_i} \le p_{iEM} \\ &\le \frac{W}{a_i\eta\ln 2} - \frac{\Gamma\sigma}{h_i} \quad \gamma \ge \gamma_{\min}, \end{aligned} \tag{18}$$

where $\max\{P_{-i}\}$ is the maximum interference that link i can tolerate. It can be expressed as $\max\{P_{-i}\} = (p_{i\max}h_i/\gamma_{\min}) - \sigma$.

Since $0 \le p_{iEM} \le p_{i\max}$, we have (19)

$$\begin{aligned} 0 \le &\frac{W}{a_i\eta\ln 2} - \frac{\Gamma(\max\{P_{-i}\}+\sigma)}{h_i} \le p_{iEM} \\ &\le \frac{W}{a_i\eta\ln 2} - \frac{\Gamma\sigma}{h_i} \le p_{i\max}. \end{aligned} \tag{19}$$

For $\gamma \ge \gamma_{\min}$ because

$$\frac{p_{iEM}h}{(P_{-i}+\sigma)} \ge \gamma_{\min}. \tag{20}$$

We also have (21)

$$p_{iEM} \ge \gamma_{\min}\frac{(P_{-i}+\sigma)}{h_i}. \tag{21}$$

Thus, from (12), (19), and (21) we get inequality set as follows:

$$\begin{aligned} &0 \le \frac{W}{a_i\eta\ln 2} - \frac{\Gamma(\max\{P_{-i}\}+\sigma)}{h_i}, \\ &\frac{W}{a_i\eta\ln 2} - \frac{\Gamma(P_{-i}+\sigma)}{h_i} \ge \gamma_{\min}\frac{(P_{-i}+\sigma)}{h_i}, \\ &\frac{W}{a_i\eta\ln 2} - \frac{\Gamma\sigma}{h_i} \le p_{i\max}. \end{aligned} \tag{22}$$

And then, we have (23)

$$\begin{aligned} &\frac{Wh_i}{(p_{i\max}h_i+\Gamma\sigma)\eta\ln 2} \le a_i \\ &\le \operatorname{Min}\left\{\frac{W\gamma_{\min}}{\Gamma p_{i\max}\eta\ln 2}, \frac{Wh_i}{(\gamma_{\min}+\Gamma)(P_{-i}+\sigma)\eta\ln 2}\right\}. \end{aligned} \tag{23}$$

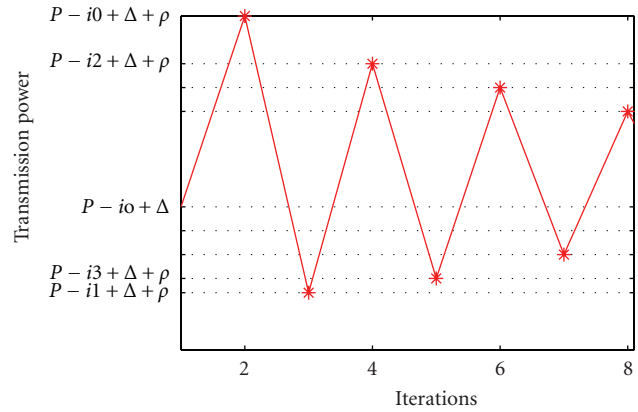


Figure 3: System converged with high magnitude of fluctuation.

From (19) we notice that in any circumstances (24) is satisfied, thus (25) is always satisfied:

$$\begin{aligned} &0 \le \frac{W}{a_i\eta\ln 2} - \frac{\Gamma(\max\{P_{-i}\}+\sigma)}{h_i}, \\ &\frac{W}{a_i\eta\ln 2} - \frac{\Gamma\sigma}{h_i} \le p_{i\max}, \end{aligned} \tag{24}$$

$$\frac{Wh_i}{(p_{i\max}h_i+\Gamma\sigma)\eta\ln 2} \le a_i \le \frac{W\gamma_{\min}}{\Gamma p_{i\max}\eta\ln 2}. \tag{25}$$

Therefore in (23) the communication can be well established if

$$(p_{i\max}h_i+\Gamma\sigma) > (\gamma_{\min}+\Gamma)(P_{-i}+\sigma). \tag{26}$$

Otherwise the communication will shut down.

So players, based on their needs, could adjust the value of a_i under the restriction of (23).

3.2. NPC Algorithm. Assuming that each transmitter knows its SINR before making their self-fulfilling choice based on the feedback of the receiver. Let p_n denote power strategy on step n and p_n^* denote BR power strategy of step n based on (12). Because we took it as Markov process, the power strategy selection is given in (27) to avoid highly fluctuation of the system during iterations, where ψ is a preset parameter that guarantees the smoothness of variation during power updating. In order to be eligible, ψ should be at least in the same order of magnitude of $P_{\max}$:

$$p_{n+1} = p_n\frac{|p_n - p_n^*| + n + \psi}{p_{\max}+n+\psi} + p_n^*\frac{p_{\max} - |p_n - p_n^*|}{p_{\max}+n+\psi}, \quad n \ge 0. \tag{27}$$

Figure 4 presents the whole process of NPC algorithm in details.

Each time step transmitter update its power strategy based on (27). C_{TH} is a threshold that is defined individually. Transmission power is stable when Counter = C_{TH} is

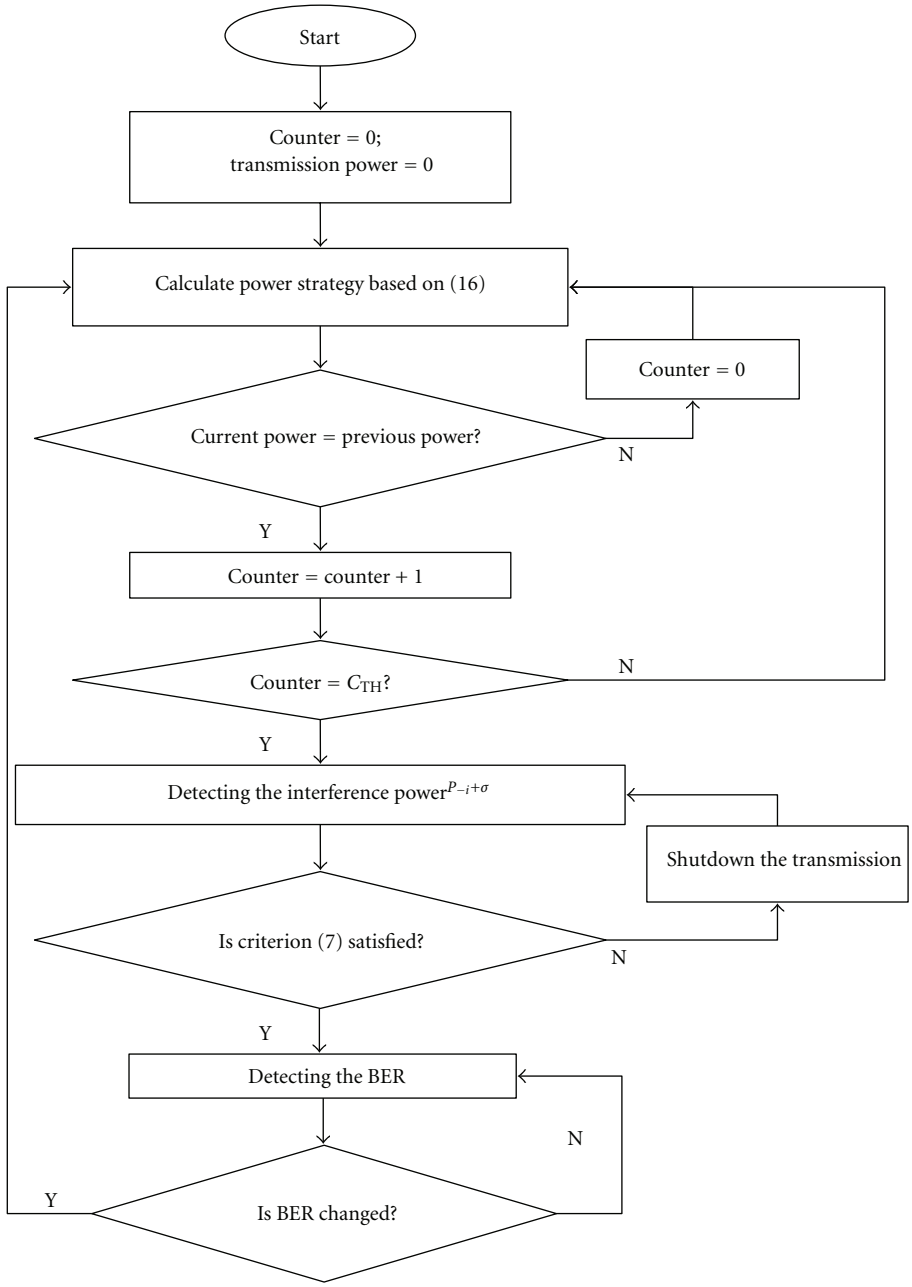


Figure 4: The process of NPC algorithm.

satisfied. n is the time step that has the same order of quantity as the smallest unit of transmission power. When power is stable, node will check if the criterion (26) is satisfied. If not, node will turn off transmission immediately, because in this case it is too much costly for node to communicate with high BER. After that node will wait until the above criterion is satisfied and starts the transmission again.

4. Experimental Results

4.1. Theoretic Simulation. We simulated a network on the MATLB platform in a 500 × 500 square meters area with 100 nodes randomly placed on it. Two nodes can communicate normally if their SINR level is higher than $\gamma_{\min}$. Without loss of generality, the utility function is parameterized conservatively as follows to model the typical environment outside.

The bandwidth $W = 2 \times 10^7$ Hz, the AWGN power $\sigma = 10^{-10}$ w, SINR boundary $\gamma_{\min} = 50$, and $\Gamma = 1.5$. For $p_r = p_t K[d_0/d]^{\beta}$, K, d_0, and β were given by $K = 10^{-4}$, $d_0 = 10$ m, and $\beta = 4$, respectively. Transmission power p_t was quantitatively ranged from 0 mw to 200 mw, and urgency index a_i was given with the boundary of (23). Finally the parameter ψ and η were defined as 400 and 5.

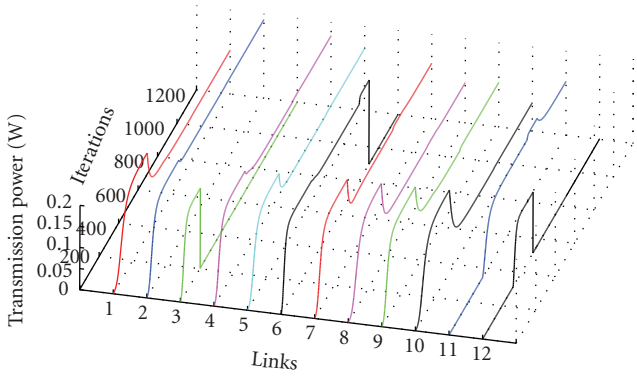


Figure 5: Transmission power of each link during iterations.

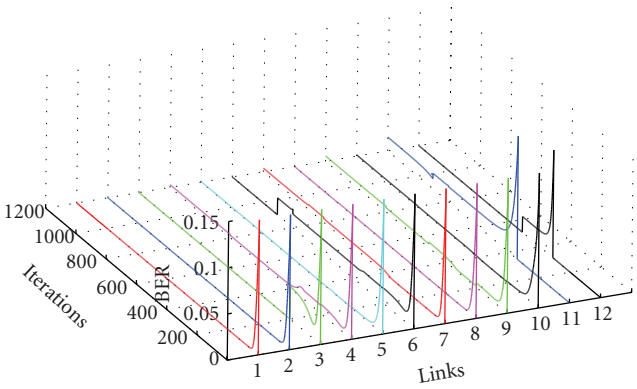


Figure 6: BER of each link during iterations.

We randomly chose N links in the above scenario, and each link with it own utility function working successfully if SINR $\geq \gamma_{\min}$. The results were evaluated by stability of the system as well as network performance.

We chose 12 different links. On the first stage 10 links started working at the same time under the highest data urgency level. After that each link defined its own unique urgency index asynchronously while link 11 and 12 joined the network. Finally the background noise was raised to examine robustness of the system.

Figure 5 shows that on the first stage (0–300), for approximately 200 iterations, nodes could be power stabilized with smooth changes. However, for link 3 it was so unsustainable to its current BER that the communication will be shut down. The second stage (300–800) reveals that no matter new links access joined or the existing links changed their data urgency level, the system can still reach to the balance, and it is the same truth that those cannot afford the high BER will do the same as link 3 did. Finally on the last stage (800–1200), we raised the background noise and the result shows that the system was sustainable for the sudden changes. The respective BER for each node during the whole process is shown in Figure 6. When system is stable, the BER of each node which remains in the system is in a relatively low level.

The links showed in Tables 1 and 2 are typical to represent the quality status of all the experimental links. From Table 1 we can see that link 1′s quality was good because its BER was constantly at a low level, which means link 1 was rarely affected by the neighbors. However, for link 6, it was always with a high transmission power. Finally when the power stabilized BER was unaffordably high as marked with asterisk token, the transmitter of link 6 chose to shutdown transmission immediately. For link 5 it did not perform as good as link 1 did, because its noise was much higher. Link 11 accessed into the network after 300 iterations, despite the fact that the (26) is satisfied, its BER was constantly high. The reason for this is that link 11 was under a relatively low urgency level, so it was unnecessary for link 11 to raise its transmission power.

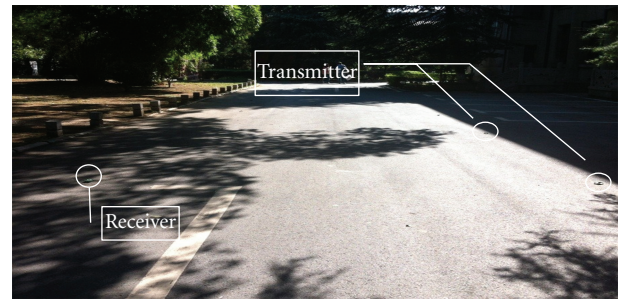


Figure 7: Network topology of the experiment.

Table 1: Power strategies of certain links during the iterations.

Iterations	Link			
	1	5	6	11
100	0.1251	0.1607	0.1916	0.0000
300	0.1976	0.1985	0.1945	0.0000
400	0.1196	0.1475	0.1834	0.1034
600	0.1051	0.1462	0.1825	0.1463
800	0.1035	0.1460	0.1824	0.1469
900	0.1041	0.1477	0.2000*	0.1574
1200	0.1041	0.1471	—	0.1276

Table 2: BER of certain links during the iteration.

Iterations	Link			
	1	5	6	11
10	$8.77e-2$	$8.92e-2$	$8.91e-2$	—
30	$1.18e-2$	$2.51e-2$	$2.32e-2$	—
100	$9.83e-7$	$1.51e-3$	$1.42e-2$	—
300	$8.95e-9$	$4.59e-4$	$1.64e-2$	—
350	$1.22e-7$	$8.21e-4$	$1.51e-2$	$1.13e-1$
400	$1.24e-5$	$5.11e-4$	$1.44e-2$	$2.73e-2$
600	$9.74e-8$	$4.59e-4$	$1.45e-2$	$1.62e-2$
800	$1.19e-8$	$4.47e-4$	$1.45e-2$	$1.61e-2$
900	$5.98e-7$	$7.57e-4$	$1.83e-2^*$	$1.66e-2$
1200	$5.54e-7$	$6.21e-4$	—	$1.48e-2$

4.2. Real Testbed. To analyze the feasibility of our approaches, a *SOC* solution, CC2530, tailored for *IEEE*

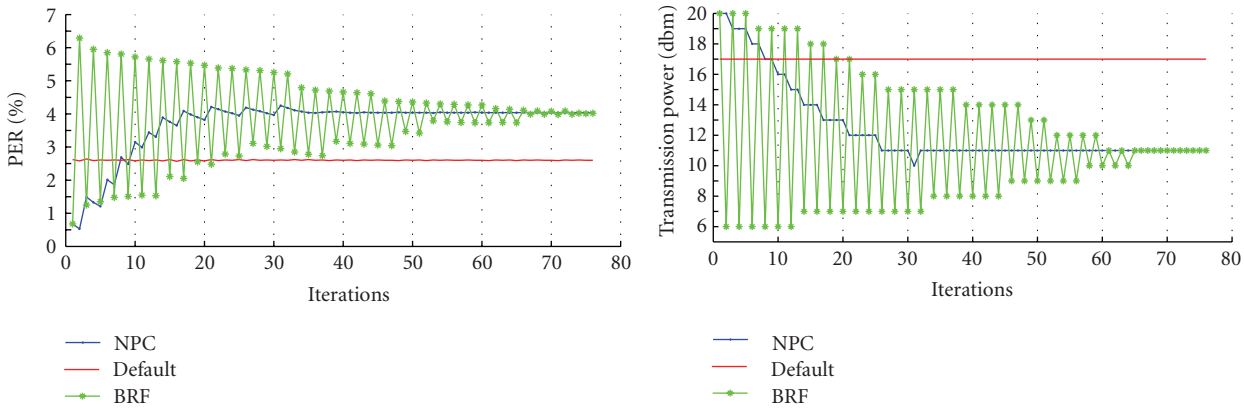


Figure 8: PER and Transmission Power of the best performed node.

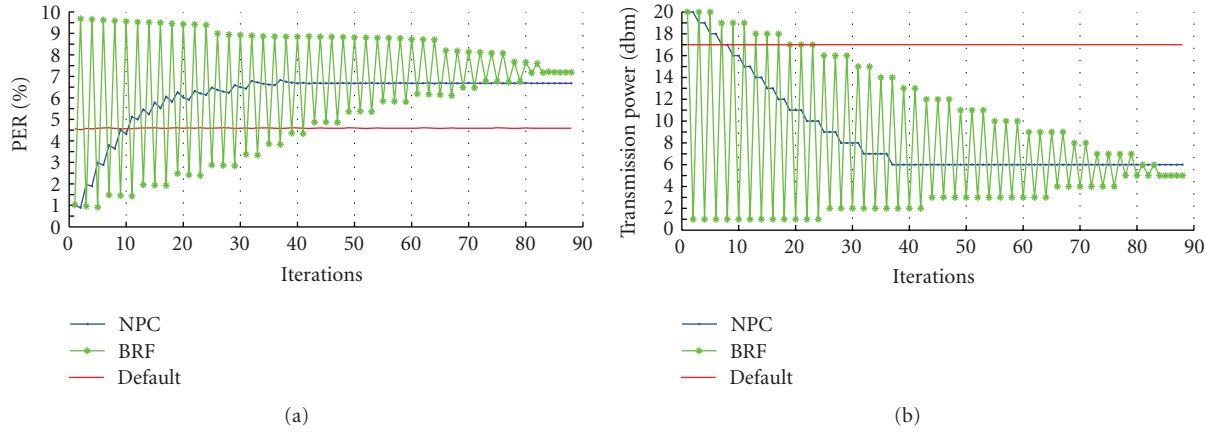


Figure 9: PER and Transmission Power of the worst performed node.

802.15.4/*Zigbee* applications, was used in our implementation for real testbed. The transmission power cannot be quantized into as many as the theoretic analysis did, but by connecting with the RF front-end, *CC2591*, there were 20 available levels of transmission power to operate.

The network was made up of 5 pairs of nodes with each pair consisting of one transmitter and receiver. Nodes were placed in a mesh form as shown in Figure 7. The distance from transmitter to receiver of each pair was 5 m. And any adjacent pair was 6 m away from each other. Each node was working in the same channel, and message was delivered with only one hop.

The receivers at each end got the best quality of signal for the reception, while the receivers at middle got the signal with most interference. The transmitter can adjust its transmission power according to the feedback of its SINR, which was detected by its own pair of receiver.

The parameters of the node and algorithm were defined as follows.

The maximum transmission power was 20 dbm and the minimum interval of the transmission power was 1 dbm, the bandwidth was 5×10^6 Hz, and we defined ψ as 40 dbm. Since path-loss exponent cannot be detected directly, we used SINR as an alternative way to calculate the extreme value of transmission power in (12). The rest parameters were same as that in simulations.

To evaluate performance of NPC, 5 transmitters were initialized with the maximum transmission power which was 20 dbm. Two other counterpart methods were applied as comparison. The Default is a default setting of transmission power recommended by Manufacture Company. BRF is a strategy selection method directly based on the best response function (12) that is applied by [8].

We compared the performance of two pairs of links which had the highest interference and lowest interference, respectively.

Figure 8 shows that in less than 50 iterations the transmission power was stabilized by NPC algorithm, and the transmission power of the least interfered link pair ended up to be 11 dbm with packet error rate (PER) of 4%. While on default setting, the power level was 17 dbm, and placed on the same position, the PER was 2.7%. BRF ended up with the

same power as NPC did. However on default setting the PER was not much lower than NPC but the power consumption was exaggeratedly higher. While on BRF, the fluctuation of the transmission power would greatly jeopardize the stability of the network. Our network scale was not big enough; otherwise the transmission power would risk to be evolved divergently.

Figure 9 shows the comparison of PER and Transmission power of each methods on link pair with the heaviest interference. The whole process is the same like Figure 8 illustrated. As we can see, because of the high interference, the transmitter had a great cutback in its power, and compared with the default setting, PER was only less than 2%, but the amount of power diminished was 17 dbm–6 dbm, which on a great extent, conserved the energy for the node. To the BRF, the Transmission power ended up to be 1 level lower than on the NPC algorithm, the result may due to the variation of the environment.

To analyze the energy efficiency of NPC algorithm, we evaluated the performance of each pair under the three different methods with PER/Consumption as measurement of energy efficiency and stabilization time as convergence speed.

During the experiment, consumption was measured as total consumption of link pair. Figure 10 shows that NPC and BRF were with almost the same energy efficiency which was much better than default setting, however according to Figure 11 BRF iterated with twice more time than NPC and as interference increased convergence speed will become smaller. Thus, energy efficiency has been remarkably enhanced by NPC algorithm, while the convergence speed was a little less than default setting and was much faster than BRF offered. Therefore, the superiority of the NPC algorithm is that without accessing into the profile of others, the power selection strategy can optimize the energy usage of each node with guaranteed smooth changes without which the sudden burst of interference is inevitable.

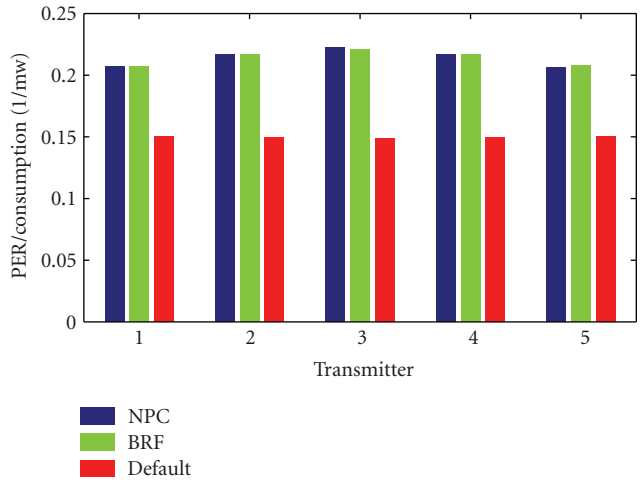


Figure 10: Energy Efficiency of 5 nodes under different methods.

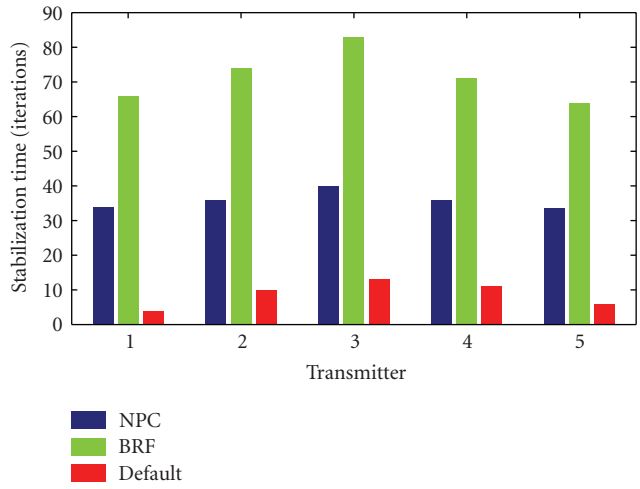


Figure 11: Convergence Speed of 5 nodes under different methods.

5. Conclusions

In wireless sensor network, nodes prefer to form and organize the system in a distributed way. Without central node, it is quite recommendable to take Noncooperative behavior into consideration. The conditions of formulating a NPC game were firstly proved in the paper. After that we presented a utility function based on the dilemma of power usage and proved the existence of unique NE in this specific game model. The notion of urgency index was given for node to define its utility with the consideration of data urgency level. For system convergence, NPC algorithm was presented and compared with other methods. The experiments showed that, without inquiring the profile of each node, NPC algorithm can quickly lead the system to stabilize with relatively smooth changes as well as good network performance. For those qualities, NPC algorithm can be applied to the scenarios where nodes do not share information together while the system design requires energy efficiency. Currently the design, however, mainly concentrates on the physical layer for transmitter to choose its transmission power based on the SINR that makes the network performance not quite good at the beginning. For this reason the next step of our research will focus on the cross-layer power control design with noncooperative game approaches.

Acknowledgments

This work is partially supported by Program for New Century Excellent Talents in University; Hunan young core teacher project; Young teacher project from Hunan University.

References

[1] R. D. Yates, "Framework for uplink power control in cellular radio systems," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1341–1347, 1995.

[2] J. Zander, "Performance of optimum transmitter power control in cellular radio systems," *IEEE Transactions on Vehicular Technology*, vol. 41, no. 1, pp. 57–62, 1992.

[3] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S. Y. Kung, "Network planning in wireless ad hoc networks: a cross-layer approach," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 136–149, 2005.

[4] C. Wu and D. P. Bertsekas, "Distributed power control algorithms for wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 50, no. 2, pp. 504–514, 2001.

[5] Z. Zhou, S. Zhou, J. H. Cui, and S. Cui, "Energy-efficient cooperative communication based on power control and selective single-relay in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 8, pp. 3066–3079, 2008.

[6] C. Long, Q. Zhang, B. Li, H. J. Yang, and X. Guan, "Non-cooperative power control for wireless ad hoc networks with repeated games," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 6, pp. 1101–1112, 2007.

[7] E. Altman, K. Avrachenkov, and A. Garnaev, "Transmission power control game with SINR as objective function," in *Network Control and Optimization*, vol. 5435 of *Lecture Notes in Computer Science*, pp. 112–120, 2009.

[8] Q. Sun, X. Zeng, N. Chen, Z. Ke, and R. U. Rasool, "A non-cooperative power control algorithm for wireless ad hoc & sensor networks," in *Proceedings of the 2nd International Conference on Genetic and Evolutionary Computing (WGEC '08)*, pp. 181–184, Jingzhou, China, September 2008.

[9] Y. Shi, R. K. Mallik, and K. B. Letaief, "Power control for relay-assisted wireless systems with general relaying," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, Beijing, China, May 2010.

[10] E. Tsiropoulou, G. Katsinis, and S. Papavassiliou, "Distributed uplink power control in multi-service wireless networks via a game theoretic approach with convex pricing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 61–68, 2012.

[11] T. Kesselheim, "A Constant-factor approximation for wireless capacity maximization with power control in the SINR model," in *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '11)*, pp. 1549–1559, San Francisco, Calif, USA, January 2011.

[12] Q. Lu, T. Peng, C. Hu et al., "Convergence of utility-based power control in Gaussian interference channel," *The Institution of Engineering and Technology Communication*, vol. 5, no. 8, pp. 1052–1059, 2011.

[13] C. U. Saraydar, N. B. Mandayam, and D. J. Goodman, "Efficient power control via pricing in wireless data networks," *IEEE Transactions on Communications*, vol. 50, no. 2, pp. 291–303, 2002.

[14] S. Toumpis and A. J. Goldsmith, "Capacity regions for wireless ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 2, no. 4, pp. 736–748, 2003.

[15] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, John Wiley & Sons, London, UK, 2005.

[16] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.

[17] A. Goldsmith, *Wireless Communications*, Cambridge University Press, Cambridge, UK, 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 280515, 12 pages
doi:10.1155/2012/280515

Research Article

IPARK: Location-Aware-Based Intelligent Parking Guidance over Infrastructureless VANETs

Hui Zhao, Li Lu, Chao Song, and Yue Wu

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Hui Zhao, jenniferzhao09@gmail.com

Received 16 October 2012; Accepted 13 November 2012

Academic Editor: Haigang Gong

It has been pointed out that about 30% of the traffic congestion is caused by vehicles cruising around their destination and looking for a place to park. Therefore, addressing the problems associated with parking in crowded urban areas is of great significance. One effective solution is providing guidance for the vehicles to be parked according to the occupancy status of each parking lot. However, the existing parking guidance schemes mainly rely on deploying sensors or RSUs in the parking lot, which would incur substantial capital overhead. To reduce the aforementioned cost, we propose IPARK, which taps into the unused resources (e.g., wireless device, rechargeable battery, and storage capability) offered by parked vehicles to perform parking guidance. In IPARK, the cluster formed by parked vehicles generates the parking lot map automatically, monitors the occupancy status of each parking space in real time, and provides assistance for vehicles searching for parking spaces. We propose an efficient architecture for IPARK and investigate the challenging issues in realizing parking guidance over this architecture. Finally, we investigate IPARK through realistic experiments and simulation. The numerical results obtained verify that our scheme achieves effective parking guidance in VANETs.

1. Introduction

Finding an available parking space in a crowded urban area or a large parking lot is always time-consuming, which would incur great economic losses, for example, searching for free parking spaces causes total of 20 million Euros economic damage per year in the district Schwabing of Munich, including 3.5 million Euros for gasoline and diesel and 150000 hours of waiting time [1]. Moreover, a recent study [2] discovers that about 30% of the traffic congestion is caused by vehicles cruising around their destination and looking for a place to park. Therefore, it is of great significance to address the problems associated with parking in crowded urban areas. Obviously, if drivers are provided with efficient parking guidance services, the parking spaces searching costs would be greatly reduced and the traffic congestion would be also relieved to some extent.

In the last decade, many research efforts have been devoted to designing efficient parking guidance systems [3–7]. These works could be generally classified into two categories: sensors based scheme and RSUs based scheme. In the sensors based scheme [3–5], fixed sensors are deployed in the parking lot to monitor parking spaces, which provide real-time parking space availability information for the drivers looking for parking spaces. One of the biggest weaknesses of this approach is that it necessitates a large installation cost and operational cost in order to adequately monitor the parking spaces at a city-wide level. According to a Department of Transportation report [8], the installation cost of typical per spot parking management systems ranges from \$250–\$800 per spot. Moreover, the deployed sensors might become invalid over a period of time owing to the power exhaustion. In the RSUs based scheme [6, 7], it employs the parking lot RSUs to monitor the whole parking lot and perform parking guidance. This scheme is proved to be efficient and effective. However, deploying RSUs at a large scale also requires a large amount of investment and elaborate design.

In view of the drawbacks of the existing schemes, we put forward an intelligent parking guidance scheme IPARK, which harnesses the free resource offered by parked vehicles to perform parking guidance in urban areas. In IPARK,

the cluster formed by parked vehicles generates the parking lot map automatically, monitors the occupancy status of each parking space in real time and provides services for vehicles searching for parking spaces. Overall, IPARK could provide the drivers with two kinds of services: parking availability information dissemination outside the parking lot and real-time parking navigation inside the parking lot. Specifically, our IPARK scheme is substantiated with a distributed architecture. Over this architecture, the cluster formed by parked vehicles establishes the parking occupancy map according to the reported location of each parked vehicle, and provide real-time navigation for vehicles entering this parking lot. Moreover, different parking clusters in a specific area share their parking data periodically and jointly response to the parking data query request. We try to tackle the challenging issues in performing efficient parking guidance over the proposed architecture, for example, how to manage the parked vehicles to facilitate parking guidance and how to generate the internal map of a parking lot. Finally, we investigate our scheme through realistic experiments and simulation. The results prove that our scheme achieves high performance in parking guidance.

The original contributions that we have made in the paper are highlighted as follows.

(i) To the best of our knowledge, we are the first to consider the using of parked vehicles in parking guidance. Our scheme aims at fully exploiting the benefits of parked vehicles, without requiring any infrastructure investment.

(ii) We propose a distributed architecture for parking guidance over infrastructureless VANETs and tackle the challenging issues in perform parking guidance over this architecture.

(iii) We evaluate the proposed scheme through analyzing realistic parking data and performing simulation in NS-2.33. The numerical results obtained verify that our scheme achieves effective parking guidance in VANETs.

The remainder of this paper is structured as follows. Section 2 makes a brief overview of related work. Section 3 describes the framework of IPARK. In Section 4, we explain the design of IPARK step by step. Section 5 evaluates IPARK through realistic experiments and simulation. Finally, Section 6 summarizes the paper.

2. Related Work

In recent years, many research efforts have been devoted to developing effective parking guidance schemes for urban areas.

Some researchers propose to exploit the wireless sensor networks to achieve parking guidance. In [3], an optical wireless sensor network is deployed in the parking lot, which monitors the parking spaces and informs drivers of the number of available parking spaces and in which area should they be directed to. Similar scenario has also been proposed in [4], which detects the presence of parked vehicles over parking spots using fixed sensors and provides the parking statistics for vehicles searching for parking spaces. Furthermore, the city of San Francisco is presently installing a stationary sensor network to cover 6000 parking spaces under the SFPark project [5]. This network detects the presence of a vehicle using a magnetometer and delivers the data to a centralized parking monitoring system, which then carries out parking guidance based on the collected data. The deficiencies of the sensors based schemes is as follows. First, deploying sensors in a large parking lot can be very expensive. Second, the sensors can become inaccurate and would stop functioning easily when time passes.

Alternatively, approaches based on RSUs to accomplish efficient parking management have also been considered. SPARK [6] employs RSUs to surveil and manage the parking lot. It provides the drivers with real-time parking navigation service, intelligent antitheft protection, and friendly parking information dissemination. In [7], Panayappan et al. also provide a RSUs-based approach for parking space availability monitoring. The parking lots are managed by RSUs, and these RSUs can provide open parking space information to the drivers, which is very similar as the proposed SPARK scheme. However, deploying RSUs at a large scale also requires a large amount of investment and elaborate design.

In addition, Caliskan et al. [9] propose a topology independent scalable information dissemination algorithm for free parking places discovery. With the friendly parking lot information disseminated by the parking automats and inter-vehicle broadcast, the drivers can conveniently find their preferred free parking lot. However, this scheme relies on the roadside parking fees payment terminal, which has not gained popularization until now. ParkNet [10] collects parking space occupancy information by moving vehicles. Each ParkNet vehicle is equipped with a GPS receiver and a passenger-side-facing ultrasonic rangefinder to determine parking spot occupancy. The data is aggregated at a central server, which builds a real-time map of parking availability and could provide this information to clients that query the system in search of parking. The drawback of ParkNet is that periodically reporting the parking status to a central server over cellular uplink would incur high communication cost. In addition, it only focuses on the roadside parking, which has limited practicality.

Finally, in [11, 12], the authors believe that the huge amounts of vehicles on parking lots are an abundant and underutilized computational resource that can be tapped into for the purpose of providing third-party or community services This means that more and more attention will be paid to the parked vehicles, which inspires our motivation to utilize parked vehicles to perform intelligent parking guidance.

3. System Model

3.1. Assumptions and Design Goals. First, we assume that vehicles are equipped with GPS and preloaded electric maps, which are already popular in new cars and will be common in the future. Second, we assume that some vehicle users will share their devices during parking. This could be motivated

by effective incentives. In view of the powerful resources possessed by parked vehicles, the businessmen may be willing to provide all sorts of incentives to make it attractive for the owner of parked vehicles to share the resources in their parked vehicles [12]. Finally, we assume that vehicles parked within the same parking lot form one cluster, which is feasible owing to the high occupancy of urban parking lot.

According to a real world urban parking report [13], street parking, off-street parking and interior parking (garages or underground parking lots) account for 69.2%, 27.1%, and 3.7% of total, respectively. Here, we mainly focus on the outside parking lots, including roadside parking lots and off-street parking lots. Our IPARK aims at exploiting the cluster formed by vehicles parked within the same parking lot to surveil the whole parking lot and provide effective guidance for vehicles searching for parking spaces. Specifically, this intelligent parking scheme is designed to achieve the following two goals: real-time parking navigation inside the parking lot and efficient parking information dissemination outside the parking lot.

3.1.1. Real-Time Parking Navigation. While a vehicle enters a large parking lot, the parking lot cluster should send the current parking occupancy map (a map showing each parking spot as occupied or vacant) to it and help it find an available parking space.

3.1.2. Efficient Parking Information Dissemination. While a vehicle user is driving on a road and issues a parking data query request, a parking lot nearby should respond to this request and return the query results (the number of parking spaces available on each parking lot nearby) at a small delay.

3.2. Architecture. To achieve the aforementioned two design goals, the architecture of IPARK is elaborately designed, which is as shown in Figure 1. It consists of three components: parking lot clusters, intermediate nodes and end users.

Parking lot clusters are composed of vehicles parked within the same parking lot. They work both as parking monitoring units and data dissemination units, which gather the information necessary for parking guidance and provide services for the vehicles in search of parking spaces.

Intermediate nodes are the vehicles helping to connect different parking lot clusters. They could be either vehicles moving on the road or individual parked vehicles in a private parking lot.

End users are the vehicle users who are seeking for an unoccupied parking space while driving on a road or entering a large parking lot.

Over this architecture, each parking lot cluster monitors the corresponding parking lot and establishes a parking occupancy map in real time, based on which it provides navigation services for the incoming vehicles. Moreover, the parking lot clusters within a specified area share their real time parking space availability information periodically, which jointly respond to the parking data query request.

This proposed architecture has the following two distinct advantages. First, it fully exploits the benefits of parked vehicles, and does not require any infrastructure investment. Thus, it is cost saving. Second, with the parking availability information distributed to many parking lot clusters, a driver seeking for an available parking lot could get a response within short time. In addition, with the parking space occupancy map in hand, a driver entering a large parking lot could find a vacant parking space swiftly. Thus, it is time saving. The above two merits make IPARK attractive to both parking managers and vehicle users. Thus, it could easily obtain supports from all participators.

4. The Design of IPARK

At a high level, the design goal of IPARK is to exploit the parking lot cluster to realize efficient parking guidance over VANETs. To achieve this goal, the following four issues need to be addressed.

(1) How to manage the parked vehicles within a parking lot to facilitate parking guidance?

(2) How to obtain the internal map of a parking lot?

(3) How to achieve effective parking navigation inside the parking lot?

(4) How to realize efficient parking availability information dissemination outside the parking lot?

In this section, we attempt to investigate feasible solutions for the above issues.

4.1. Parking Lot Cluster. In IPRAK, we perform parking guidance on the basis of parking lot cluster. Vehicles parked within the same parking lot are considered to be one parking lot cluster. A typical parking lot cluster is as shown in Figure 2, which has one cluster head H and some cluster members as M_1–M_{23}. It is established as follows. At the beginning, there is not any parked vehicle. Then, among the earliest incoming vehicles, the one which provides PVA (Parked Vehicle Assistance) service becomes the cluster head. After the cluster head is determined, other parked vehicles join the cluster and become the cluster members. We notice that the parking lot cluster will malfunction at once if the cluster head became invalid (e.g., the wireless device is turned off suddenly or the cluster head is left). Thus, we introduce a quasi-head, as QH in Figure 2, to ensure fault-tolerance with respect to exception handling. A quasi-head is the cluster member next to a cluster head, which always keeps a copy of recent cluster status from the cluster head. It becomes a special cluster member, working as a "warm backup" for the management of the parking lot cluster. While the cluster head is invalid, the quasi-head becomes the new cluster head and a vehicle close to the cluster head is appointed as the new quasi-head. Once the cluster head is determined, it manages all cluster members, act as local service access points, and perform parking guidance. Whenever a parking lot becomes empty, the parking lot cluster needs to be re-established and the parking data needs to be gathered again. However, it has been proved that the possibility for a parking lot in urban areas being completely empty is very slim [14, 15]. Once a parking lot

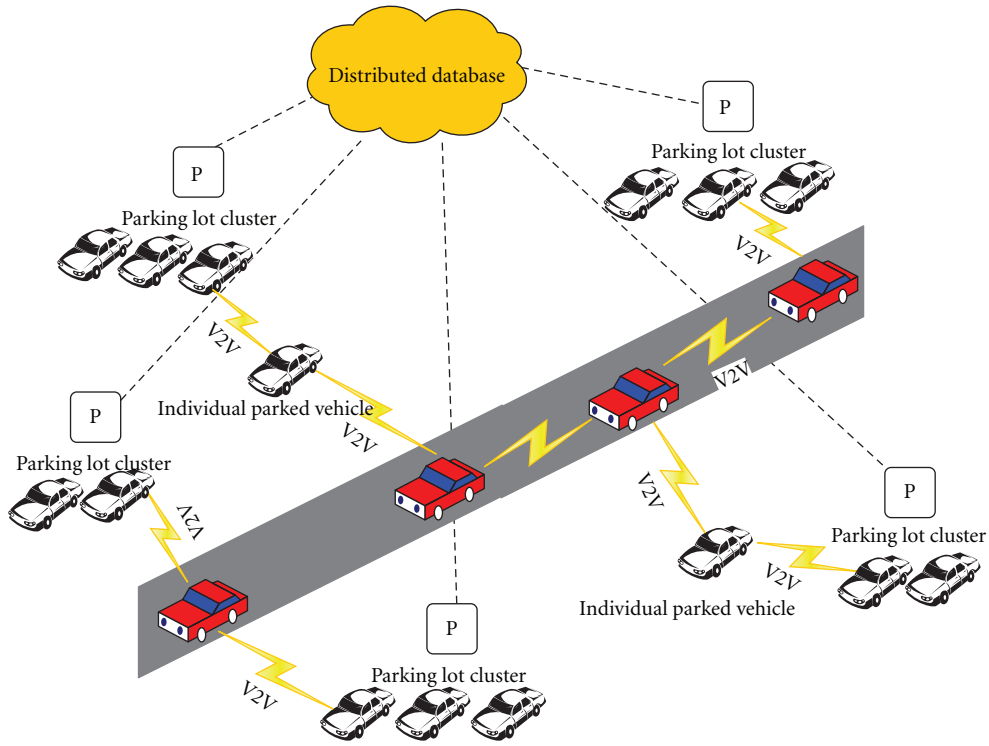


Figure 1: System architecture.

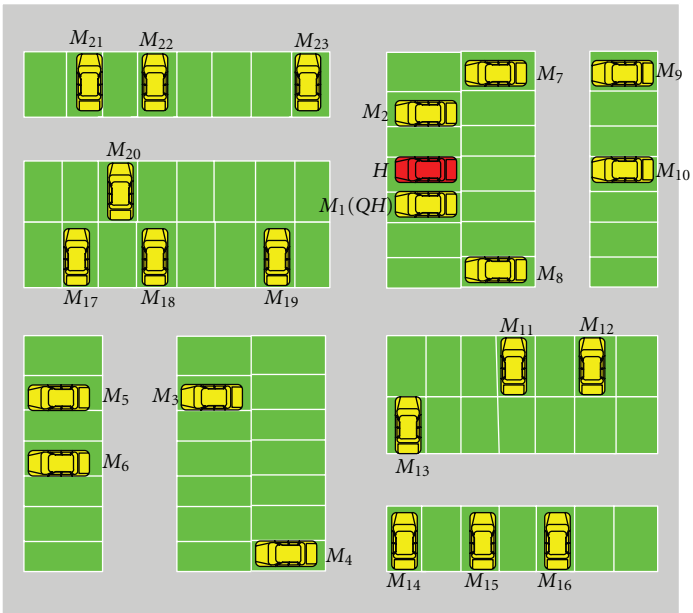


Figure 2: A typical parking lot cluster.

cluster is established, it could work steadily for a long period of time.

4.2. Location-Aware-Based Map Generation. To realize effective parking guidance, the parking lot cluster needs to obtain an internal map of the parking lot beforehand. This map should include the total number of parking spaces and their specific locations. The total number of parking spaces indicates the capacity of the parking lot, while the specific location could be either a physical location or a logical location. As there is no way for a vehicle getting such a map until now, we offer an automatic parking lot map generation scheme.

To establish a parking lot map, one intuitive way is letting the parked vehicles report the GPS readings at the corresponding parking spots to the cluster head. However,

the existing GPS technology is not accurate enough to locate each parking space. Even the GPS reader corrected by WAAS has an error of 3 m (standard deviation), not to mention the ones in the area without WAAS. Thus, instead of utilizing the physical location, we record the logical location of each parked vehicle. A vehicle estimates its logical location according to the road topology of the parking lot and the recorded GPS trajectory. Before turning off the engine, it reports the estimated logical location to the cluster head, based on which the cluster head detects the existence of a parking space. We will explain how to get the accurate logical location of a vehicle.

As demonstrated in Figure 2, a typical parking lot is composed of many parking spaces and several roadways. After entering a parking lot, the vehicle users have to drive along the roadways and park in one of the vacant spaces. While analyzing the GPS trajectories of different vehicles, we observe the following two characteristics. First, when a vehicle is turning from one roadway to another, the trajectory direction displays a significant difference, as shown in Figure 3. Second, the GPS error is highly correlated for a long driving distance, which is reflected by the fact that the vehicle trajectory is nearly paralleled with the real roads in Figure 3. It is worth to note that, we are not the first ones to make such observations, similar characteristics have already been discovered and utilized by many works [10, 16]. We assume that the turnings in the GPS trajectories could be matched to the road bends in the parking lot using the existing map matching techniques [16, 17]. Then, according to the first observation, the turning of a roadway and the related GPS reading could be easily recognized from the vehicle trajectory. Thus, we could take the roadway turnings as the reference points of the parking lot, based on which the relative locations of the parked vehicles are estimated. According to the second observation, while a vehicle turns from one roadway to another and drives on the new roadway, the GPS error could be deemed as identical within a certain driving distance. That is to say, if we calculate a distance on the basis of the GPS readings at the turning and a road point within a certain distance, it could be used to represent the relative distance from that road point to the turning (referenced point). By this means, the logical location of the parked vehicle could be obtained easily. The logical location estimation algorithm is depicted in Algorithm 1.

The parking lot cluster generates the parking lot map through the aggregation of reported logical location of different parked vehicles. Theoretically, once each parking space is occupied one time, the parking lot map could be generated. However, the logical location reported by the parked vehicle might not accurate occasionally owing to some unpredicted errors. Therefore, we let the parking lot cluster remove the ones with obvious derivation and correct the parking lot map through long periods of time. An example of a parking lot map is as shown in Figure 4. Each parking space is represented by ⟨*Referencepoint,Logicalnumber*⟩, which indicates the logical location relative to the chosen reference point. Although this location representation scheme is very simple, it could uniquely identify a parking space. After each parking space is occupied over a period of time, the capacity of the

Figure 3: GPS trajectory sample.

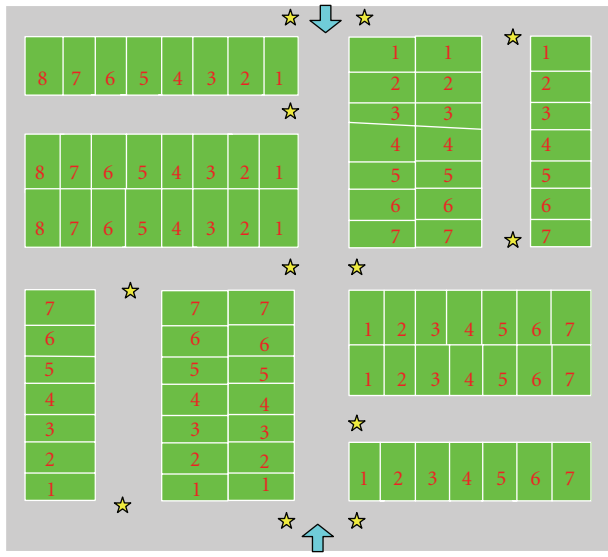


Figure 4: An example of logical parking lot map.

parking lot could also be estimated. Assume that the parking lot is divided into n blocks by the roadway, then, the capacity of the parking lot could be calculated as follows:

$$C = \sum_{i=1}^{n} C_i. \tag{1}$$

Additionally, as the GPS error is only correlated within a certain driving distance, the threshold within which the GPS error correlation could be utilized to compute the logical location of a parked vehicle is of great significance. In our simulation, we investigate this threshold through large number of experiments under urban scenario. The results show that within a driving distance of 100 m, 90% of the error between the estimated distance and the real distance is less than 1.5 m, which is small enough to uniquely determine a parking space. Thus, we take 100 m as the threshold. Out of this threshold, the reference point is redefined. While a vehicle drives as long as 100 m along a road segment, the average GPS reader at that point is selected as a new reference point, based on which the relative distance is estimated.

Require: GPS trajectory data *GPS_DATA* of vehicle v_j, road topology *PARK_ROADS* of the parking lot, a threshold Φ
Ensure: The logical location of the parked vehicle v_j, represented by (R_j, d_j).
1: Preprocess the GPS trajectory data and remove the GPS readers with large derivation as $GPS\_DATA = preprocess(GPS\_DATA)$;
2: **for** $i = 1; i <= size(GPS\_DATA) - 2; i++$ **do**
3: $A = GPS\_DATA[i], B = GPS\_DATA[i+1], C = GPS\_DATA[i+2]$;
4: Find the angle between $\overrightarrow{AB}$ and $\overrightarrow{BC}$ as $\theta = angle(\overrightarrow{AB}, \overrightarrow{BC})$;
5: **if** $\theta \geq \Phi$ **then**
6: Put B into the turning point linked list as $S.add(B)$;
7: **end if**
8: **end for**
9: Derive the trajectory path *PATH* of vehicle v_j as an ordered set $PATH = (S[n-1]S[n]) \mid 1 \leq n \leq size(S)$
10: Match the derived path *PATH* with the road segments within the parking lot and find the last road segment the vehicle drive on as $R_j = match(PATH, PARK\_ROADS)$;
11: Compute the distance between the turning point of R_j and the road point corresponding to the parking space of vehicle v_j as $d_j = distance(turningPoint(R_j), roadPoint(v_j))$;
12 :**return** (R_j, d_j)

Algorithm 1: Logical location estimation within a parking lot.

4.3. Parking Navigation. In SPARK [6], the parking navigation is realized through recommending an available parking space to the incoming vehicle and then guiding it to that space step by step. However, this scheme is impractical for the following two reasons. First, it neglected the demand of vehicle users. For the vehicle user, compared with being recommended with a vacant parking spot, they prefer to get the occupancy status of the parking lot, and then choose one available parking space according to personal preference and driving proficiency. Second, considering the fact that the vehicle driver might not adopt the recommended parking space, it is unreasonable to guide the vehicle users to a recommended parking space step by step.

In our scheme, we let the parking lot cluster establish a parking occupancy map and send this map to the incoming vehicles. To achieve this goal, a vehicle is required to report the estimated logical location to the cluster head before turning off the engine. For the cluster head, it detects the occupancy status of each parking space and maintains a record containing three fields as follows for it:

PID: the ID of this parking space.

OCC: the occupancy status of this parking space. 0 represents unoccupied, and 1 represents occupied.

VID: if this parking space is occupied, VID represents the ID of the parked vehicle.

With the internal map of the parking lot and the occupancy status of each parking space, it is not difficult to derive the corresponding parking occupancy map. One part of a parking occupancy map is as shown in Figure 5.

While a vehicle entering a large parking lot, it would periodically broadcast a beacon message. After receiving this beacon message, the cluster head sends the parking occupancy map to this vehicle, and the vehicle user chooses to stop in one of the vacant parking spot.

In order to reduce the transmission overhead, the cluster head could determine whether to provide parking navigation service according to the current occupancy ratio of the parking lot. While the parking occupancy is less than a certain threshold, it is easy for a vehicle user to find a vacant parking spot, and there is no need for the cluster head to send a parking occupancy map to it. While the parking occupancy is higher than a certain threshold, the cluster head begins to perform parking navigation.

4.4. Parking Availability Data Dissemination. To help a vehicle moving on a road find an available parking space conveniently, we organize the parking lot clusters within a specified area into a distributed parking database, and provide near real-time parking data query services for the vehicle users. According to the strategy used, this process could be divided into two phases: parking data sharing among different

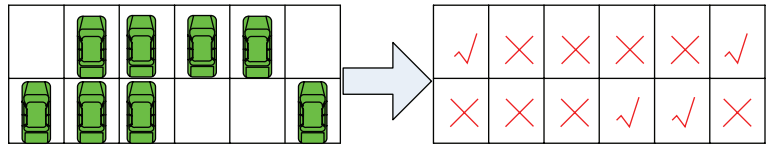


Figure 5: An example of parking lot occupancy map.

parking lot clusters and parking data dissemination. We will describe them in detail in the following part.

4.4.1. Parking Data Sharing among Different Parking Lot Clusters. We assume that one parking lot cluster knows the location of other parking lot clusters within a certain range. This could be realized through a simple mechanism with the help of intermediate vehicles, for example, each parking lot cluster periodically broadcasts its location (the location of cluster head) to the parking clusters within two hop (the TTL is set as 2), and adjacent parking lot clusters exchange the information (similar like ⟨*clusterID*,*location*⟩) they obtain with each other. This process is similar to Link-State Broadcast [18]. Due to the high occupancy of urban parking lots, a long broadcast cycle is enough.

Under this assumption, we abstract the parking lot clusters within a specified area as a weighted connected graph $G(V, E)$, where V is the set of parking lot clusters, E is the set of communication links between adjacent parking lot clusters (which might be intermittently connected) and weight D_{ij} on E is the transmission delay of the link between adjacent parking lot clusters. Figure 6 shows one such weighted connected graph. We let adjacent parking lot clusters periodically send a delay probe packet to each other and estimate the transmission delay according to the history record. As the transmission delay between two parking lot clusters is affected by their mutual distance, the traffic density, and other factors that changes slowly, this approximation is reasonable.

After this weighted connected graph G is acquired, each parking lot cluster periodically broadcast the parking availability information to other parking lot clusters over the minimum spanning tree MST of graph G. The minimum spanning tree MST could be easily acquired at each parking cluster through the classic Kruskal's algorithm or Prim's algorithm, both of which are of polynomial complexity:

$$\text{MST} = \arg\min_{T \subset G} \sum_{i,j \in T} D_{ij}. \tag{2}$$

With this parking data sharing scheme, each packet only needs to be replicated while new tree branch appears, which greatly decreases the transmission overhead. Moreover, routing along the minimum spanning tree could guarantee that total estimated data sharing delay is minimized. Although routing along any one spanning tree could make sure that the packet could be received by each parking cluster, routing along the minimum spanning tree could realize the same goal in a shorter time.

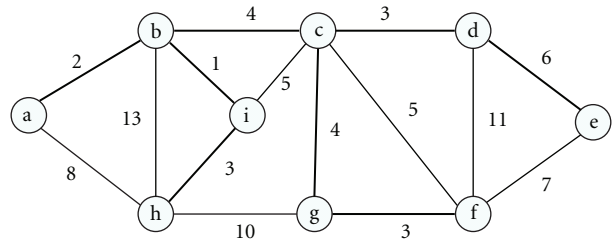


Figure 6: A minimum spanning tree.

4.4.2. Parking Data Dissemination. After the distributed parking database is established, it is responsible to deal with the parking data query request and disseminate the parking availability data to the end user.

In regards to the end user, there is no determined destination for query dissemination. However, some query request may occur between vehicles and parking lot cluster within mutual radio range. Since the vehicles can monitor nearby beacons, query messages can be directly sent to the parking lot cluster and the query results are returned immediately. In regards to the vehicles outside the radio range, we need a proper strategy to spread queries over VANETs. Epidemic routing [19] ensures eventual message delivery via random pair-wise exchanges of messages between mobile hosts, which has decent performance, but often causes heavy resource consumption in the network. GPSR [20] provides a location-based scheme that all packets transmitted into the network are marked by the originator with their destination's location. In greedy forwarding, intermediate nodes simply choose to forward the packet to the neighbor located closest to the destination, which is efficient and robust in straight roads, but often causes extra transmission and delay in complex topologies. In order to respond to a parking space query quickly and efficiently, we develop an enhanced GPSR to disseminate a query in a neighboring area around a vehicle.

When a driver issues a parking space query request, it is disseminated as follows. In the straight mode (as shown in Figure 7(a)), this query message is forwarded to the furthest neighbor along the driving direction. If the intermediate forwarder encounters a parking lot cluster, it sends the query message to the parking lot cluster, and the parking lot cluster returns the query results along a revered path as the query message using GPSR. If no parking lot cluster encounters, the intermediate forwarder sends the query request to the next intermediate node until a parking lot cluster is encountered. In the intersection mode (as shown in Figure 7(b)), the

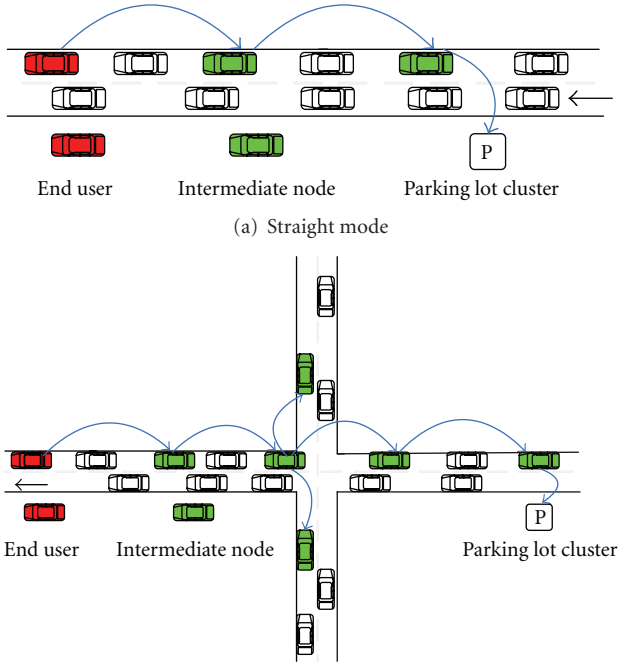


Figure 7: Query dissemination in IPARK.

query message is replicated so that there is one copy at each possible driving direction. While receiving a query request, the parking lot cluster returns the query results along a reversed path as the query message.

By this means, we could assure that the vehicle user could get a reply with small delay regardless of which path it would take at the crossing. Considering the fact that some other vehicles driving within the same area might be also seeking for parking spaces, we further improve our scheme as follows: while the query message is delivered using Improved GPSR, if the intermediate forwarder encounters a vehicle which has already received the response message recently, the results will be returned and the response message will be dropped.

5. Performance Evaluation

In order to evaluate IPARK accurately, we collect realistic parking data in a large parking lot and perform a series of numerical experiments on these data. Furthermore, we also examine the performance of IPARK in NS-2.33.

5.1. Numerical Experiments and Results. We collect realistic parking data at a large parking lot of Chengdu, a city in China. As shown in Figure 8, this parking lot has about 2000 parking spots. Due to the fact that this parking lot has a high occupancy most of the time, we only choose a relatively vacant subarea, as marked in Figure 8, for study. Our private car is quipped with a Garmin 18–5 Hz GPS, which has 12 channel receiver and provides 5 fresh GPS readings per second. During the experiment, we repeatedly drive into the parking lot. To simulate the real parking behavior of

Figure 8: Parking lot under study.

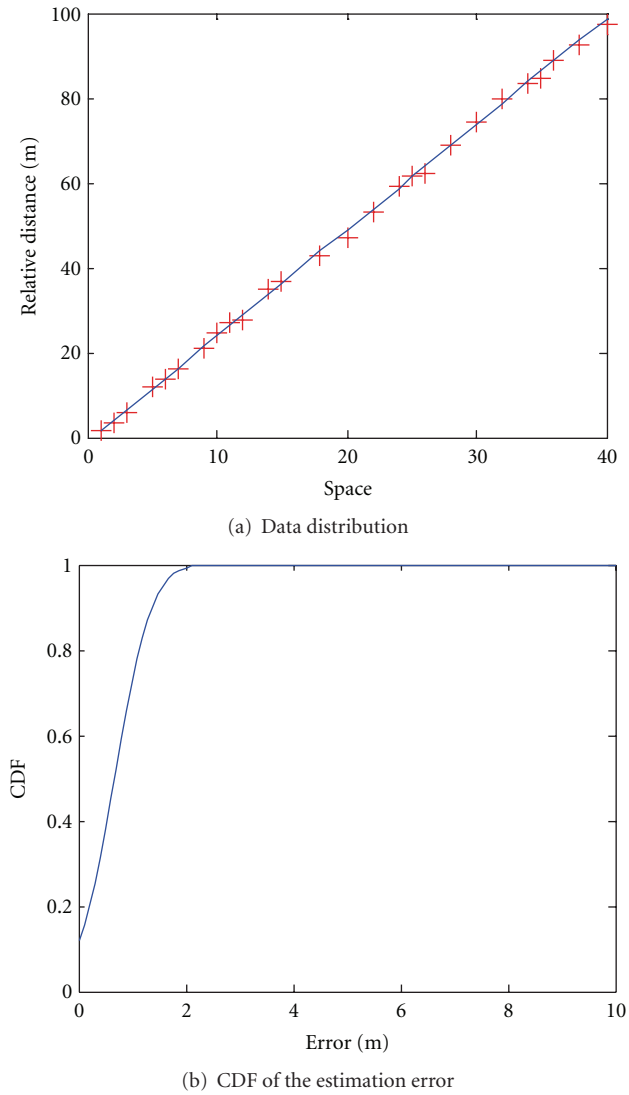
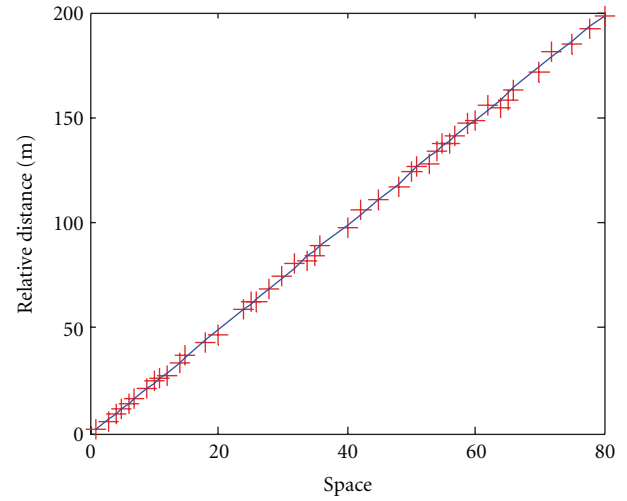
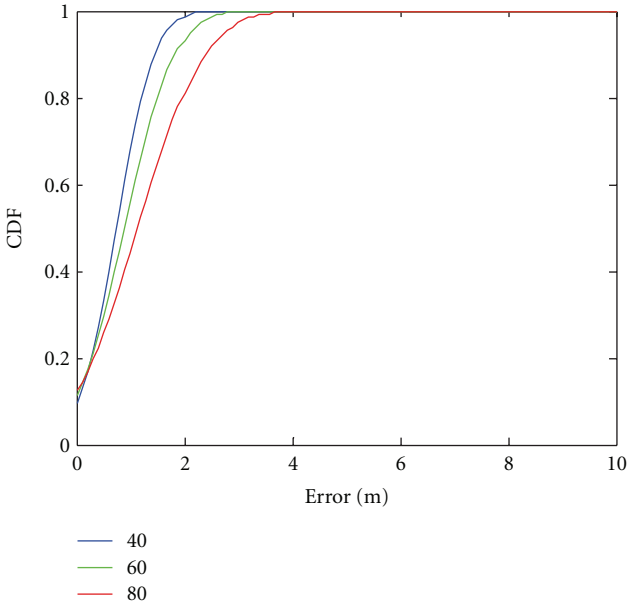


Figure 9: Error distribution of the samples collected in the parking lot.

different vehicles, we park at a different parking space in the chosen subarea each time. Specially, the corresponding GPS trajectory is recorded and the relative logical location (logical number relative to the reference point) of the parked space is recorded manually. Apart from this, the data collection process was not controlled in any other manner (e.g., speed,

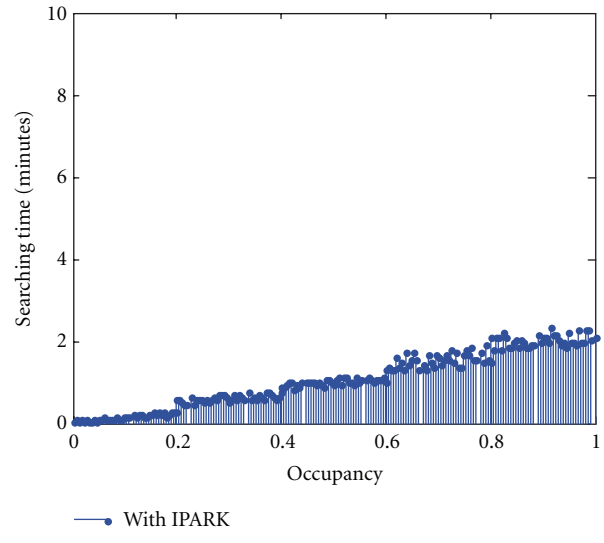


(a) Data distribution

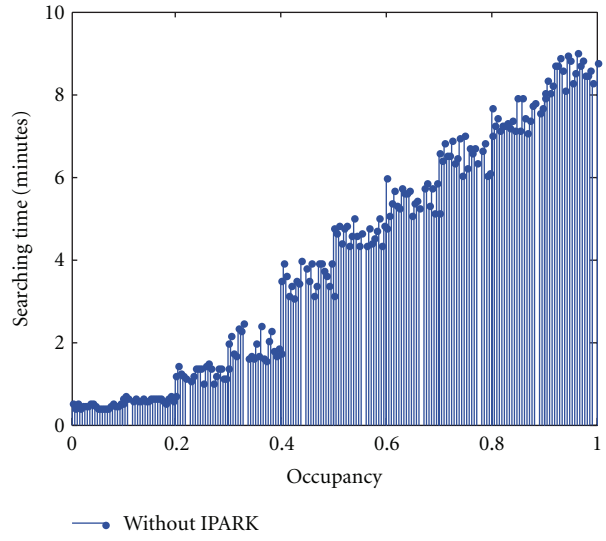


(b) CDF of the estimation error

Figure 10: Error distribution of the samples collected in the virtual parking lot.



(a) With IPARK



(b) Without IPARK

Figure 11: Searching time under different parking occupancy.

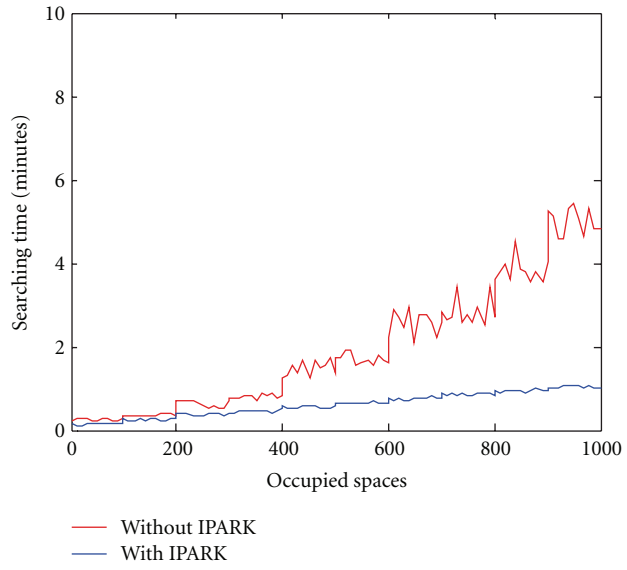
obstacles, etc.). After the survey, the data provided by GPS is accessed via an USB serial port and processed on a computer.

To verify the effectiveness of the proposed logical location estimation algorithm, we analyze the collected data in the following way.

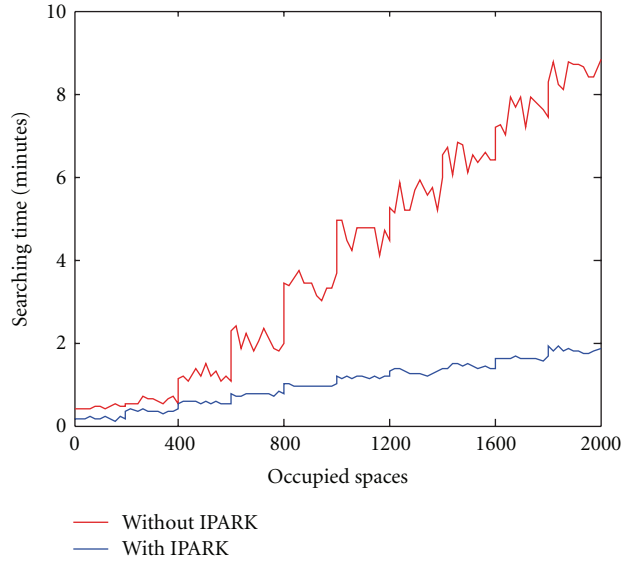
Let $V = [v_1; v_2; \ldots; v_N]$ denote the set of N locations on a road corresponding to the mid-point of N different parking spaces along this road segment, and $X_V = [X_1; X_2; \ldots; X_N]$ the corresponding random variables describing the estimated distance relative to the reference point at these locations. Assume err(i) is a certain error function measuring the distance between the real value and the estimate at the ith location, which is denoted by $\text{err}(i) = X_i - \overline{X}_i$. We assume err($i$) to be Gaussian and perform analysis.

Denote the mean error at the ith location by μ_i, $i = 1; 2; \ldots; N$, then, the sample mean is $\overline{\mu} = (1/N)\sum_{i=1}^{n} \mu_i$, and the sample variance is $\sigma^2 = (1/N)\sum_{i=1}^{n}(\mu_i - \overline{\mu})^2$. The possibility density function of err(i) is $p(x) = (1/\sigma\sqrt{2\pi})e^{-(x-\overline{\mu})^2/2\sigma^2}$, and the cumulative distribution function is $F(x, \overline{\mu}, \sigma) = (1/\sigma\sqrt{2\pi})\int_{-\infty}^{x} e^{-(x-\overline{\mu})^2/2\sigma^2} dt$.

Based on the above analysis, we investigate the error distribution of the collected samples, with the results are shown in Figures 9(a) and 9(b). Figure 9(a) is a scatter plot showing the estimated relative distance against the actual relative distance of the sample spaces. Figure 9(b) is the cumulative distribution function of the estimation error. We find that about 90% of the estimation error is less than 1.5 m, which is small enough to uniquely determine a parking space.



(a) 1000 spaces



(b) 2000 spaces

Figure 12: Searching time under different capacity.

The above experiment illustrates that our logical location estimation scheme works extremely well in large parking lot. As our parking lot map generation scheme directly relies on the reported logical location of the parked vehicles, we could conclude that the parking lot map could be automatically establish over long periods of time using our scheme.

To determine the threshold for reference point selection, we carry out another group of experiment. As the above parking lot does not have more than 50 consequent spaces along the same road segment, we take a road with a length of 200 m as a line of parking spaces. We assume that there are 80 parking spaces along this road (with each parking space has a width of 2.5 m) and sample similarly like the way in the parking lot. Finally, we analyze the error distribution of these samples, with the results shown in Figures 10(a) and 10(b). We find that within 40 spaces (100 m), 90% of the estimation error is less than 1.5 m. Within 60 spaces (150 m), only 80% of the estimation error is less than 1.5 m. Within 80 spaces (200 m), the situation becomes even worse. Therefore, we take 100 m as the threshold. Out of this threshold, the reference point has to be redefined.

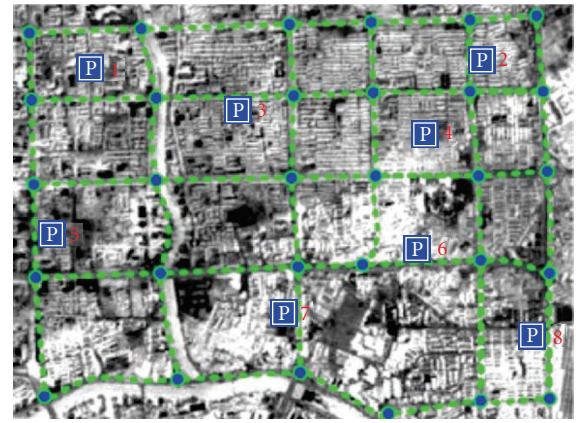
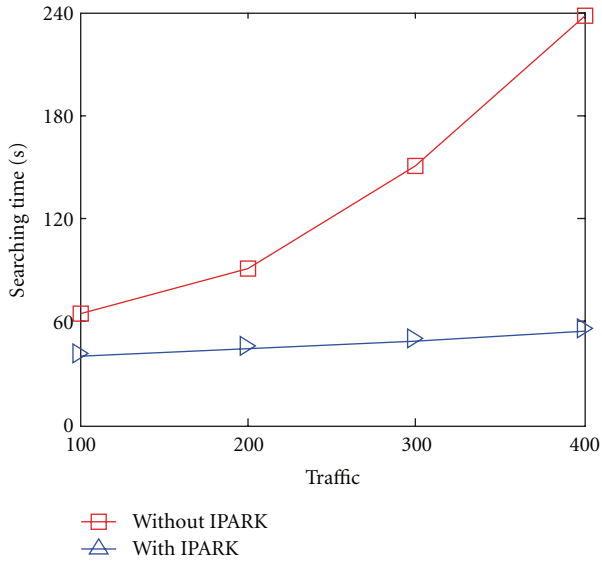


Figure 13: Map used in our simulation.

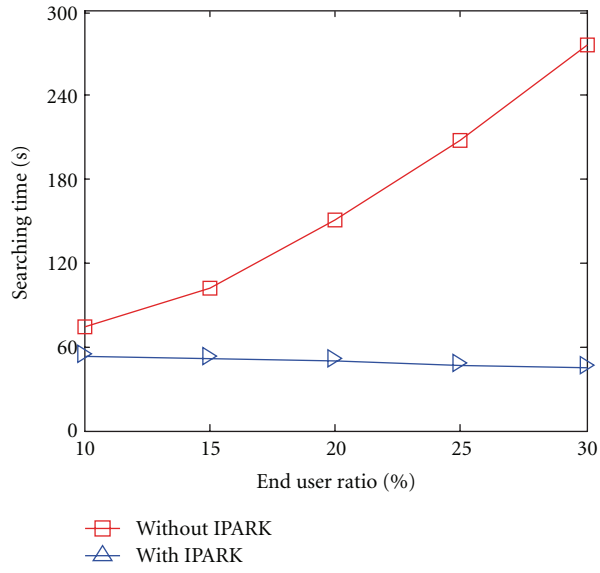
5.2. Simulations. In this section, simulations are conducted in NS-2.33 to verify the effectiveness of the proposed IPARK scheme, including the parking navigation inside the parking lot and the parking data dissemination outside the parking lot.

5.2.1. Parking Navigation. For the parking navigation, we investigate the searching time of a vacant parking space with and without the assistance of IPARK, respectively. The large parking lot used in our simulation is the one in Figure 8. For simplicity, we assume that a driver prefers to park in an available parking spot close to the entrance. After entering the parking lot, each vehicle is driving with a randomly fluctuated speed in a range of 10 percent centered at the parking lot speed limit. With SPARK, the driver will drive to a vacant parking space closest to the entrance directly. Without SPARK, the driver will be searching for a parking spot close to the entrance of the mall and keep circling around until finding the nearest available parking space.

First, we investigate the impact of the occupancy ratio of the parking lot on the searching time. We test in the parking lot with IPARK navigation and without IPARK navigation. For each case, we test 20 times, and the average searching time is considered. As shown in Figure 11, without IPARK navigation schme, with the increase of the occupancy ratio, the searching time (ST) for an vacant parking space increases significantly after the occupancy ratio reaches 40%. Especially, when the occupancy ratio is above 80%, the time it takes a driver to find an vacant parking space is too long. However, with the assistance of the proposed parking navigation scheme, the searching time (ST) for an vacant parking space becomes much lower. Second, we investigate the impact of the capacity of a parking lot on the searching



(a) ST under different traffic



(b) ST under different end user ratio

Figure 14: Searching time of an available parking lot.

time. We study the searching time under 1000 spaces and 2000 spaces, respectively. As shown in Figure 12, with the assistance of IPARK, the searching time just increase slightly with the capacity of the parking lot changing from 1000 to 2000. However, without the assistance of IPARK, the searching time increase obviously with the capacity of the parking lot varying from 1000 to 2000.

5.2.2. Parking Data Dissemination. For parking data dissemination, we study the searching time of an available parking lot with and without IPARK, respectively. As shown in Figure 13, we extract a regional urban area with the range of 3200 m × 2200 m from a real street map of Chengdu. We deploy different vehicle numbers, that is, 100, 200, 300, and 400, to the map, with the average speed ranges of 40 to 80 kilometers per hour. Among these vehicles, 20% of them have a random destination within this area and they would find a parking spot to park while arriving at the destination. For the other 70% vehicles, they move back and forth within this area. According to our survey, there are 8 parking lots within this area, with 6 roadside parking lots and 2 off-street parking lots, as are marked in Figure 13. The capacity of each parking lot is listed in Table 1. Specially, for the roadside parking lot, the initial occupancy ratio is set as a random value between 90–95%; for the off-street parking lot, the initial occupancy ratio is set as a random value between 80–85%. During the simulation, the occupancy ratio of each parking lot changes with the parking behaviors of the vehicles. The radio range is set at 250 m, and the MAC protocol is 2 Mbps 802.11. The beacon interval is 1 s, and all messages have a uniform size of 1 kb. We assume that the parking lot clusters are established at the beginning of simulation, and they share the parking data with each other at an interval of 60 seconds.

Table 1: Capacity of the eight parking lots.

i	1	2	3	4	5	6	7	8
Capacity	124	46	62	136	37	54	42	58

With IPARK, a vehicle could get parking data from a parking lot by the means presented in Section 4. Once obtaining the parking data, the driver would directly drives towards the closets parking lots with available parking spaces. Without IPARK, a vehicle would drive towards the closest parking lots while having parking demand. While arriving at a parking lot, if the current parking lot has no vacant parking space, the driver has to leave for the second closest parking lot. This process will be repeated until a parking lot with available parking spaces is founded.

The searching time is defined as the time interval between the instant when a driver has a parking demand and the instant when it arrives at one of the parking lots with available parking spaces. The average searching time is the mean value of that of all the vehicles searching for parking spaces. We test the average searching time of an available parking lot with and without IPARK under different vehicle density, with the results are shown in Figure 14(a). We notice that with the assistance of IPARK, it takes a driver less than 55 s to arrive at a parking lot with available parking space, which is tolerable to most of the drivers. However, without the assistance of IPARK, the searching time of an available parking lot is 65–238 seconds. In addition, we observe that while the vehicle number varying from 100 to 400, the searching time with IPARK increases slightly, while the searching time without IPARK increases obviously. Moreover, we investigate the impact of end user ratio on the searching time delay of the two schemes. As reported in Figure 14(b), with the growing of end user ratio, the searching time is decreased. This is because a vehicle could acquire the response more easily with more end user. Without IPARK, with the growing of end user ratio, the number of vehicle searching for parking spaces is increased, and the probability for the parking lot being full is also increased. Thus, the searching time becomes even longer.

6. Conclusion

We propose IPARK to achieve intelligent parking guidance over infrastructureless VANETs. The basic idea of IPARK is simple: if some of the parked vehicles are willing to provide PVA (Parked Vehicle Assistance) services, why not let them obtain the parking statistics in real time and offer guidance for vehicles searching for parking spaces? In this paper, we group individual parked vehicles within the same parking lot into one cluster, develop an automatic parking lot map generation scheme, and investigate efficient parking navigation and parking data dissemination based on the parking lot cluster. As a novel parking guidance scheme, IPARK successfully establishes the effective management and collaborative communication without the support of any infrastructure. At last, the analysis on real parking data illustrates the effectiveness of the proposed logical location estimation scheme and the simulation results demonstrate that IPARK could help a vehicle driver find a vacant parking space at a small delay.

Acknowledgments

This work is supported by National Science Foundation of China under Grant nos. 61170256, 61103226, 61103227, 61173171, and 61272526 and the Fundamental Research Funds for the Central Universities under Grant nos. ZYGX2011J060, ZYGX2010J074, and ZYGX2011J073.

References

[1] D. Schrank and T. Lomax, "Annual urban mobility report," 2005, http://www.pittsburghregion.org/public/cfm/library/reports.

[2] D. Shoup, "Cruising for parking," *Access*, vol. 30, pp. 16–22, 2007.

[3] J. Chinrungrueng, U. Sunantachaikul, and S. Triamlumlerd, "Smart parking: an application of optical wireless sensor network," in *Proceedings of the International Symposium on Applications and the Internet Workshops (SAINTW '07)*, pp. 66–69, January 2007.

[4] Y. Z. Bi, L. M. Sun, H. S. Zhu, T. X. Yan, and Z. J. Luo, "Parking management system based on wireless sensor network," *Acta Automatica Sinica*, vol. 32, no. 6, pp. 877–968, 2006.

[5] http://sfpark.org/.

[6] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: a new VANET-based smart parking scheme for large parking lots," in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM '09)*, pp. 1413–1421, April 2009.

[7] R. Panayappan, J. M. Trivedi, A. Studer, and A. Perrig, "VANET-based approach for parking space availability," in *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks (VANET '07)*, pp. 75–76, September 2007.

[8] *Advanced Parking Management Systems: A Cross-Cutting Study-Taking the Stress Out of Parking*, Federal Highway Administration, Intelligent Transportation Systems, U.S. Department of Transportation, 2007.

[9] M. Caliskan, D. Graupner, and M. Mauve, "Decentralized discovery of free parking places," in *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET '06)*, pp. 30–39, September 2006.

[10] S. Mathur, T. Jin, N. Kasturirangan et al., "ParkNet: drive-by sensing of road-side parking statistics," in *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '10)*, pp. 123–136, June 2010.

[11] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: stopped cars are not silent," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 431–435, April 2011.

[12] S. Olariu, I. Khalil, and M. Abuelela, "Taking VANET to the clouds," *International Journal of Pervasive Computing and Communications*, vol. 7, pp. 7–21, 2011.

[13] C. Morency and M. Trépanier, *Characterizing Parking Spaces Using Travel Survey Data*, CIRRELT, 2006.

[14] A. Adiv and W. Wang, "On-street parking meter behavior," *Transportation Quarterly*, vol. 41, pp. 281–307, 1987.

[15] B. Albanese and G. Matlack, "Utilization of parking lots in Hattiesburg, Mississippi, USA, and impacts on local streams," *Environmental Management*, vol. 24, no. 2, pp. 265–271, 1999.

[16] Y. Bao, H. Xu, and Z. Liu, "Vector map geo-location using GPS tracks," in *Cartographic Theory and Models (Geoinformatics '07)*, vol. 6751 of *Proceedings of SPIE*, May 2007.

[17] G. R. Jagadeesh, T. Srikanthan, and X. D. Zhang, "A map matching method for GPS based real-time vehicle location," *Journal of Navigation*, vol. 57, no. 3, pp. 429–440, 2005.

[18] J. F. Kurose and K. W. Ross, *A Top-Down Approach Featuring the Internet*, 2012.

[19] A. Vahdat and D. Becker, *Epidemic Routing for Partially Connected Ad Hoc Networks*, 2000.

[20] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 319276, 8 pages
doi:10.1155/2012/319276

Research Article

You Take Care of the Drive, I Take Care of the Rule: A Traffic-Rule Awareness System Using Vehicular Sensors and Mobile Phones

Xing Zhang, Jidong Zhao, Jinchuan Tang, and Bang Liu

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

Correspondence should be addressed to Xing Zhang, zhangsimba@gmail.com

Received 28 September 2012; Accepted 23 October 2012

Academic Editor: Nianbo Liu

Traffic rules are used to regulate drivers' behaviours in modern traffic systems. In fact, all driving behaviours are presented by vehicles' behaviours. If vehicles have awareness of their behaviours, it is possible that traffic rules are able to regulate vehicles instead of drivers. There are three advantages of vehicle regulation: (1) without worrying about violations of traffic rules and searching for traffic signs, drivers can pay more attention on emergency situations, such as jaywalking. (2) Many traffic violations are due to attention distraction; machines do not have the attention issues; therefore they can provide good traffic-rule obeying. (3) New traffic rules can be spread and applied more quickly and effectively through the Internet or Vehicular Ad hoc NETworks (VANETs). In this paper, we propose a novel traffic-rule awareness system using vehicular sensors and mobile phones. It translates traffic rules into combinations of vehicular sensors, GPS device, and Geography Information System (GIS); the system can tell whether a driver violates the traffic rules and help him to amend his driving behaviour immediately. Experiments in real driving environments show that our system can be aware of the traffic rules accurately and immediately.

1. Introduction

During the year 2011, there were 11856 traffic accidents that happened in Sichuan province of China and 95% of the accidents are caused by traffic violations [1]. If traffic-violation rate can be significantly decreased, a lot of lives can be saved from traffic accidents. In fact, most traffic violations are not on purpose; the reason of high traffic-violation rate is that the traffic rules are designed to regulate drivers' behaviours. As long as drivers are human beings, they will suffer from memory issues and attention distraction. Their memory and concentration will be severely affected by mood, alcohol, drugs, and environment. Even a short conversation during driving will distract drivers and cause unnecessary traffic violations.

Google self-driving car [2] is a good attempt to decrease the traffic-violation rate, because machines do not have the memory and concentration issues like humans. However, google self-driving car only focuses on self-driving; it has not yet taken traffic rules into consideration except for traffic lights. Some other works like [3] monitor dangerous driving behaviours like aggressive turns, acceleration, and braking and help drivers to correct these unsafe behaviours. However, they did not take traffic rules into consideration. Actually, even if driving behaviours are not aggressive, as long as the driver violates the traffic rule, there is still a big possibility that traffic accidents may happen.

If vehicles have awareness of whether they violate traffic rules, they can warn the drivers about the violations and help them to amend their driving behaviours. In fact, although it is a human who drives the vehicle, the driving behaviours are presented by vehicles' behaviours. No matter who drives a vehicle or how he drives a vehicle, the driving behaviours can be described by two factors of vehicle's behaviours: speed and direction. Traffic rules are location based; different location has different traffic rule; therefore location is the third factor to describe driving behaviours for traffic rules. For example, if we want to describe a traffic rule of which speed limit is 40 kilometers per hour and one way from south to north at road A, we can set the three factors as Table 1 shows.

Modern mobile phones are equipped with various sensors like accelerometer, gyroscope, and orientation. They also

Table 1: Three factors of traffic rule.

Location	Seed (km/h)	Direction
Road A	<40	south to north

equipped with GPS localization devices. Accelerometer can be used to monitor the speed of a vehicle; gyroscope and orientation can be used to monitor the direction changes. GPS device can be used to monitor the location. It seems that mobile phones alone are enough to build the traffic-rule awareness system. In fact, traffic rules not only regulate the driving vehicles, but also regulate the parking vehicles. We need to know whether a vehicle is parking or not. No speed does not mean the vehicle is parking; it can be waiting for traffic lights or something else. We need to know whether the engine has been flamed out for a while to judge if the vehicle is parking, therefore we have to take advantage of vehicular sensors to monitor the engine states.

In this paper, we propose a novel traffic-rule awareness system. It translates traffic rules into combinations of the three aforementioned factors. It will judge whether there is the tendency of violating the traffic rules while driving and warn drivers of this tendency. Therefore, it will free drivers from searching for traffic signs and remembering all the traffic rules; drivers can put more energy on driving. Mobile phones can provide 2G/3G/4G data connection for downloading translated traffic-rule from remote servers or construct Mobile Ad-hoc NETwork (MANET) to get the traffic rules from neighbors.

The rest of paper is organized as follows. Section 2 describes the details of system design. Section 3 shows the experiment results. Related works and conclusions will be given in Sections 4 and 5.

2. System Design

As mentioned before, mobile phones alone can not meet all the requirements of the system; vehicular sensors are good complements. On-Board Diagnostics (OBD) is widely equipped in modern automobiles. The OBD implementations use a standardized digital communications port to provide real-time data in addition to a standardized series of diagnostic trouble codes [4], or DTCs, which allow one to rapidly identify and remedy malfunctions within the vehicle [5]. With the help of OBD scanner, the values of vehicular sensors can be easily obtained by our system.

2.1. System Overview. As Figure 1 shows, there are three modules in the system: communication module, process module, and alert module. Communication module is used to download traffic rules from remote server via internet or neighbors via MANET; also it provides download service for its neighbors. Process module contains two parts: information collecting part and information processing part. Information collecting part collects three kinds of information which are related to the vehicle: speed, direction, and location. Speed can be obtained from accelerometer of mobile phone; however, as mentioned before, we also need to know whether the vehicle is parking when the speed is zero. Engine state is a way to judge the parking state, and it can be read from OBD scanner. ELM 327 Diagnostic Scanner [6] is a common off-the-shelf product which provides an interface to access the OBD system. The scanner uses a female 16-pin SAE J1962 connector to connect to the OBD system, and it uses Bluetooth to connect to a control device such as mobile phone. Gyroscope and orientation which are embedded in mobile phones are used to generate the direction information. GPS device which is embedded in mobile phone is used to obtain the location information. The information process part will compare the collected information with the traffic rules stored in DB to judge whether the driver violates the traffic rules. Once there is a trend of traffic violation, the violation alert module will warn the driver.

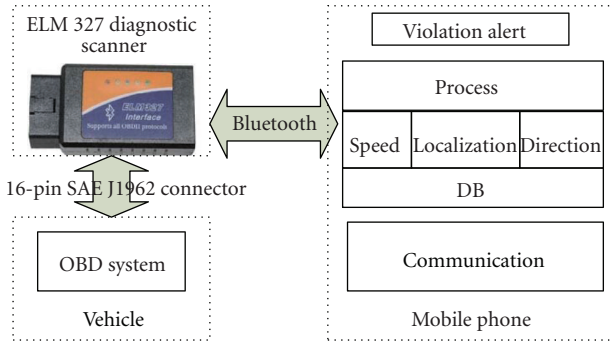


Figure 1: System overview.

2.2. Traffic Rules Translation. As mentioned before, vehicles' behaviours can be described by three factors: speed, direction, and location. In this research, we use a collection of triples *TR* (*speed*, *direction*, and *location*) to represent the traffic rules. *Speed* is a 2-tuple *Speed* (*lower*, *upper*); the *lower* and *upper* elements are the lower and upper bounds of an interval of speed limit. For example, for the highway, the speed limit is not less than 80 km/h and not more than 120 km/h, so the 2-tuple is *Speed* (80, 120). *Direction* is a 2-tuple *Direction* (*road direction*, *turns*); *road direction* is the direction allowed on the road and it is a set of five values: NORTH, EAST, WEST, SOUTH and TWOWAY. If the road is one way street, *road direction* gives the heading direction. If the road is two way street, *road direction* gives the TWOWAY value. *Turns* is a quadruples *Turn* (*Left*, *Right*, *U*, *Lane Change*), each element indicates whether this kind of turn is allowed. *Location* is a 2-tuple *Location* (*road*, *area*). Element *road* is the road name on the map; *area* is a 2-tuple *area* (*from*, *to*) which indicate the area of *road* applying this traffic rule. *From* and *to* are GPS coordinates. For example, as Figure 2 shows, Chengdu road is an one-way street and the speed limit is not more than 40 km/h. Between Locations 2 and 3, vehicles can choose to turn right or go straight. The traffic rules at Chengdu road can be translated to three *TR* (*speed*, *direction*, *location*) triples as Table 2 shows. We can see that the lower bound of *Speed* (*lower*, *upper*) is 1; the reason is that parking on the road is not allowed. If a location allows parking, the lower bound of speed will be set to zero.

Table 2: Traffic-rule triples.

TR (*speed*, *direction*, *location*)
TR (*speed* (1, 40), *direction* (*NORTH*, *turn* (0, 0, 0, *lane change*)), *location* (*"Chengdu road," area* (*Location* 4, *Location* 3)))
TR (*speed* (1, 40), *direction* (*NORTH*, *turn* (0, *RIGHT*, 0, *lane change*)), *location* (*"Chengdu road," area* (*Location* 3, *Location* 2)))
TR (*speed* (1, 40), *direction* (*NORTH*, *turn* (0, 0, 0, *lane change*)), *location* (*"Chengdu road," area* (*Location* 2, *Location* 1)))

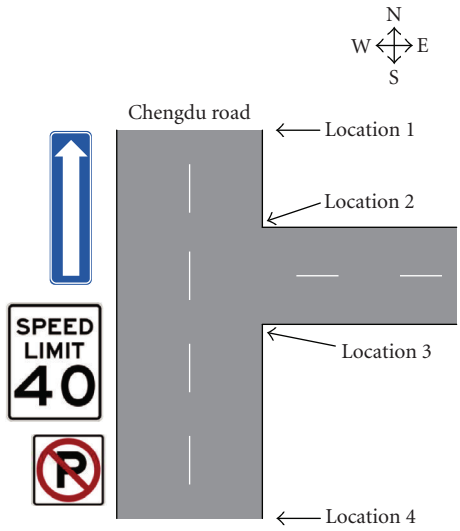


Figure 2: Traffic rules example.

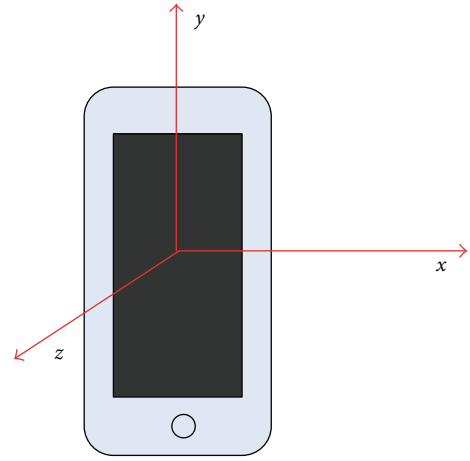


Figure 3: Coordinate system of mobile phone.

Table 3: Mapping relation between Azimuth and *road direction*.

North	East	South	West
[0°, 45°), [315°, 360°)	[45°, 135°)	[135°, 225°)	[225°, 315°)

2.3. Information Collection. There are three kinds of information need to be collected: speed, direction and location. The process module uses OBD parameter IDs (PIDs) codes to request data from a vehicle. SAE standard J/1979 defines many PIDs, but manufacturers also define many more PIDs specific to their vehicles [4]. In this research, we use standard PIDs for good compatibility.

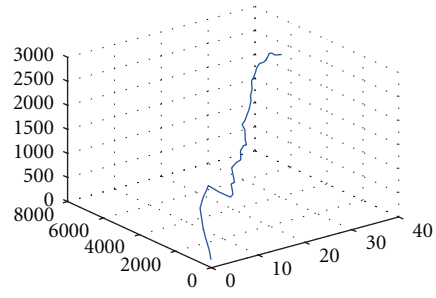
2.3.1. Speed. Speed can be calculated through GPS coordinates or accelerometer, but a more easier way is that sending a PID code VEHICLE_SPEED to OBD scanner, the OBD system will return the vehicle speed. However, speed is not enough for traffic-rule translation; we also need to know the engine states. To detect whether the vehicle is parking, we need to know whether the engine is off for a period of time. There is an OBD PID code ENGINE_RPM which indicate the rotation speed of engine, if the engine is off, the rotation speed is zero. Therefore, a vehicle is parking when VEHICLE_SPEED is zero and ENGINE_RPM is zero for a period of time. As (1) shows, only when the vehicle is parking, we set the speed of the vehicle to zero, otherwise the speed is larger than zero:

$$\text{speed} = \begin{cases} 0 : \text{VEHICLE\_SPEED} = 0, \\ \quad \text{ENGINE\_RPM} = 0,\ T \geq t, \\ \text{VEHICLE\_SPEED} : \text{VEHICLE\_SPEED} > 0, \\ \qquad\qquad\qquad \text{ENGINE\_RPM} > 0, \\ 1 : \text{Otherwise.} \end{cases} \quad (1)$$

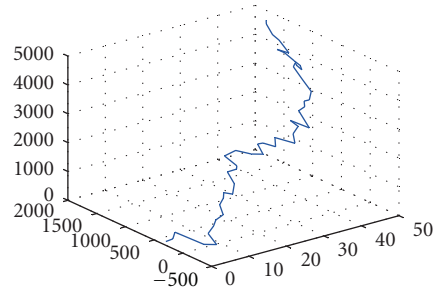
2.3.2. Direction. To detect the direction, we need three kinds of sensors: accelerometer, orientation, and gyroscope. Before discussing about how to collect direction information, we will first discuss the coordinate system of mobile phone. As Figure 3 shows, the x-axis is horizontal and points to the right, the y-axis is vertical and points up, and the z-axis points towards the outside of the front face of the screen [7]. Data from all the sensors embedded in mobile phone is three-dimensional and obeying this coordinate system.

To obtain the value of *road direction*, we need to use orientation sensor. The data from orientation sensor follows three dimensions: Azimuth (degrees of rotation around the z-axis); Pitch (degrees of rotation around the x-axis); Roll (degrees of rotation around the y-axis) [8]. In this research, we fix the mobile phone head forward and flatwise; therefore, Azimuth is the degree we needed. Table 3 shows the mapping relation between Azimuth and *road direction*.

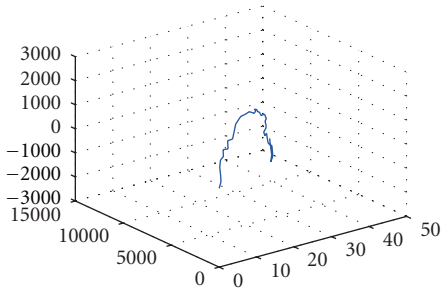
We can use accelerometer to detect the turn event of driving behaviour, as Figure 4 shows; three kinds of turns, lane change, left turn and U turn are easily distinguished from each other. There are many works discussed about the driving patterns and gestures recognition [9–11]. However, we cannot tell the driver: "You cannot turn left" when he already turned left. We have to make prediction of drivers' turning intentions. Gyroscope is a very sensitive device to detect the rotation action; we can recognize the turning intentions at the very beginning of drivers' turning



(a) Lane change



(b) Left turn



(c) U turn

Figure 4: Three patterns of turning.

behaviours by taking full advantage of gyroscope. Detailed discussion is in the next section.

2.3.3. Location. Modern mobile phones are equipped with GPS devices; therefore, the location of vehicles can be easily obtained. With the help of Geography Information System (GIS), we can locate the vehicle on the map by inputting GPS coordinates.

2.4. Challenges. The main role that the traffic-rule awareness system plays is to warn drivers about the possible future traffic violations. Therefore, different from other works of recognizing driving patterns which are already happened, the system must predict drivers' intentions. Existing researches in prediction of driving intentions are mostly using cameras which would bring inconvenience and privacy issue to the drivers. In this research, we only take advantage of sensors from mobile phones to achieve the prediction.

2.4.1. How to Predict Drivers' Intention? According to velocity formula $v = v_0 + at$, to predict the velocity of a vehicle,

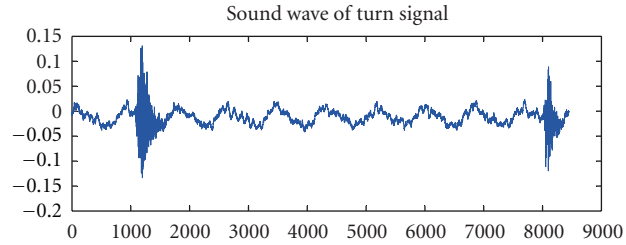


Figure 5: Sound wave of turn signal.

we need to know current velocity v_0 and the acceleration a. v_0 can be obtained through OBD system a can be obtained through accelerometer; therefore, v can be predicted.

In the coordinate system of mobile phone, data from accelerometer are three dimensions: A_x (acceleration at x-axis), A_y (acceleration at y-axis), and A_z (acceleration at z-axis). Acceleration can be calculated as (2) shows. To predict the velocity v, we need to average a, as (3) shows, to reduce measurement error. t_0 is the current time and t is the time interval of prediction. Equation (4) gives the final prediction. Once the prediction v does not belong to *Speed* (*lower*, *upper*), a warning will be alerted:

$$a_t = \sqrt{A_x^2 + A_y^2 + A_z^2}, \tag{2}$$

$$\overline{a} = \frac{\sum_{i=0}^{n} a_{t_0 - i}}{n+1}, \tag{3}$$

$$v = v_0 + \overline{a}t. \tag{4}$$

Predicting the turning event is a great challenge. Firstly, turning event is a short-time process; the prediction must be made at the very beginning to give enough response time for drivers. Secondly, normal driving behaviours contain a lot of turning actions, such as dodging and parking. To distinguish these turning actions from real turning event, we have to take advantage of turn signal. Turning on turn signal is a standard process in turning event; every time drivers want to make turns, they will firstly turn on the turn signal. Therefore, if we can detect the turn signal, we can recognize the real turning event.

2.4.2. How to Detect the Right/Left Turn Signal? OBD system is no doubt the first choice of detecting the left/right turn signal; it can accurately tell whether the left or right turn signal is on. For example, Chevrolet Corvette OBD-II Codes contain the codes to return the information of left and right turn signals [12]. However, the OBD codes are not the standard OBD codes; they are exclusive to a certain vehicle manufacturer. To find a more compatible way to recognize the turn signal, we use a method called Sound Cross-Correlation (SCC).

The main idea of SCC is that the sound features of turn signal are precaptured; every time a sound signal received, the system will cross-correlate the received sound signal with the sound of turn signal. When a spike occurs in the correlation, it means THAT turn signal is in the received sound signal. Figure 5 shows the sound wave feature of turn

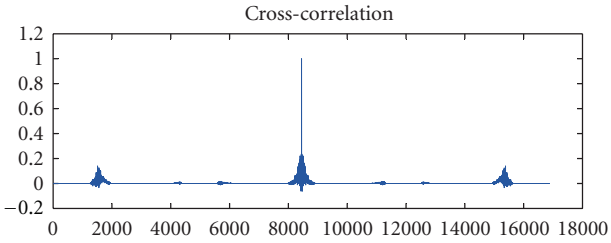


Figure 6: Cross-correlation.

signal. Cross-correlation is a widely used method in recent research works [13]. Suppose the known symbol pattern of sound wave of turn signal is ts of length L, $X(n)$ is the complex number representing the nth received symbol, then the cross-correlation at a shift position p is

$$C(ts, X, p) = \sum_{k=1}^{L} ts^*[k]X[k+p], \tag{5}$$

ts^* is the complex conjugate of ts and $C(ts, X, p)$ is the correlation coefficient. Once the $X[n]$ is aligned with ts, there will be a sudden spike in the correlation as Figure 6 shows.

SCC cannot distinguish right turn and left turn; however, it will not affect the recognition of left turn and right turn in out system, because turn signal is only used for detecting drivers' turning intention.

2.4.3. How to Distinguish Lane Change, Left Turn, and U Turn? Only if we can hack in drivers' brains, otherwise we cannot know whether they want to turn left or change lane or make U turn in advance. Now the turn signal is detected, the driver's turning intention is known, and the problem is there are three kinds of turning; all of them are related to the traffic rules; therefore the three kinds of turning must be distinguished. Only when the driver makes turns, we can know what kind of turns he makes. If we recognize the kind of turning after the turning behaviour is finished like the pattern recognition or gesture recognition does, it is meaningless to issue a warning to the driver after the traffic violation. The challenge is that the recognition must be finished at the very beginning of the turning behaviour.

The movement of turning can be divided into two kinds of movements: translation movement and spin movement. Spin movement is the key factor to distinguish turning style. Accelerometer measures the acceleration of vehicles; it can be used to detect the translation movement, but it can not detect the spin movement; therefore it is not suitable for turning detection. Gyroscope is a sensitive device which can detect angular speed in three dimensions according to the coordinate system of mobile phone. If the mobile phone is fixed flatwise, the rotation around z-axis can be used to reflect the spin movement; therefore gyroscope can be used to distinguish the turning style.

Figure 7 shows the gyroscope readings of lane change (from right to left), left turn, and U turn (from right to left); we choose the Azimuth reading to reflect the spin movement. We can see that the first positive peak of three turning style

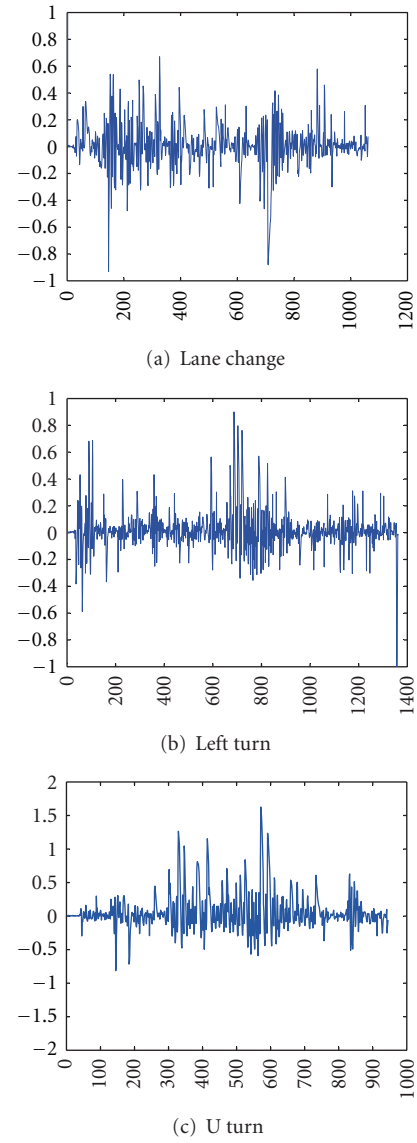


Figure 7: Gyroscope Azimuth readings of three turning patterns.

all appears at the beginning (there is a process of approaching to right side before U turn, so the first positive peak appears a little late). The peak represents the start of spin movement of the three turning styles, and the three turning style have different peak intervals. The peak interval of lane change is $(0.5, 0.7)$; the peak interval of left turn is $(0.7, 0.9)$; the peak interval of U turn is $(1.2, 1.6)$. Therefore, we can distinguish the three turning styles at the beginning of turning by which intervals of the first positive peak belong to.

3. Implementation and Evaluation

The experiment evaluates three aspects of the system: turn signal detection, turning event prediction, and traffic rules awareness. All the experiments are running in real driving environment. The automobile we use is Buick Regal which provides a standard OBD system. The mobile phone we use is Lenovo S2 which has 1 GB ram and 1.4 GHz CPU core

frequency which can provide enough processing ability, also the phone has all the sensors we need including orientation sensor, acceleration sensor, and gyroscope sensor.

3.1. Turn Signal Detection. Cross-correlation can well recognize a signal in a mixed signal; however, it has difficulties in recognizing when SINR is low; here we treat the sound wave of turn signal as Signal, the others as Interference and Noise. We test the cross correlation in three different environments: quiet, talking, and playing music. Figure 8 shows the results of cross correlation in the three environments. We can see from Figure 8(a) that cross correlation can well recognize turn signal in a quiet environment; four separate spikes indicate four turn signals; in a talking environment as Figure 8(b) shows, human voices sometimes submerge the turn signals and make the recognition impossible. However, talking is a discrete process, it has nontalking gap for us to recognize the turn signal, and there is a spike at the beginning which indicates the existence of turn signal. Other spikes are caused by interference which must be excluded; the spikes are usually mixed together; therefore we choose the separated spikes as the indication of turn signal; Figure 8(c) shows the cross correlation in a music playing environment; we can see that there is no separated spike; therefore recognition of turn signal is almost impossible in a loud and continuous interfering environment.

Loud and noisy driving environment is bad for driving safety; therefore, in most cases, driving environment is quiet and suitable for turn signal recognition.

3.2. Turning Event Prediction. We test the turning event prediction in the campus. Once the system recognizes any of three turning patterns, it will give a warning and the driver stops the car. Figure 9 shows the results of experiments; we can see that the car stops at the very beginning of turning movement and gives the driver chances to amend their driving behaviours.

3.2.1. Traffic Rules Awareness. We choose an area in the campus which is sparsely populated and convenient for us to do the tests. As Figure 10 shows, we choose four road segments and label them with four colors: red, green, blue, and yellow. Red road is for testing lane change and left turn restriction; green road is for testing U turn and one-way restriction; blue road is for testing speed limit; yellow road is for testing parking restriction. The traffic rules are listed as Table 4 shows.

The results show that our system has good awareness of traffic rules; however, there are a few misjudgments during the tests; when we do the same tests again at another time, misjudgments disappear. The reason probably is that the accuracy of GPS localization is sometimes not good; it will cause a vehicle at road B violates a rule at a neighbor road A.

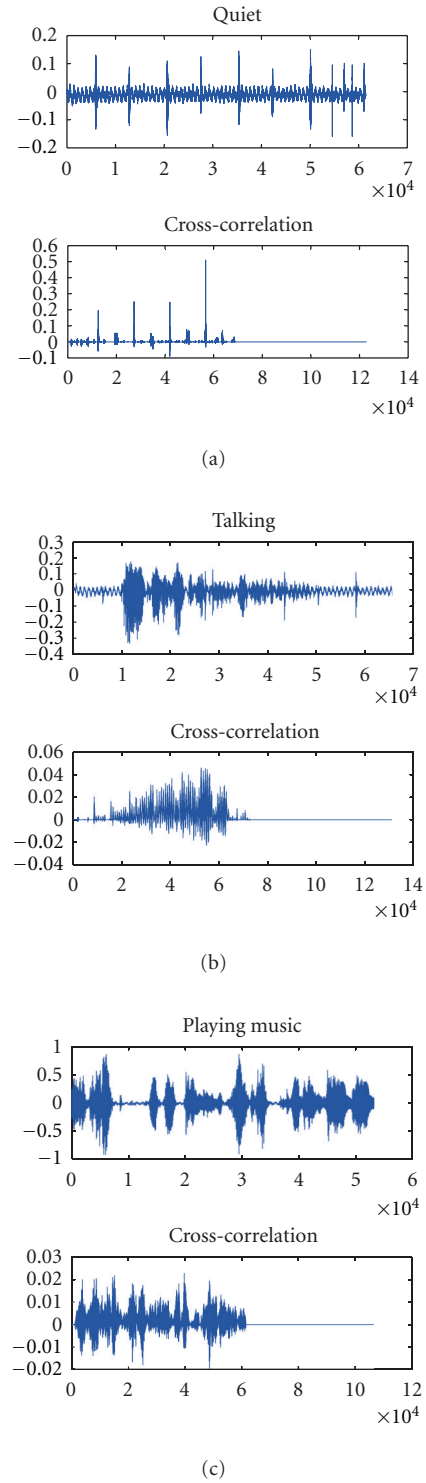


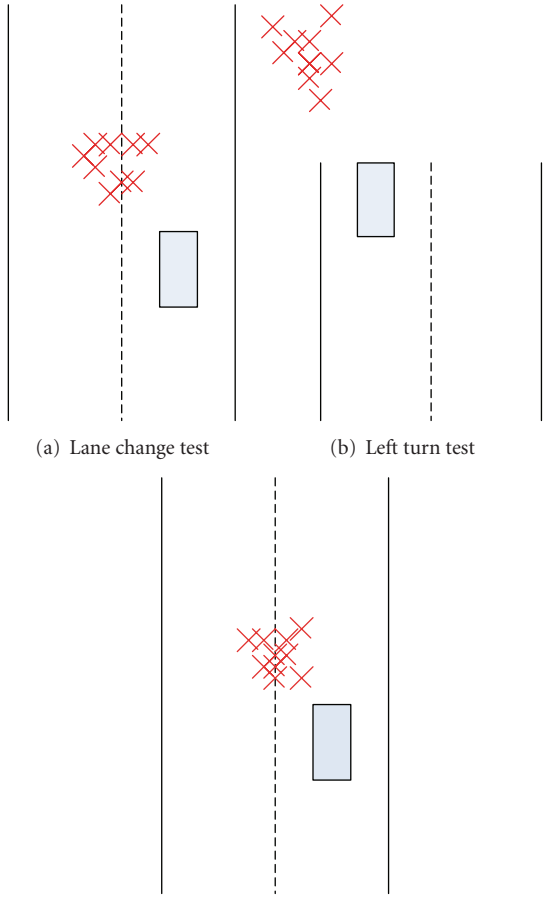
Figure 8: Cross-correlation of turn signal in different environments.

4. Related Works

Vehicle safety is a permanent topic; the main reason VANET and Vehicular Sensor Network are proposed as to enhance the safety. Traffic rules are closely related to driving safety in real driving environment; however, there are few researches take it into consideration. Most researches focus on recognizing drivers' driving behaviours and try to correct some dangerous behaviours.

Table 4: Traffic rules for tests.

TR (*speed, direction, location*)
TR (*speed* (0, 40), *direction* (*TWOWAY, turn* (*left*, 0, 0, *lanechange*))), *location* ("*red road*," *area* ((30.744140993322855, 103.92523527145386), (30.74342175330864, 103.92620086669922)))
TR (*speed* (0, 40), *direction* (*SOUTH, turn* (0, 0, *U*, 0))), *location* ("*green road*," *area* ((30.74342175330864, 103.92620086669922), (30.744620483682212, 103.92817497253418)))
TR (*speed* (0, 40), *direction* (*TWOWAY, turn* (0, 0, 0, 0))), *location* ("*blue road*," *area* ((30.744620483682212, 103.92817497253418), (30.74463892557145, 103.93083572387695)))
TR (*speed* (1, 40), *direction* (*TWOWAY, turn* (0, 0, 0, 0))), *location* ("*yellow road*," *area* ((30.74463892557145, 103.93083572387695), (30.74480490241566, 103.93188714981079)))



(a) Lane change test (b) Left turn test

(c) U turn test

Figure 9: Tests of turning event prediction.

Figure 10: Test map.

4.1. Intentions Prediction. Most researches use vision-based method to recognize body gestures to predict driving intentions. [14] uses camera to capture body pose to predict turning intention; [15] uses camera to capture head pose to judge whether the driver focuses on driving; [16] generates 3D visions to assist driving. [17] establishes a probabilistic model to analyse and predict driving intentions to assist brake controls.

4.2. Behaviours Recognition. Driving behaviours recognition can be used to detect some dangerous behaviours, like drunk drive [18]. Most recognition works are using accelerometer [19, 20]; it is enough for driving pattern recognition, but not enough to describe more complex driving behaviours. [3] takes advantage of more sensors: gyroscope, magnetometer, to detect more complex and dangerous driving behaviours.

4.3. Traffic Lights. Traffic lights detection is related to traffic rules; [21] uses camera from mobile phone to detect and predict the traffic light schedule. It is a complement to our research.

5. Conclusion

Traffic rules regulate drivers' behaviours so that there could be fewer traffic accidents and fewer lost lives. However, a human is not a machine; he has memory and attention troubles, he is easily affected by many things; therefore traffic violation rates never come down too much. Modern vehicles already have powerful processing ability and various sensors for sensing the world; they can be aware of the traffic rules for human so that drivers can focus on dealing with emergency situations such as jaywalking; it will no doubt significantly decrease traffic-accident rates and save thousands of lives.

However, how to predict drivers' intentions as soon as possible and improve localization accuracy will be the future works.

Acknowledgments

This work is supported by the National Science Foundation under Grant nos. 60903158, 61003229, 61170256, and 61103226 and the Fundamental Research Funds for the Central Universities under Grant nos. ZYGX2010J074 and ZYGX2011J102.

References

[1] Sichuan traffic accidents statistics, http://www.sc122.gov.cn/system/2012/02/01/013433326.shtml.

[2] How google self-driving car works, http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works.

[3] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA, 2011.

[4] On board diagnostics parameter ids, http://en.wikipedia.org/wiki/Table of OBD-II_Codes#Bitwise_encoded_PIDs.

[5] On board diagnostics, http://en.wikipedia.org/wiki/On-board-diagnostics.

[6] Elm327 diagnostic scanner, http://www.tmart.com/OBD-Diagnostics/.

[7] Android sensor coordinate system, http://developer.android.com/reference/android/hardware/SensorEvent.html.

[8] Orientate sensor, http://developer.android.com/guide/topics/sensors/sensors_position.html.

[9] D. Mitrović, "Reliable method for driving events recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 198–205, 2005.

[10] G. A. ten Holt, M. J. Reinders, and E. A. Hendriks, "Multi-dimensional dynamic time warping for gesture recognition," in *Proceedings of the 13th Annual Conference of the Advanced School for Computing and Imaging*, 2007.

[11] R. Muscillo, S. Conforto, M. Schmid, P. Caselli, and T. D'Alessio, "Classification of motor activities through derivative dynamic time warping applied on accelerometer data," *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '07)*, vol. 2007, pp. 4930–4933, 2007.

[12] 2002 chevrolet corvette obd-ii codes, http://www.obd-codes.com/trouble_codes/chevrolet/2002-corvette-obd-ii-codes.php.

[13] S. Sen, R. R. Choudhury, and S. Nelakuditi, "CSMA/CN: carrier sense multiple access with collision notification," in *Proceedings of the 16th Annual Conference on Mobile Computing and Networking (MobiCom '10)*, pp. 25–36, September 2010.

[14] S. Y. Cheng and M. M. Trivedi, "Turn-intent analysis using body pose for intelligent driver assistance," *IEEE Pervasive Computing*, vol. 5, no. 4, pp. 28–37, 2006.

[15] E. Murphy-Chutorian, A. Doshi, and M. M. Trivedi, "Head pose estimation for driver assistance systems: a robust algorithm and experimental evaluation," in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC '07)*, pp. 709–714, October 2007.

[16] C. Tran and M. M. Trivedi, "Towards a vision-based system exploring 3D driver posture dynamics for driver assistance: issues and possibilities," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV '10)*, pp. 179–184, June 2010.

[17] J. C. McCall and M. M. Trivedi, "Driver behavior and situation aware brake assistance for intelligent vehicles," *Proceedings of the IEEE*, vol. 95, no. 2, pp. 374–387, 2007.

[18] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan, "Mobile phone based drunk driving detection," in *Proceedings of the 4th International Conference on Pervasive Computing Technologies for Healthcare (Pervasive Health '10)*, March 2010.

[19] G. A. ten Holt, M. J. Reinders, and E. A. Hendriks, "Multi-dimensional dynamic time warping for gesture recognition," in *Proceedings of the 13th annual conference of the Advanced School for Computing and Imaging*, 2007.

[20] R. Muscillo, S. Conforto, M. Schmid, P. Caselli, and T. D'Alessio, "Classification of motor activities through derivative dynamic time warping applied on accelerometer data," *Proceedings of the 29th Annual International Conference of the IEEE*, vol. 2007, pp. 4930–4933, 2007.

[21] E. Koukoumidis, L. S. Peh, and M. R. Martonosi, "SignalGuru: leveraging mobile phones for collaborative traffic signal schedule advisory," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*, pp. 127–140, ACM, New York, NY, USA, July 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 387541, 7 pages
doi:10.1155/2012/387541

Research Article

Do Not Stuck at Corners: A Data Delivery Algorithm at Corners in Vehicular Sensor Networks

Jidong Zhao and Xing Zhang

University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Jidong Zhao, jdzhao@uestc.edu.cn

Received 8 October 2012; Revised 18 November 2012; Accepted 21 November 2012

Academic Editor: Nianbo Liu

Vehicular sensors capture a large amount of data, and a routing algorithm is needed to effectively propagate the data in vehicular sensor network (VSN). The existing routing algorithms in vehicular ad hoc network (VANET) can not be transplanted to VSN directly. After analyzing the mobility of vehicles, we find that the delivery time of vehicle to vehicle (V2V) or vehicle to infrastructure (V2I) is very short especially when a deliverer is at a corner or crossroad and tries to deliver data to its left/right direction. Using existing routing algorithms in VANET will cause the big amount of data to stuck at corners or crossroads and to be transmitted back and forth with very few data packets being delivered. It is very time consuming and computation consuming. In this paper, we propose a data delivery algorithm called distribution-based data delivery to handle the above-mentioned corner problem with the help of some vehicular sensors like accelerometer. Evaluations show that the time cost of data delivery at corners is saved for at least 30% by our algorithm comparing to other routing algorithms like VADD in VANET.

1. Introduction

VANET is a hot topic in recent years; it is proposed to enhance driving safety by propagating traffic information among driving vehicles. Although great progress has been made in data routing, VANET still cannot find enough convincing applications. Modern vehicles are equipped with various sensors, the sensors are used to monitor the states of vehicle and surroundings. The information vehicular sensors collected would be very useful for some crowdsourcing applications. For example, fuel tank sensor gives fuel level value; if the fuel level increases at a location, then the vehicle probably is refueling and the location is a fuel station; a vehicle can request the locations of fuel station from its neighbors so that it knows where to refuel without the help of online GIS system. Vehicular sensors enrich the information VANET could propagate; therefore, the combination of wireless sensors and VANET to construct vehicular sensor network (VSN) or vehicular ad hoc and sensor network (VASNET) [1] is a good complement of VANET.

There are many good routing algorithms in VANET can be transplanted to VSN, like GPSR [2] and VADD [3], because VSN has the same network architecture as VANET. However, because vehicular sensors will collect a large amount of data, VSN needs a routing algorithm which can handle this amount of data. In most cases, routing algorithm works at network layer, it simply forwards the packet passed down from upper layers to the next hop according to routing algorithm. The packet is usually very small; therefore, the assumption that a packet can be delivered completely to the next hop in VANET is correct. But, if we look at the data delivery in the perspective of data itself and combined with the characteristics of VSN, we find that existing data delivery algorithms will cause the big amount of data to stuck at corners or crossroads and cause big time and resource waste.

There is no problem in communication of V2V or V2I in a straight road; however, when communication meets corners, especially in city environment, time will be a problem. As Figure 1 shows, there are many obstacles by the road side, like buildings, trees, and crowds, the wireless communication signals will be blocked by these obstacles. If a vehicle has to deliver data to its left direction at crossroad or corner, the time for delivery is d/v, d is the width of road, and v is the velocity of the vehicle driving across the crossroad.

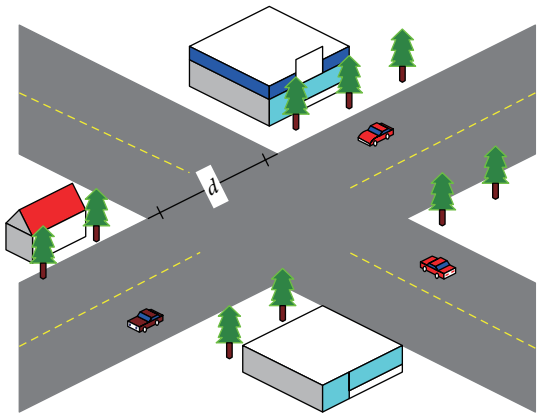


Figure 1: Road side obstacles. There are trees and buildings by the road side, these obstacles will block or at least weaken signal propagation.

A common lane width is 3.7 meters, for a four-lane road, the road width is nearly 15 meters. Suppose v equals 40 km/h, then the time for data delivery is 1.36 seconds. Even we do not consider wireless channel competition and conflict and set the data link bandwidth to 1 MB/s, the vehicle can only transmit 1.36 MB data. For a large bulk of data which is at least 10 MB, it needs eight times for transmitting in an ideal situation, actually it will cost much more using the existing routing algorithm in VANET.

Figure 2 shows how existing routing algorithms, like VADD, handle the corner data delivery. Red block that represents a data block can be completely delivered when a vehicle driving across a corner. Figure 2(a) shows that S1 has three red blocks to deliver to D, but it can only deliver one block to D when it drives across the crossroad; Figure 2(b) shows that S1 recalculats a route to reach D, and it will deliver all the left two blocks to S2, because S2 is heading to the crossroad (the heading direction can be detected by various sensors like accelerometer, orientation) and S2 is closest to the crossroad compared to other vehicles which S1 can communicate with; S2 can only deliver one block to D, as Figure 2(c) shows, S2 recalculats a route to D and passes the left one block to S3; Figure 2(d) shows that S3 delivers the last block to D. To calculate time cost and computation cost, we suppose the time of sending one block is T_{block}, the time of calculating a routing path is T_{route}, the average time of driving to the crossroad is T_{drive}, the computation cost of sending one block is C_{send}, the computation of receiving and storing a block is C_{recv}, and the computation of calculating a route is C_{route}. The time cost of data delivery in Figure 2 is $6T_{\text{block}} + 3T_{\text{drive}} + 2T_{\text{route}}$, computation cost is $6C_{\text{send}} + 3C_{\text{recv}} + 2C_{\text{route}}$, here we ignore the time and computation cost of D.

The idea of solving this problem is simple: we split the data into equal blocks, each block can be completely delivered when a vehicle driving across a crossroad or corner, the blocks are distributed to the neighbors of source vehicle which are also going to drive across the same crossroad or corner, each neighbor will receive one data block. In this case, the data will be completely delivered to destination in a single round. If we use Figure 2 as an example, the time cost of our solution is $3T_{\text{block}} + T_{\text{drive}}$, the computation costs is $5C_{\text{send}} + 2C_{\text{recv}} \cdot T_{\text{drive}}$ is much bigger than the other two time cost, our solution only have one T_{drive} and the existing solutions has three. The improvement is obvious and the algorithm will be discussed in detai l in the next section.

The rest of paper is organized as follows. Section 2 will discuss the details of our solution, Section 3 will give results of evaluations, related works and conclusions will be given in Sections 4 and 5.

2. Distribution-Based Data Delivery at Corners

In this paper, we propose a distribution-based data delivery at corners (DBDDC), it split large amount of data into small blocks and distribute them to several vehicles which are driving to the same corner or crossroad; therefore, it can deliver all the data blocks to the destination successively and need not any recalculation of routing path.

The idea of DBDDC is simple, however, it has some challenges to handle: (1) how to recognize a vehicle driving to the same corner or crossroad as that of source vehicle. (2) how to count the number of qualified vehicles to make sure there are enough vehicles to carry all the blocks.

2.1. Challenges. To know whether a vehicle is driving to the same crossroad as that of source vehicle, we need to know two things: the driving direction and the segment of road the vehicle is at. The direction is easy to be obtained, accelerometer sensor or orientation sensor embedded in vehicles or mobile phones give three-dimensional readings: Azimuth, Pitch, Roll. Take accelerometer for example, Azimuth is the acceleration on the z-axis, Pitch is the acceleration on the x-axis, and Roll is the acceleration, on the y-axis; therefore, we can judge the direction of driving according to the coordinate system of accelerometer. If a vehicle is heading to the same direction as that of source vehicle and they are in the same segment of road, then we can judge that the vehicle must be driving to the same crossroad as that of the source vehicle. As Figure 3 shows, when N1, N2, N3, and S are in the same segment between crossroad 1 and 2, then they are all heading to crossroad 2; in this paper, we call these vehicles distribution nodes. A vehicle can judge whether it is between crossroad 1 and 2 through comparing its GPS location with the locations of crossroads 1 and 2.

To count the number of distribution nodes, the source vehicle will broadcast a beacon message for requesting the distribution nodes to reply their IDs. As Figure 4 shows, the neighbor vehicles in the communication range of source vehicle will all receive a beacon packet, the vehicles will judge whether they are distribution nodes, only the distribution nodes will send replies to source vehicle with their IDs and broadcast the received beacon for those distribution nodes out of communication range of source vehicle, also the distribution nodes will relay the received replies from other distribution nodes to the source vehicle. After a short time,

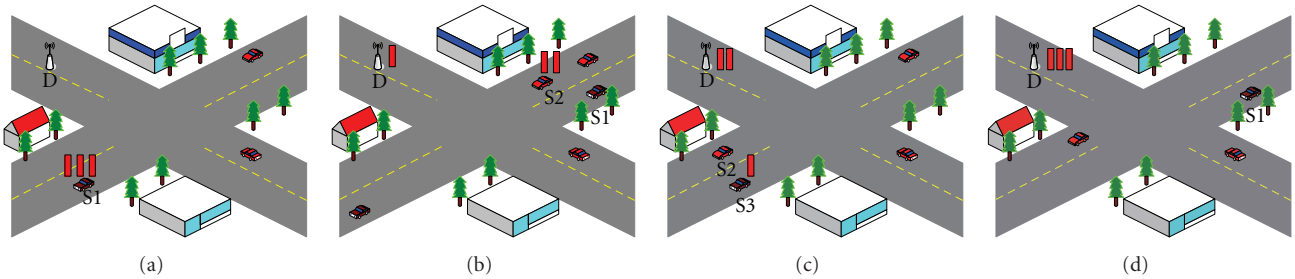


Figure 2: Data delivery at crossroads using existing routing algorithm.

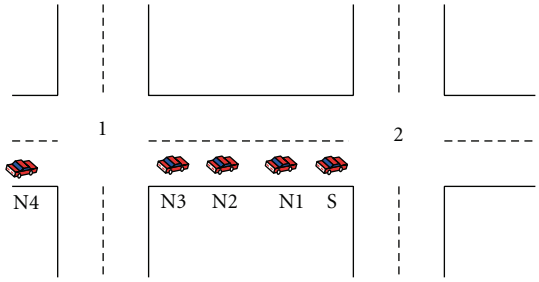


Figure 3: The vehicles which will drive across the same crossroad as source vehicle does.

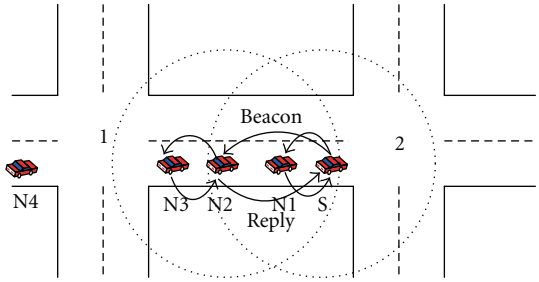


Figure 4: The process of detecting distribution nodes.

the source vehicle will have a list of IDs of distribution nodes and the number of distribution nodes can be calculated.

2.2. Algorithm. Figure 5 shows that there are four cases in DBDDC. Figure 5(a) is the case that before source vehicle S drives across the crossroad, there are enough distribution nodes for carrying all data blocks. Figure 5(b) is the case when there are not enough distribution nodes and source vehicle already turned left or right, so the undelivered data will be delivered in a straight road. Figure 5(c) is the case when there are not enough distribution nodes and source vehicle drives straight and already crossed the crossroad, the undelivered blocks will be sent to the vehicles driving in a reverse direction; in this paper, we call these vehicles reverse distribution nodes. There are enough reverse distribution nodes in this case. Figure 5(d) is the case when there are not enough distribution nodes and reverse distribution nodes, source vehicle has to carry the undelivered blocks and drive to the next crossroad, when source vehicle approaches the crossroad, it will recalculate a route and do the DBDDC again. Suppose the driving direction values are *straight*, *left*, and *right*; the number of distribution nodes is N_d; the number of reverse distribution nodes is N_{rd}; the block size which can be completely delivered when a vehicle driving across a crossroad is *Blk*; the data size is *Data*, then the four cases can be described by

$$\text{Case} = \begin{cases} \text{Case 1}: & \text{direction} = \text{straight and } \dfrac{Data}{Blk} \le N_d, \\ \text{Case 2}: & \text{direction} = \text{left or right} \\ & \text{and } \dfrac{Data}{Blk} > N_d, \\ \text{Case 3}: & \text{direction} = \text{straight and } \dfrac{Data}{Blk} \\ & \le (N_d + N_{rd}), \\ \text{Case 4}: & \text{Otherwise.} \end{cases} \tag{1}$$

Case 1. If source vehicle is approaching a crossroad and data will be delivered at the crossroad to its left/right direction, it will judge whether case 1 is satisfied (see Algorithm 1).

Case 2. If source vehicle already turned left/right, the data will be delivered in a straight road mode which is easy and implemented by existing routing algorithm (see Algorithm 2).

Case 3. If source vehicle already passed the crossroad, it will use Algorithm 3.

Case 4. If source vehicle has not delivered all data to the destination crossroad and is approaching another crossroad, it will recalculate a new route (see Algorithm 4).

Until now, we can give the DBDDC algorithm. DBDDC is not a complete routing and data delivery algorithm, it is a complement of existing routing algorithm, if existing routing algorithms like VADD combines DBDDC, it will achieve better performance at corners and crossroads as Section 3 shows.

3. Evaluation

In this paper, we use simulations to evaluate the performance of DBDDC. We compare the time cost of data delivery

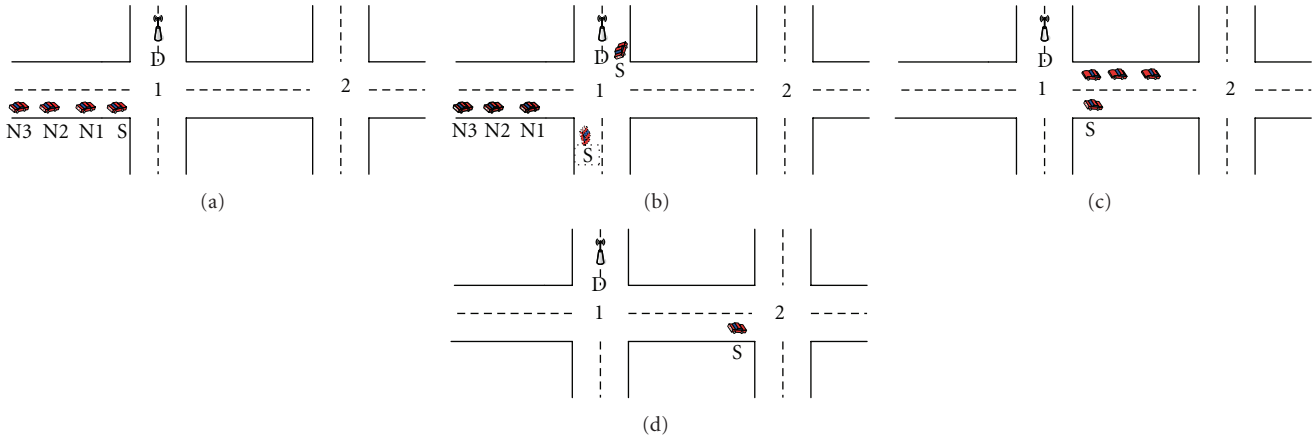


Figure 5: Four cases of DBDDC.

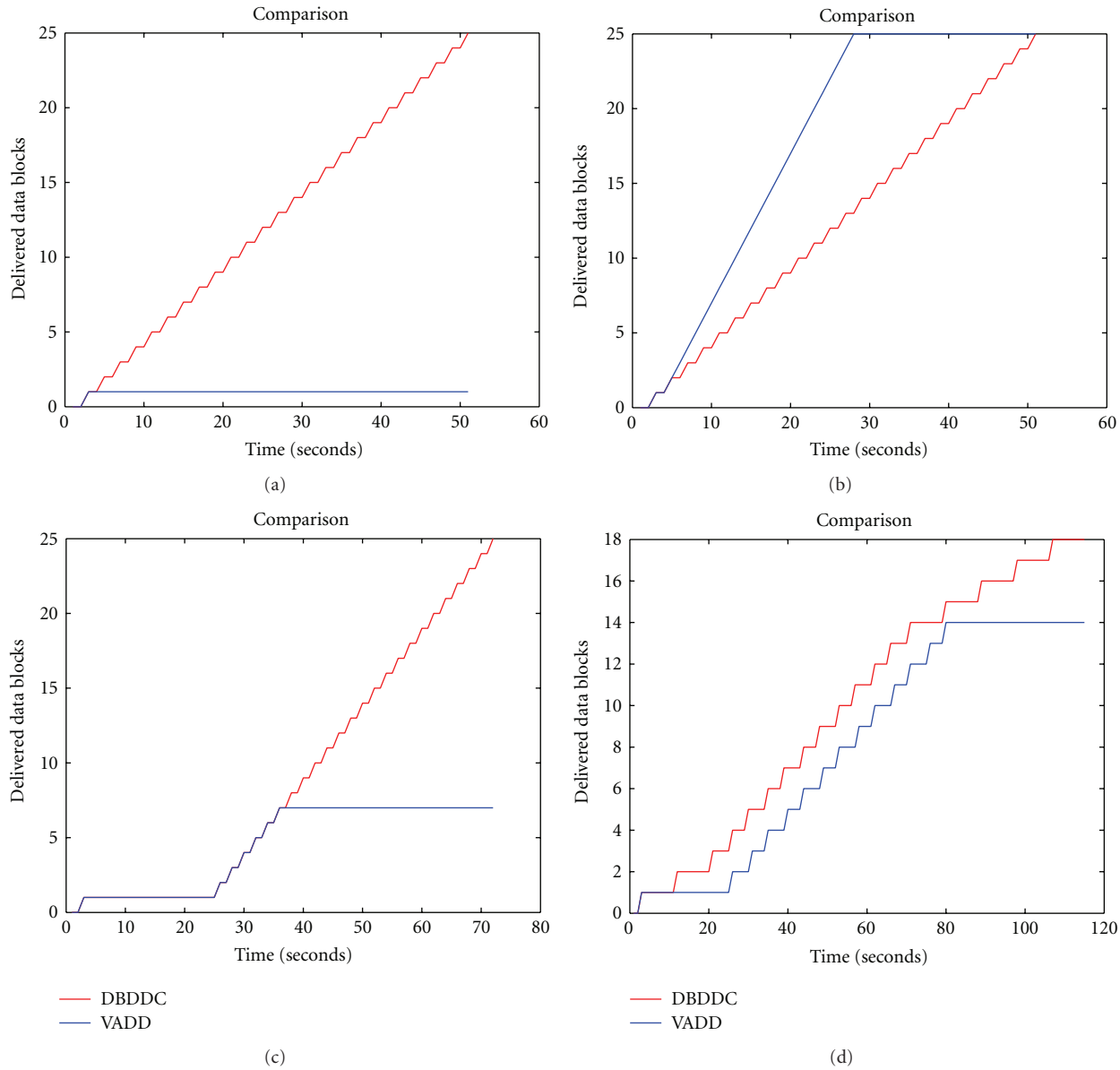


Figure 6: Evaluations of DBDDC and VADD in four cases.

```
Require:
  Data;
Ensure:
  true or false;
  send beacon to detect the number of distribution nodes;
  read reply list and get the number N_d;
  if Data/Blk ≤ N_d then
    split Data into blocks of Blk size;
    send each block to a vehicle in the reply list;
    return true;
  else
    Data1 ← Data(1), Data(2), ..., Data(N_d);   //assign the
    front N_d blocks of data to Data1
    Data ← Data − Data1;
    split Data1 into blocks of Blk size;
    send each block to a vehicle in the reply list;
    return false;
  end if
```

Algorithm 1: Data delivery for Case 1: bool DDC1 (Data).

```
Require:
  Data;
Ensure:
  true or false;
  use existing routing algorithm for Data delivery in straight
  road mode;
  return true;
```

Algorithm 2: Data delivery for Case 2: bool DDC2 (Data).

between DBDDC and VADD at crossroad. Suppose there are two crossroads: crossroad 1 and crossroad 2, each road segment can hold 25 vehicles, the average velocity of vehicle is 40 km/h, the average arrival time at a crossroad of a vehicle is 23 seconds, the time a data block can completely be delivered is 2 seconds, there are 25 data blocks to be delivered on source vehicle, source vehicle is driving from west to east and it will deliver the data to the north direction of crossroad 1, and the communication range is 100 meters.

We compare our solution with VADD in four different cases. Figure 6(a) shows a case when there are enough distribution nodes and no reverse distribution nodes, which is 24 vehicles, source vehicle can distribute 24 data blocks to them, and the 25 blocks will be delivered to destination successively; if using VADD, source vehicle will deliver one block, and it cannot find any vehicle on the reverse direction to carry the left blocks back to crossroad 1.

Figure 6(b) shows a case when there are enough distribution nodes and source vehicle turns to north at crossroad 1, in our solution, source vehicle already distributed all the blocks, therefore the time cost does not change compared to Figure 6(a); if using VADD, source vehicle will deliver all the blocks directly to the destination, the time cost is small; however, in our solution, if there are no distribution nodes,

```
Require:
  Data;
Ensure:
  true or false;
  repeat
    if a new reverse distribution node drive into
    communication range of source vehicle then
      send one block to the reverse distribution node;
      Data ← Data − transmitted block;
      if Data = 0 then
        return true;
      end if
    end if
  until source vehicle is approaching to another crossroad
  return false;
```

Algorithm 3: Data delivery for Case 3: bool DDC3 (Data).

```
Require:
  Data, Crossroad;
Ensure:
  true or false;
  if source vehicle is approaching Crossroad then
    if DDC1(Data) = true then
      return true;
    else
      if source vehicle turned left/right then
        DDC2(Data);
        return true;
      end if
      if DDC3(Data) = true then
        return true;
      else
        recalculate a route;
        return false;
      end if
    end if
  end if
```

Algorithm 4: DBDDC.

the solution will act exactly as VADD does, the time cost is small too.

Figure 6(c) shows a case when there are no distribution nodes and enough reverse distribution nodes, source vehicle is driving straightly. In our solution, source vehicle deliver one block when it is driving across crossroad 1 and pass the left blocks to each reverse distribution nodes, because the average arrival time is 23 seconds, the delay time is nearly 23 seconds; if using VADD, source vehicle will deliver one block during crossroad passing, after 7 seconds, source vehicle will meet the first vehicle V1 on the reverse direction, source vehicle has 9 seconds to deliver data blocks, V1 will receive 4 blocks, and it will deliver 1 block. In a similar way, the second vehicle V2 on the reverse direction will receive 4 blocks and

deliver 1 block, and so on. Finally, 6 vehicles will receive all blocks and 6 blocks will be delivered.

Figure 6(d) shows a case when we set random numbers of distribution nodes and reverse distribution nodes, in this figure there are 8 vehicles for distribution nodes and 9 vehicles for reverse distribution nodes. Source vehicle is driving straightly. In our solution, 9 blocks will be delivered when source vehicle and distribution nodes driving across crossroad 1, 9 left blocks will be delivered to destination by reverse distribution nodes. 7 blocks are left waiting for recalculating route; if using VADD, source vehicle will deliver 4 blocks to V1, V1 will deliver 1 block. If we suppose each vehicle is 11 meters from each other, then V1 will not meet any distribution nodes because they already crossed crossroad 1; therefore, we set each vehicle at 100 meters from each other. V1 will meet the third distribution node D3, it will pass 3 left blocks to D3, D3 will deliver 1 block at crossroad 1, and it will meet the third reverse distribution node V3 and pass the left 1 block to V3. The back and forth process will keep going until no distribution nodes and reverse distribution nodes are driving to crossroad 1.

4. Related Works

VANET was a hot topic and many researches focus on routing algorithms. Table driven is the early version of routing algorithm, it is based on the routing methods in static networks. Each node will discover and maintain a routing table [4–6], because of the high mobility of VANET, discovering and maintaining a routing table become very difficult. A new concept of routing then appeared, it is based on geographic positions. The source node needs to know the location of destination node, every hop of routing, the relay node will choose a next hop which is closest to the destination geographically, it needs not routing tables; therefore, it is suitable in mobile network. Typical routing algorithms are GPSR [2] and GPCR [7]. Vehicles' driving patterns are constrained by road patterns; therefore, routing along road is more predictable and effective, some typical routing algorithms are VADD [3] and CAR [8].

One reason why there are not enough applications in VANET is that vehicle cannot sense the world and therefore cannot generate enough information propagation in VANET. Some research works take advantage of vehicular sensors and sensors embedded in mobile phones to monitor drivers' behaviours [9–14], these applications can enrich the quantity of information propagated in VANET and create a new kind of vehicular network, VSN. VASNET [1] is a kind of VSN which combines wireless sensors and VANET; however, it only considers the applications for highway safety. One interesting application for VSN is MobiEyes [15], it uses vehicular cameras for urban monitoring. Video streams are usually very large in data size; therefore, the solution of the corner problem in this paper has its practical meanings.

5. Conclusion

Corner problem in vehicular sensor network is neglected yet important, it will cause data blocks transmitted back and forth at corners or crossroad and cause big time waste and computation waste. Splitting big data into small blocks and distributing them to other vehicles will help solve the problem, and for most times, the data delivery rate is much higher than existing solutions. However, GPS-based localization may have some inherent defects [16], how to deal with the errors GPS brings in will be in our future works.

References

[1] M. J. Piran, G. R. Murthy, and G. P. Babu, "Vehicular ad hoc and sensor networks; principles and challenges," *International Journal of Ad hoc, Sensor and Ubiquitous Computing*. In press.

[2] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.

[3] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–12, April 2006.

[4] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '99)*, pp. 90–100, February 1999.

[5] G. Pei, M. Gerla, and T. W. Chen, "Fisheye State Routing: a routing scheme for ad hoc wireless networks," in *Proceedings of the IEEE International Conference on Communications (ICC '00)*, pp. 70–74, June 2000.

[6] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds., The Kluwer International Series in Engineering and Computer Science, Springer, 1996.

[7] C. Lochert, M. Mauve, F. Holger, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, pp. 69–72, 2005.

[8] V. Naumov and T. R. Gross, "Connectivity-aware routing (CAR) in vehicular ad hoc networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1919–1927, May 2007.

[9] E. Murphy-Chutorian, A. Doshi, and M. M. Trivedi, "Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation," in *Proceedings of the 10th International IEEE Conference on Intelligent Transportation Systems (ITSC '07)*, pp. 709–714, October 2007.

[10] G. A. T. Holt, M. J. Reinders, and E. A. Hendriks, "Multi-dimensional dynamic time warping for gesture recognition," in *Proceedings of the 13th Annual Conference of the Advanced School for Computing and Imaging*, 2007.

[11] R. Muscillo, S. Conforto, M. Schmid, P. Caselli, and T. D'Alessio, "Classification of motor activities through derivative dynamic time warping applied on accelerometer data," *Proceedings of the 29th Annual International Conference of the IEEE*, vol. 2007, pp. 4930–4933, 2007.

[12] D. A. Johnson and M. M. Trivedi, "Driving style recognition using a smartphone as a sensor platform," in *Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA, 2011.

[13] S. Y. Cheng and M. M. Trivedi, "Turn-intent analysis using body pose for intelligent driver assistance," *IEEE Pervasive Computing*, vol. 5, no. 4, pp. 28–37, 2006.

[14] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan, "Mobile phone based drunk driving detection," in *Proceedings of the 4th International Conference on Pervasive Computing Technologies for Healthcare (Pervasive Health)*, March 2010.

[15] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, "Mobeyes: Smart mobs for urban monitoring with a vehicular sensor network," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 52–57, 2006.

[16] M. J. Piran and G. R. Murthy, "A novel routing algorithm for vehicular sensor networks," *Internation Journal of Wireless Sensor Networks*, vol. 2, no. 12, pp. 919–923, 2010.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 758340, 10 pages
doi:10.1155/2012/758340

Research Article

Distributed Intrusion Detection of Byzantine Attacks in Wireless Networks with Random Linear Network Coding

Jen-Yeu Chen and Yi-Ying Tseng

Department of Electrical Engineering, National Dong Hwa University, No. 1, Sec. 2, Da Hsueh Road, Shoufeng, Hualien 97401, Taiwan

Correspondence should be addressed to Jen-Yeu Chen, jenyeu@ieee.org

Received 23 October 2012; Accepted 27 November 2012

Academic Editor: Haigang Gong

Network coding (NC) can be applied to achieve the maximal information flow in a network. In energy-constraint wireless networks such as wireless sensor networks, applying NC can further reduce the number of wireless transmissions and hence prolong the life time of sensor nodes. Although applying NC in wireless networks is obviously beneficial, it is possible that a malicious node (Byzantine attacker) can take advantage of the inherent vulnerability of error propagation in the NC scheme to corrupt all the information transmissions. In the NC scheme, an intermediate node linearly combines several incoming messages as its outgoing messages. Thus a data error injected in any intermediate nodes will corrupt the information reaching a destination. Recent research efforts have shown that NC can be combined with classical error control codes and cryptography for secure communications or misbehavior detections. Nevertheless, when it comes to Byzantine attacks, these results have limited effects. In this paper, a distributed algorithm is developed to effectively detect, locate, and isolate the Byzantine attackers in a wireless ad hoc network with random linear network coding (RLNC). To the best of our knowledge, our work is the first to address the problem of Byzantine failures in a wireless network with RLNC.

1. Introduction

1.1. Network Coding. Network coding has become a paradigm shift in information transmission, it is first brought up by Ahlswede et al. [1]. Instead of traditional information transmission method, simply storing and forwarding, network coding allows intermediate nodes to mix received information together and transmit new information generated by the received information in terms of encoding. Due to encoding operation at intermediate nodes, data can be regarded as information flowing through a network, which is a sense of data compression. Therefore, throughput and bandwidth efficiency can be increased and delay can be decreased via network coding. In [1], it has showed that network capacity with network coding can be bounded by min-cut max-flow theory, which is larger than traditional storing-and-forwarding method.

1.2. Random Linear Network Coding. Recent research proving throughput gain of network coding in a variety of application makes network coding an attractive topic. With algebraic approaches, such as [2], a communication pattern with network coding of a network can be designed and achieve its promised capacity, which is the min-cut from the source to the sinks in a network graph [1]. However, the requirement of global topology information and the adoption of centralized optimization make algebraic approaches difficult to implement [3]. Therefore, a distributed network coding scheme, named random linear network coding (RLNC) was proposed [4]. RLNC is a powerful tool to disseminate information in networks for it is distributed and robust against dynamic topology. Without knowing global information such as network topology, RLNC regards every encoded packet as a coding vector over a finite field $\mathbb{F}_q$ and generates new packets at intermediate nodes by linearly combining received packets with random coefficient. Some overhead in packet's header is introduced to record how packets are combined (in [4], it is called global encoding vector) and sinks can do decoding and recover original information as long as they retrieve enough packets.

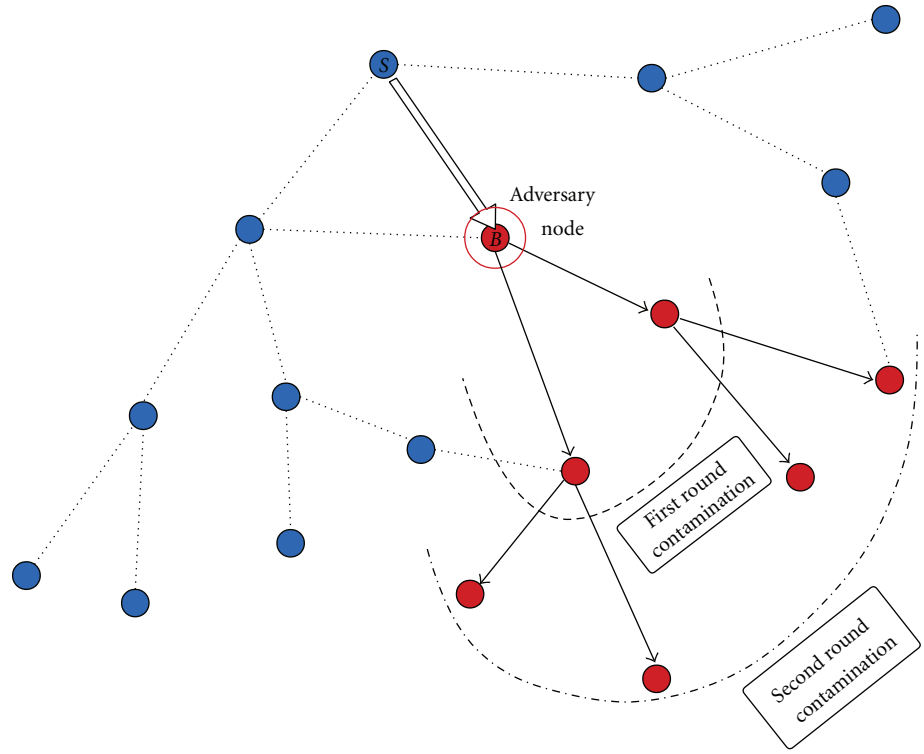


Figure 1: Error propagation due to modifying packets by Byzantine nodes in a network with RLNC.

1.3. Security Issue of Network Coding. Network coding shows its variety of possibilities and benefit in information dissemination; however, it also introduces a new type of security issue. The most serious security challenges posed by network coding thus come from various types of Byzantine attacks, especially packet-modifying attack. In particular, RLNC has been very robust to packet losses induced by node misbehavior [5]. Nevertheless, when it comes to packet-modifying attack, RLNC has become quite vulnerable. In RLNC, one intermediate node will linearly combine received packets and generate new packets to next multiple receivers. If this node has been compromised and generates error packets, other nodes receiving those error packets will also be modified for those error packets will stay in buffer and keep being combined with normal packets. Hence, nodes on a path that these error packets go through would become new compromised nodes without self-awareness and disseminate more error packets. In other word, the error due to modified packets will *propagate* in network with RLNC. Eventually, the whole communication network may be crushed just because of one single adversary node. Figure 1 shows how a single adversary node propagates error.

The paper is organized as follows: Section 2 illustrates pros and cons of related works on Byzantine attacks, Section 3 describes our model and algorithm, Section 4 gives the simulation results and analysis, and Section 5 shows mathematical analysis. Section 6 concludes the paper with a summary of the results and discussion of further work.

2. Related Work

Existing method mostly modifies the format of coded packet against Byzantine attacks and can be divided into two main categories: (1) misbehavior detection and (2) end-to-end error correction.

2.1. Misbehavior Detection. Misbehavior detection applies error control technique or information-theoretic frameworks of encryptography to detect the modification introduced by Byzantine attackers. By types of nodes who take care of coding burden, misbehavior detection can be further divided into *generation-based* and *packet based*. *Generation based* detection takes similar advantage as error-correcting codes and lays expensive computation tasks on destination nodes. As long as enough information is retrieved by destinations, modification can be detected. Reference [6] proposes an information-theoretic approach for detecting Byzantine modification in networks employing RLNC. Each exogenous source packet is augmented with a flexible number of hash symbols that are obtained as a polynomial function of the data symbol. This approach depends only on the adversary not knowing the random coefficient of all other packets received by the sink nodes when designing its adversarial packets. The hash schemes can be used without the need of secret key distribution but the use of block code forces an priori decision on the coding rate. Moreover, the main disadvantage of generation-based detection schemes is that

only nodes with enough packets from a generation are able to detect modifications and thus, result in large end-to-end delays.

On the contrary to generation-based detection schemes, *packet-based* detection schemes allow intermediate nodes in the network detecting modified data on the fly and drop modified packets instead of only relying on destinations, which is more suitable for high attack probability compared to generation-based detection schemes. Packet-based detection schemes require active participation of intermediate nodes with the ability to compute hash function or generate signature based on homomorphic hash functions [7, 8]. Hash of a coded packet can be easily derived from the hashes of previously encoded packets; in that way, intermediate nodes can verify validity of encoded packets before linearly combining them. This characteristic also prevents from error propagating in network. Unfortunately, homomorphic hash function is also computationally expensive and cannot be used in intersession network coding scenario while different sources combine their own source information together.

2.2. End-to-End Error Correction. End-to-end error correction schemes include error-correcting code method into the process of encoding packets and sinks can correct error and recover original information under certain amount of error. Like generation-based detection schemes, end-to-end error correction schemes lay all encoding and decoding tasks on sources and sinks, such that intermediate nodes are not required to change their mode of operation. The transmission mode for end-to-end error correction schemes with network coding can be described by matrix channel $Y = AX + Z$, where X is the matrix whose rows are the source packets, Y corresponds to the matrix whose rows are received packets at sinks, A denotes the transfer matrix, which records linear transformation operated on packets while they traverse the network, also called global encoding vectors, and Z describes the matrix according to the injected error packets after propagating over the network. With error-correcting code, we can recover X from Y. References [9–11] discuss performance of error correction ability while some channel information, such as loss rate or error probability, is known. Reference [12] proposes a simple coding schemes with polynomial complexity for a probabilistic error model of random network coding and provides bounds on capacity. Reference [13] provides a special coding method, which adds a zero vector in the transmitted packet at the source node with an assumption that there is a secret channel between source nodes and sink nodes to inform sinks where the zero vector locates in the transmitted packet. This information cannot be seen by intermediate nodes, and it will be very useful while Byzantine attackers maliciously modify the transmitted packet. As a matter of fact, under some modification level, the more modification occurs, the more likely sinks can recover the original information by using information from observing modified zero vectors. Reference [13] also gives bounds on capacity for two adversarial modes: when Byzantine attackers have limited eavesdropping ability, optimal rate would be C-z; when Byzantine attackers can eavesdrop all links, optimal rate would be down to $C-2z$, where C is the network capacity and z is the number of links controlled by attackers.With special error-correcting code, sinks can be more tolerant with errors, but this scheme also introduces large overhead in packets, which result in tremendous transmission efficiency decreasing.

Even though end-to-end error correcting schemes can recover original information at sinks, it cannot stop error from propagating and introducing large overhead (in worst case, only 1/3 of a packet carries data); misbehavior detection schemes can intercept modified packets on the fly to prevent errors from propagating, but it unfortunately takes expensive computation complexity. We will propose a new type of network coding packet and a distributed algorithm to locate Byzantine attackers and then isolate those nodes. Our algorithm essentially control the error propagation over the network and is not computationally expensive. Detailed introduction is in the next section.

3. Network Model and Byzantine Attackers

3.1. Network Model with RLNC. Consider a wireless network of n nodes with communication range of r randomly distributed in a square area, represented by an undirected graph $G = (V, E)$, with $|V| = n$ nodes. Let $d(i, j)$ denotes the distance from node i to node j. An edge $e_i j \in E$ when $d(i, j) \leq r$. Besides, these n nodes have the ability to access the information of their position. Without loss of generality, we assume the lower left corner of the square area to be the origin and each node knows their coordinate such as $(3, 4)$.

In the communication pattern in which we are interested, each node can perform RLNC to disseminate messages. One source S trying to multicast k messages $\{m_1, \ldots, m_k\}$ to d destinations $\{D_1, \ldots, D_d\}$ transmits those messages as vectors of bits which are of equal length u, represented as elements in the finite field $\mathbb{F}_q$, where $q = 2^u$.The length of the vectors is equal in all transmissions and all links are assumed to be synchronized with a global clock splitting time into slots or rounds which are common to all nodes in the network. In each time slot, nodes with messages in buffer send out new messages on edges to other nodes simultaneously. Let $S_i(t) = \{f_1, \ldots, f_{|S_i(t)|}\}$ be the set of all messages at nodes i at time slot t, and by definition, for $f_l \in S_i(t), 1 \leq l \leq |S_i(t)|, f_l \in \mathbb{F}_q$ and $f_l = \sum_{u=1}^{n} \alpha_{l_u} m_u, \alpha_{l_u} \in \mathbb{F}_q$. When a node i sends out a message, this message is actually a liner combination, called *local encoding*, of the messages stored in node i with payload $g_{i,\text{out}} \in \mathbb{F}_q$, where

$$g_{i,\text{out}} = \sum_{f_l \in S_i(t)} \beta_l f_l, \beta_l \in \mathbb{F}_q; \quad \Pr(\beta_l = \beta) = \frac{1}{q}, \quad \forall \beta \in \mathbb{F}_q. \tag{1}$$

The vector $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_{|S_i(t)|}]$ is called *local encoding vector*, and the message $g_{i,\text{out}}$ can be further written as follows:

$$\begin{aligned} g_{i,\text{out}} &= \sum_{f_l \in S_i(t)} \beta_l f_l = \sum_{f_l \in S_i(t)} \beta_l \sum_{u=1}^{k} \alpha_{l_u} m_u \\ &= \sum_{u=1}^{k} \left(\sum_{l=1}^{|S_i(t)|} \beta_l \alpha_{l_u} \right) m_u = \sum_{u=1}^{k} \gamma_u m_u, \end{aligned} \tag{2}$$

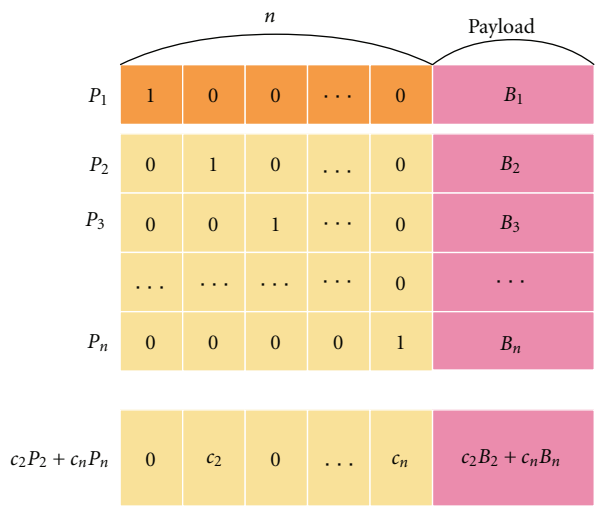


Figure 2: The practical format of transmitted packets.

where $\gamma_u = \sum_{l=1}^{|S_i(t)|} \beta_l \alpha_{l_u} \in \mathbb{F}_q$ and the vector $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_k]$ is called *global encoding vector*. The global encoding vectors are transmitted over the network for decoding, and we define our transmitted packets as Figure 2 to assure that coefficients γ_u are recoded and nodes know that.

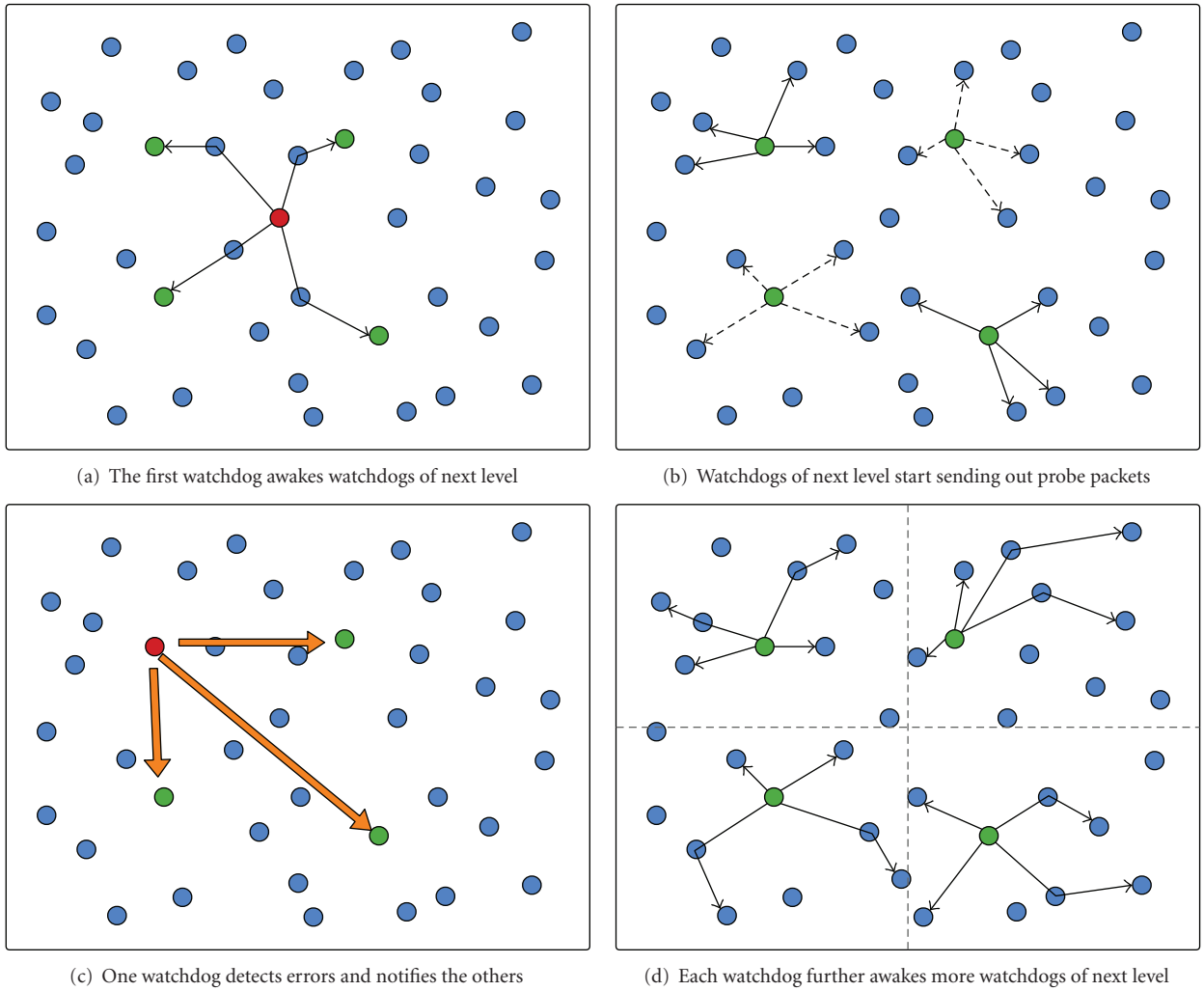
3.2. Threat Model and Our Algorithm. We propose an algorithm, Distributed Hierarchical Adversary Identification and Quarantine, to fight against packet-modifying attack introduced by compromised Byzantine nodes. Assume z_0 out of n nodes has been compromised as Byzantine nodes, and they will modify every packet they send out in order to crash the whole network transmission. Specifically speaking, these Byzantine nodes modify the global encoding coefficients or payload of newly generated outgoing messages, which result in error due to the fact that the modified vectors may not belong to the vector space spanned by source messages and further propagate the errors by following linear combinations of other nodes. We seek an algorithm to locate these Byzantine nodes and isolate them, so that they cannot affect the network.

As mentioned above, network coding is susceptible to the packet-modifying attacks for errors will propagate by operation of linear combinations. However, our algorithm, DHAIQ, uses this characteristic to let error propagate within a certain range in order to let some chosen nodes, referred as *watchdogs*, detect that there are some Byzantine nodes in the monitored area. Before starting our algorithm, we assume that node density and its being known by every nodes from operating other algorithm such as aggregate computation. DHAIQ can mainly be divided into 5 steps.

(1) When a network is under packet-modifying attacks, an arbitrary node in the network will trigger the whole algorithm. This node is the watchdog of the 1st level. This first watchdog will awake the 2nd level's four watchdogs and pass two messages, which are node density and the monitoring area size. The node density is a criterion of termination scheme and the whole deployment area is the 2nd level's monitoring range as Figure 3(a) illustrates. The awaken watchdogs are chosen by locations. These four watchdogs are situated in each corner of their common monitoring area. After awaking the 2nd level's watchdogs, the first watchdog ends its monitoring mode and turns back to its normal mode.

(2) Each of the 2nd level's watchdogs will generate its own special packet, referred as *probe packet*. It then sends this probe packet to the other three watchdogs in an area-restricted flooding way as described in Figure 3(b). Except for these watchdogs, every node that receives these packets will do encoding and then sends new packets to all its neighbors. These packets will be linearly combined via intermediate nodes and constrained to disseminate within the monitoring range. This is all determined at the 2nd level. There are four watchdogs and obviously four different probe packets which are in the same *generation*. The packets belonging to the same generation will start and terminate transmitting simultaneously based on *a time stamp*. Any node that receives the probe packets the first time will record this time stamp. Nodes will continue encoding and sending out packets until the time stamp is expired. If a probe packet reaches a node outside the monitoring range, this node will drop that packet. The information carried by probe packets only traverse in the monitoring range. With the time stamp, all nodes that belong to the same monitoring area can terminate transmitting simultaneously. Before the termination of monitoring, all watchdogs keep retrieving packets from other nodes and keep a packet pool in their buffer. An arriving packet is called *innovative packet* only if it is linear independent to each packets stored in a watchdog's buffer. The discard rule is to keep innovative packets and drop all noninnovative packets. In this way, we also can limit buffer size to a pretty small value. There will be only four packets if there is no adversary node in the monitoring area. Watchdogs also keep computing the rank of vector space spanned by buffered packets until this generation is expired.

(3) If there is any adversary node in the monitoring areas, errors would propagate in the monitoring area and some of the watchdogs would receive modified packets with high probability. Watchdogs can judge whether they receive modified packets by the rank of packet pools. For example, one can say that there is at least an adversary node located in the monitoring area when a watchdog has a packet pool of rank 5. As soon as any of watchdogs detects the existence of adversary nodes, that watchdog will notify the other watchdogs in the same generation and trigger the next level's watchdogs together as shown in Figure 3(c). These four watchdogs will divide their common monitoring rang into four subareas by their corners discussed previously. Each watchdog



(a) The first watchdog awakes watchdogs of next level

(b) Watchdogs of next level start sending out probe packets

(c) One watchdog detects errors and notifies the others

(d) Each watchdog further awakes more watchdogs of next level

Figure 3: Hierarchical division of the monitoring areas.

can then duplicate what the first watchdog does in step (1). Each of them awakes four arbitrary nodes in its corresponding subarea and pass node density and next level's monitoring range, which is a quarter of a current monitoring range according to the location of the upper level's watchdog. The awaken four nodes will also approximately locate at each corner of the subarea and there will be a total of *sixteen* watchdogs awaken for four subareas of the next level (3rd level) as displayed in Figure 3(d).

(4) Repeat step (2) and step (3), keep dividing the areas in a distributed way until we can locate adversary nodes in a small enough area. We define this "small enough area" by the number of nodes locating in it. When the number is small and under a threshold λ, we terminate the monitoring of this area. The number of the node in an area can be estimated by the information of node density and monitoring range, which are carried by probe packets. Therefore, this "small enough area" will be the least monitoring area we can divide. In the least monitoring area, it is very possible that an adversary node is chosen as a watchdog. In this case, adversary nodes may realize this is the time to temporarily act normal and stop modifying the contents of packets. The detection will fail due to adversary nodes' temporary good behaviors. Any detection in progress will be terminated if its monitoring range is under the threshold and all the nodes in this area will be marked as suspect nodes.

(5) After some random time intervals, another arbitrary node will trigger the algorithm again, and this time its monitoring range will be shifted by a short distance. In the very end of the algorithm, we will mark some small squares, which contain adversary nodes. If we shift the monitoring range a little in the beginning of the algorithm, the squares we choose will not be identically overlapped but partially overlapped. This partially overlapped area may contain adversary nodes with high probability and the other nonoverlapped areas, which may contain normal nodes but remarked as suspect, would be less suspicious. In this way, we can eliminate the number of nodes who are marked

as suspects but in fact are normal nodes, referred as *innocent nodes*. To get the final result, each node in the network maintains a suspect table. Whenever a node is reported as a suspect, its suspect level in the other nodes' tables increases by 1. The nodes with high suspect level will be regarded as adversary nodes and isolated. Our simulation results show this shift scheme can greatly reduce the amount of mistaken nodes.

4. Analysis and Simulation Result

4.1. Probe Packets and Time Stamp. In most scenarios of RLNC application, the destinations do the decoding as long as they receive full rank of packets. In our algorithm, we modify this scheme by saying that destinations do not decode to fit our requirements. Considering the worst case, to detect an adversary node is that all watchdogs gather around the center of the monitoring area and the adversary node is located at the very edge. Based on the flooding method, the least time slot required for watchdogs to receive modified packets is the hop number of the shortest path from the adversary nodes to the watchdogs, which is half diagonal of the monitoring area. Since the source of modified packets also come from watchdogs, the average number of hop for a modified packet to arrive the watchdogs is $\sqrt{2k}$. Note that k is the node number of current monitoring area, which is accessible information for watchdogs. We can set time stamps of each generation with this number $\sqrt{2k}$ to assure that watchdogs can receive modified packets and trigger the next level whenever there are Byzantine nodes. When a time stamp is expired, its corresponding nodes will terminate disseminating packets and empty their buffer.

4.2. Range of Shifting. Simply repeating the algorithm will not perform better since the sub-areas are equally divided. If the algorithm starts with the same monitoring area, it will eventually lead to the same result and be in vain. Thus we shift the starting monitoring area in order to minimize the number of innocent nodes. Now the question is how many we should shift each time. It is straightforward to see that if we shift more than a single least monitoring area, this shift is useless. Hence we know the shift range should be no larger than the length of edge of the least monitoring area.

The purpose that we use shift scheme is to further divide the least monitoring area into smaller areas so that we can eliminate the number of innocent nodes. To this end, we shift in both horizontal and vertical directions to let overlapped areas divide the least monitoring area into *four* smaller areas. Hence the question has become how to divide these four smaller areas in order to get the least innocent nodes. Basically we have two options here, equal division and nonequal division. In fact, the equal division method will have the least expected value of innocent nodes. The mathematical analysis is in Section 5, and the simulation results also support our idea.

4.3. Innocent Nodes and Overhead. When we mark the nodes in the least monitoring area as suspect nodes, we mark all the nodes in the area. In fact, some nodes are normal nodes but marked as suspects, and we call them *innocent nodes*. Consider the case in which we only perform identification algorithm once without using suspect table. It is straightforward that uniform distribution of Byzantine nodes can lead to the worst result with the most innocent nodes. The ratio of innocent nodes is upper bounded by $(\mu - 1)z_0/n$, and this bound grows linearly with respect to the number of Byzantine nodes and μ, which is quite a large number. Besides, probe packets carry no data information and the amount of probe packets transmitted of all generations in each level is $O(n\sqrt{n})$. In one identification algorithm, it will trigger $O(\log n)$ levels totally and therefore total number of transmitted probe packets is $O(n\sqrt{n}\log n)$ in time $O(\sqrt{n})$.

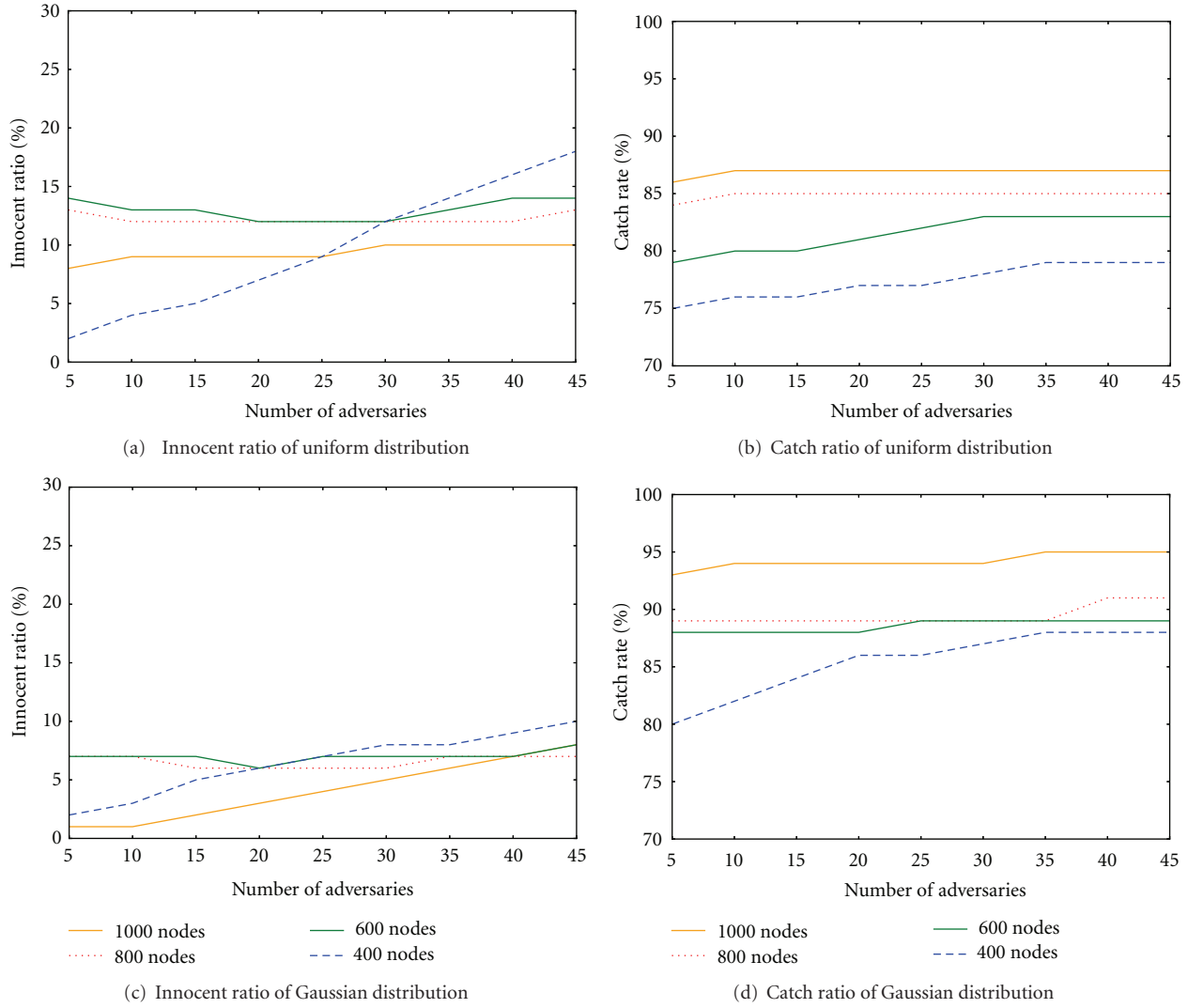
4.4. Simulation Results. In our simulation, we uniformly distribute 400, 600, 800, and 1000 nodes in a square area with width of 800 and node communication range is 50. We simulate our algorithm under the circumstance of the amount of adversary nodes varying from 5 to 45, and these adversaries are uniformly and normally distributed. Figure 4 is the first result of our algorithm, we can see that the innocent ratio of uniform distribution pattern is quite high. The uniform distribution pattern is the worst case to our algorithm. In order to decrease the amount of innocent nodes, we introduce shift scheme. The results are shown in Figure 5. The results with more nodes are in Figure 6. As we can see, our algorithm performs better in a dense topology. Performing shift scheme in our algorithm can eliminate innocent ratio effectively, but it also drags down the catch ratio a little bit, because shift scheme also generates holes around boundaries, which cannot be detected sometimes. The result shows that the catch ratio only drops a little, which is an acceptable value.

5. Analysis

The shift scheme aims to further divide the least monitoring areas into smaller areas so that we can decrease the number of innocent nodes. With it, the final results of marked areas in each run of algorithm will be different. The overlapped marked areas are smaller than the least monitoring areas and contain less innocent nodes. Consider the case that overlapped areas divide a least monitoring area A into four smaller areas, A_1, A_2, A_3 and A_4. The expectation number of innocent nodes will reach a minimum value, while $A_1 = A_2 = A_3 = A_4$. We now prove our claim.

Claim. The expectation value of number of innocent nodes will reach a minimum when the least monitoring area A is divided into four equal areas.

Proof. Assume that the area A is of size 1 and divided into four areas, A_1, A_2, A_3, and A_4, with the area size of a_1, a_2, a_3, and a_4. We have $a_1 + a_2 + a_3 + a_4 = 1$ and $a_1, a_2, a_3, a_4 > 0$. The least monitoring area A has μ nodes totally and k of the μ nodes are adversary nodes. Clearly $k < \mu$. The expectation number of innocent nodes is



(a) Innocent ratio of uniform distribution

(b) Catch ratio of uniform distribution

(c) Innocent ratio of Gaussian distribution

(d) Catch ratio of Gaussian distribution

Figure 4: Innocent ratio and Byzantine catch ratio for two different distribution pattern of adversaries.

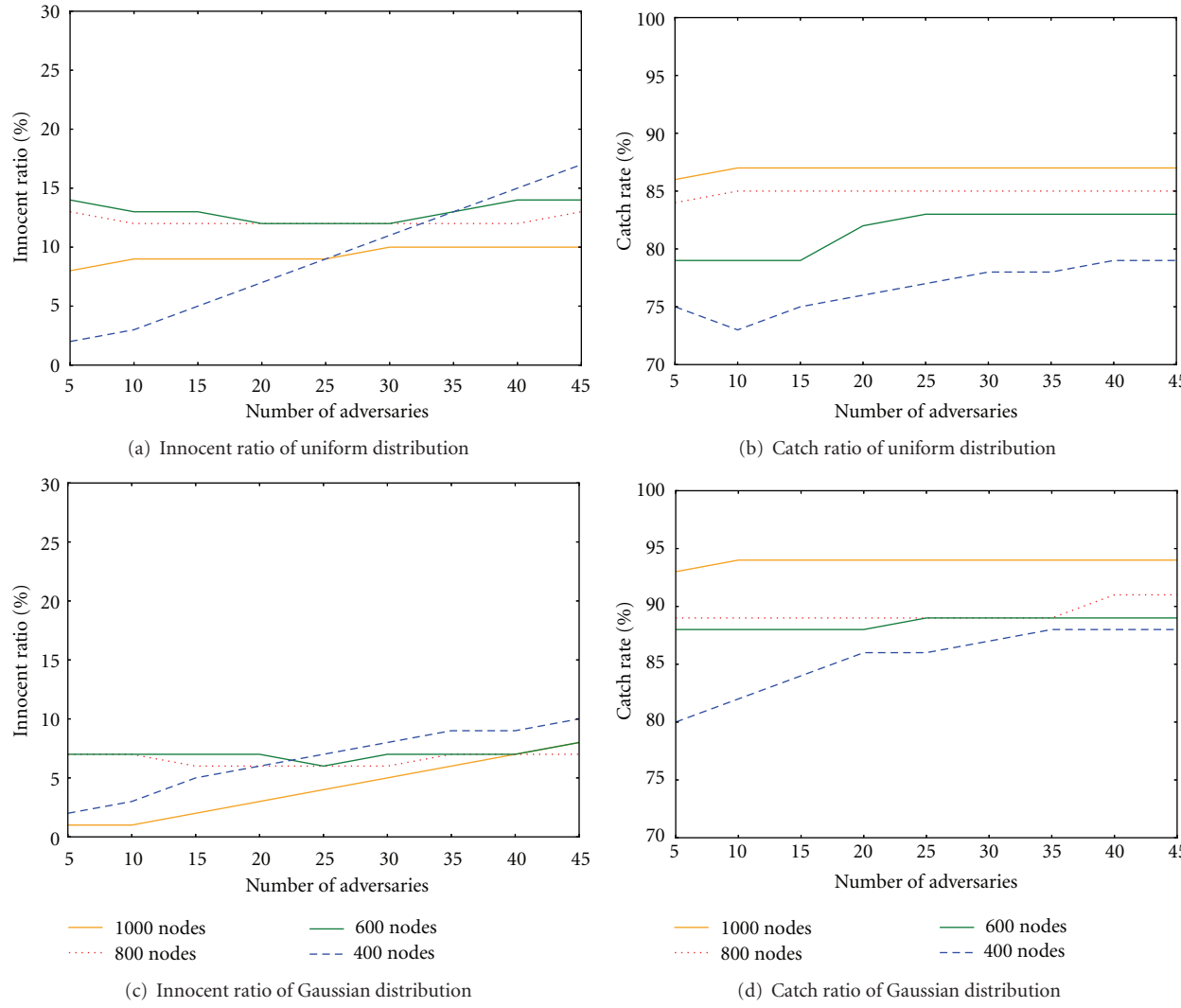
$$
\begin{aligned}
E(k) &= \left[1-(1-a_1)^k\right]a_1\mu + \left[1-(1-a_2)^k\right]a_2\mu \\
&\quad + \left[1-(1-a_3)^k\right]a_3\mu + \left[1-(1-a_4)^k\right]a_4\mu \\
&= (a_1+a_2+a_3+a_4)\mu \\
&\quad - \left[a_1(1-a_1)^k + a_2(1-a_2)^k \right. \\
&\quad \left. + a_3(1-a_3)^k + a_4(1-a_4)^k\right]\mu \\
&= \mu - \left[a_1(1-a_1)^k + a_2(1-a_2)^k \right. \\
&\quad \left. + a_3(1-a_3)^k + a_4(1-a_4)^k\right]\mu.
\end{aligned} \tag{3}
$$

We want to have $E(k) \geq$ some constant c, so the problem becomes

$$
\begin{aligned}
\text{maximize} \quad & x_1(1-x_1)^k + x_2(1-x_2)^k \\
& + x_3(1-x_3)^k + x_4(1-x_4)^k \\
\text{subject to} \quad & x_1 + x_2 + x_3 + x_4 = 1.
\end{aligned} \tag{4}
$$

We denote $f(\mathbf{x}) = x_1(1-x_1)^k + x_2(1-x_2)^k + x_3(1-x_3)^k + x_4(1-x_4)^k$ and $h(\mathbf{x}) = x_1 + x_2 + x_3 + x_4 = 1$. By the Lagrange condition, we have

$$
\begin{aligned}
(1-x_1)^k - kx_1(1-x_1)^{k-1} + \lambda &= 0, \\
(1-x_2)^k - kx_1(1-x_2)^{k-1} + \lambda &= 0, \\
(1-x_3)^k - kx_1(1-x_3)^{k-1} + \lambda &= 0, \\
(1-x_4)^k - kx_1(1-x_4)^{k-1} + \lambda &= 0, \\
x_1 + x_2 + x_3 + x_4 &= 1.
\end{aligned} \tag{5}
$$



(a) Innocent ratio of uniform distribution

(b) Catch ratio of uniform distribution

(c) Innocent ratio of Gaussian distribution

(d) Catch ratio of Gaussian distribution

Figure 5: Innocent ratio and Byzantine catch ratio with shift scheme.

Obviously,the solution to theses equations is

$$x_1 = x_2 = x_3 = x_4 = \frac{1}{4}, \qquad \lambda = \left(\frac{k}{4} - \frac{3}{4}\right)\left(\frac{3}{4}\right)^{k-1}. \quad (6)$$

Thus $\mathbf{x}^* = [1/4, 1/4, 1/4, 1/4]^\top$.

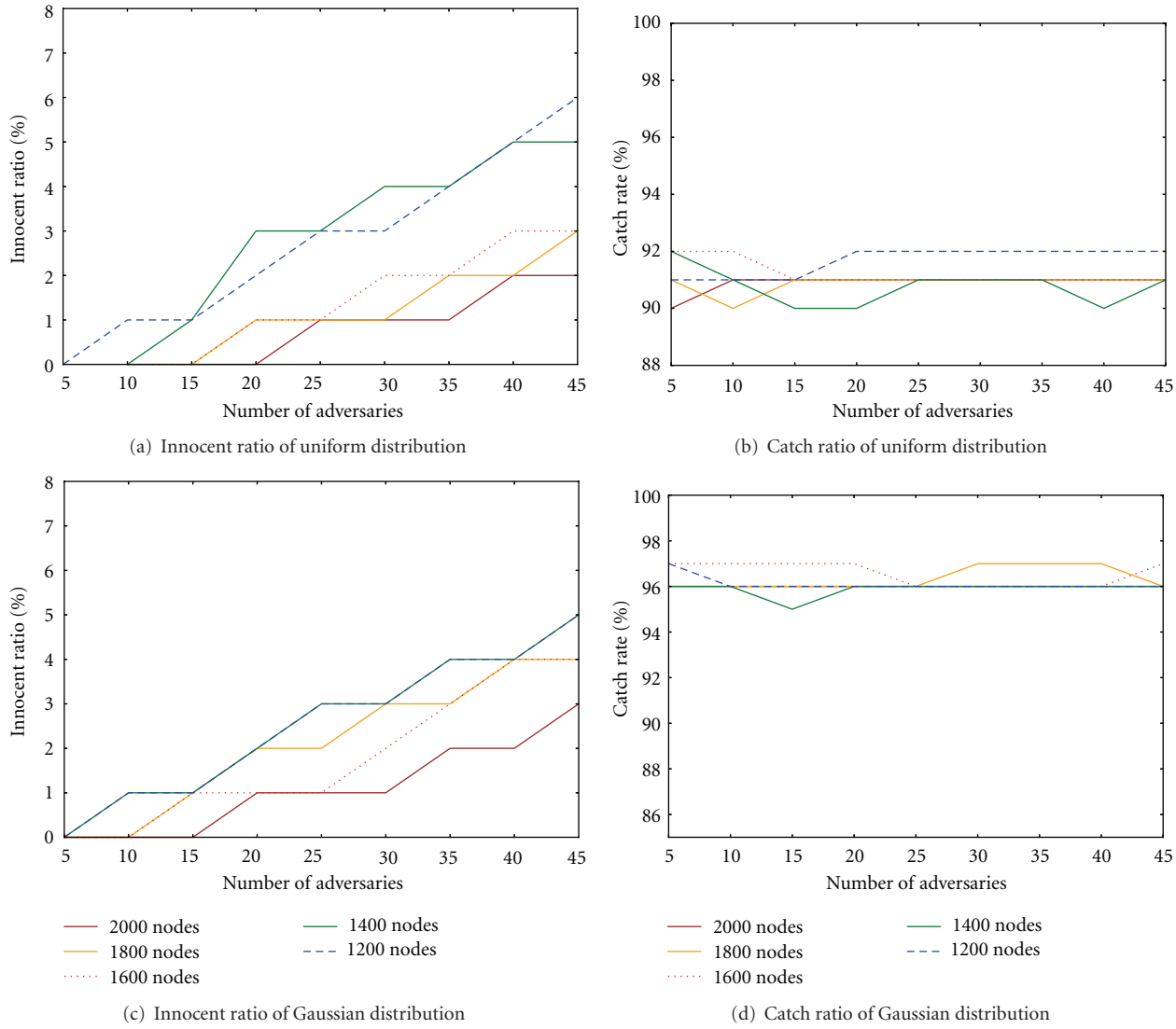
Now we need to resort to the second-order sufficient conditions to determine if the problem reaches a maximum or minimum at $x_1 = x_2 = x_3 = x_4 = 1/4$. Let $l(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda^\top h(\mathbf{x})$ and $\mathbf{L}(\mathbf{x}, \lambda)$ is the Hessian matrix of $l(\mathbf{x}, \lambda)$. We can find the matrix

$$\mathbf{L}(\mathbf{x}^*, \lambda) = \mathbf{F}(\mathbf{x}^*) + \lambda \mathbf{H}(\mathbf{x}^*) = \begin{bmatrix} \mathbf{g}(k) & 0 & 0 & 0 \\ 0 & \mathbf{g}(k) & 0 & 0 \\ 0 & 0 & \mathbf{g}(k) & 0 \\ 0 & 0 & 0 & \mathbf{g}(k) \end{bmatrix}, \quad (7)$$

where $\mathbf{g}(k) = (3/4)^{k-2}((k-7)/4)$. On the tangent space $M = \{\mathbf{y} | y_1 + y_2 + y_3 + y_4 = 0\}$, we note that

$$\mathbf{y}^\top \mathbf{L}\mathbf{y} = y_1^2\left(\frac{3}{4}\right)^{k-2}\left(\frac{k-7}{4}\right) + y_2^2\left(\frac{3}{4}\right)^{k-2}\left(\frac{k-7}{4}\right) + y_3^2\left(\frac{3}{4}\right)^{k-2}\left(\frac{k-7}{4}\right) + y_4^2\left(\frac{3}{4}\right)^{k-2}\left(\frac{k-7}{4}\right) < 0, \quad \text{for } k < 7 \text{ and all } y \neq 0. \quad (8)$$

Thus L is negative definite on M when $k < 7$ and f reaches a maximum. In our algorithm, we set our $\mu = 5$, and $k < \mu$ obviously. Therefore, we can always reach a minimum expectation value in our setup and it happens at $a_1 = a_2 = a_3 = a_4 = 1/4$. □



(a) Innocent ratio of uniform distribution

(b) Catch ratio of uniform distribution

(c) Innocent ratio of Gaussian distribution

(d) Catch ratio of Gaussian distribution

Figure 6: Results for more nodes.

6. Conclusions and Further Work

We have proposed a locating algorithm in appliance of RLNC to locate compromised Byzantine nodes in a network. Our algorithm can locate the areas where adversary nodes locate with some normal nodes being mistaken as adversary nodes. To reduce the number of mistaken nodes, we use a shift scheme to eliminate the probability of being mistaken. The simulation results show that our algorithm performs well in Guassian distribution pattern for adversary nodes. In the worst case, uniform distribution pattern for adversary nodes, we still can locate most adversary nodes and reduce almost 10% of mistaken ratio by shift scheme. We also give discussion about the best policy for shift scheme. Fixing the shift range to the half length of the least monitoring area has the best performance.

Even though we do locate the areas where adversary nodes lie, but there still exist mistaken nodes. A second stage algorithm is required in order to precisely identify each adversary node. Sampling each node one by one in the most suspicious area or combining some special coding scheme with our algorithm may be a worthy researching direction.

References

[1] R. Ahlswede, N. Cai, S. yen Robert Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] R. Koetter, M. Mdard, and S. Member, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 782–795, 2003.

[3] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2745–2759, 2005.

[4] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT' 03)*, July 2003.

[5] D. S. Lun, M. Mdard, and M. Effros, "On coding for reliable communication over packet networks," in *Proceedings of the 42nd Annual Allerton Conference on Communication, Control, and Computing*, 2004.

[6] T. Ho, B. Leong, R. Koetter, M. Médard, M. Effros, and D. R. Karger, "Byzantine modification detection in multicast networks using randomized network coding," in *Proceedings of the IEEE International Symposium on Information Theory*, p. 143, July 2004.

[7] D. Charles, K. Jain, and K. Lauter, "Signatures for network coding," in *Proceedings of the 40th Annual Conference on Information Sciences and Systems*, vol. 1, pp. 3–14, March 2009.

[8] M. N. Krohn, M. J. Freedman, and D. Mazières, "On-the-fly verification of rateless erasure codes for efficient content distribution," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 226–239, May 2004.

[9] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '07)*, pp. 791–795, June 2007.

[10] D. Silva, F. R. Kschischang, and R. Köetter, "A rank-metric approach to error control in random network coding," *IEEE Transactions on Information Theory*, vol. 54, no. 9, pp. 3951–3967, 2008.

[11] D. Silva and F. R. Kschischang, "Using rank-metric codes for error correction in random network coding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '07)*, pp. 796–800, June 2007.

[12] D. Silva, F. R. Kschischang, and R. Kötter, "Capacity of random network coding under a probabilistic error model," in *Proceedings of the 24th Biennial Symposium on Communications (BSC '08)*, pp. 9–12, June 2008.

[13] S. Jaggi, M. Langberg, S. Katti et al., "Resilient network coding in the presence of byzantine adversaries," *IEEE Transactions on Information Theory*, vol. 54, no. 6, pp. 2596–2603, 2008.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 451504, 8 pages
doi:10.1155/2012/451504

Research Article

Public-Transportation-Assisted Data Delivery Scheme in Vehicular Delay Tolerant Networks

Yong Feng,[1] Ke Liu,[2] Qian Qian,[1] Feng Wang,[1] and Xiaodong Fu[1]

[1] *Yunnan Key Laboratory of Computer Technology Applications, Kunming University of Science and Technology, Kunming, Yunnan 650500, China*
[2] *School of Foreign Language, Southwest University for Nationalities, Chengdu, Sichuan 610041, China*

Correspondence should be addressed to Yong Feng, fybraver@163.com

Received 10 October 2012; Accepted 12 November 2012

Academic Editor: Haigang Gong

As an important component of transportation system, public transportation accounts for a considerable proportion in the whole traffic flow. The public transportation vehicles can be categorized into two categories as follows: one with determinate trajectories and schedules such as bus, tramway, and light rail; the other with flexible and variable running paths, such as taxis. In this paper, we firstly present a destination-gathering-based driving path prediction method for taxis, which can make taxis' driving paths prescient in the initial stage of carrying passengers every time. Compared with ordinary vehicles, public transportation vehicles have such features as long time running on roads and no privacy-protection need, and thus their trajectories can be opened. Through utilizing the features above, we propose a novel public-transportation-assisted data delivery scheme (PTDD) used to improve the performance of data delivery of Vehicular Delay Tolerant Networks (VDTNs). Simulation results based on a real map demonstrate the effectiveness of the proposed scheme.

1. Introduction

With the rapid development of wireless communication technology, new types of wireless networks and applications are appearing constantly. In this context, vehicular networks have gradually become an important research field in wireless communication and received broad attention from both industry and academy [1, 2]. As vehicle nodes' high speed mobility and uneven distribution, vehicular networks have dynamic and changing network topology, which make it difficult to maintain persistent connection among vehicle nodes. To improve data delivery performance, vehicular networks widely adopt Delay Tolerant Networking (DTN) technology, and thus Vehicular Delay Tolerant Networks (VDTNs) emerges. VDTNs have evolved from DTNs and are formed by cars and any supporting fixed nodes.

As an important component of Intelligent Transportation Systems (ITS) [3], VDTNs promise a wide range of valuable applications including real time traffic estimation for trip planning, mobile access to Internet, and in-time dissemination of emergency information such as accidents and pavement collapses. To realize the applications above, one of key research topics is to design effective and efficient data delivery schemes. Therefore, many schemes have been presented to solve the problem in recent years. Among the existing schemes, some works mainly take advantage of geographic position information, such as GPSR [4] and CAR [5]. The performances of these protocols mainly depend on the network connectivity, and they are sensitive to the vehicle node density. Many works are based on the traffic statistics and network layout, such as SADV [6] and VADD [7]. A few protocols such as TBD [8], TSF [9], and STDFS [10] are designed to utilize available vehicle trajectories to improve the data delivery performance. To disseminate vehicle trajectory information, there protocols assume that numerous wireless access points (APs) need to be deployed along roads. That will undoubtedly request a large amount of investment. Some researchers have made explorations on the prediction of vehicle driving paths and make use of the character of anticipating vehicle routes to develop data delivery schemes. In literature [11–14], several prediction model and scheme are proposed. However, these works are based on either

historic driving records or local and current vehicle running status, and thus there are prediction accurate problems.

Public transportation is an important component of transportation system and accounts for considerable proportion in the whole traffic flow. For Beijing, public transportation occupies about 31.5% shares of the traffic flow [15, 16]. In general, the vehicles can be categorized to two types: (i) bus, tramway, and light rail, which have stable trajectories and schedules; (ii) taxi, which has flexible and variable running paths. In this paper, we firstly suggest a driving path prediction method based destination gathering for taxis, which can make taxis' driving paths prescient in the initial stage of carrying passengers every time. Comparing with general vehicles, public transportation vehicles have such features as long time running on roads, no privacy-protection need, and thus their trajectories can been opened for the public. Based on the character, we propose a novel public-transportation-assisted data delivery scheme (PTDD), which can increase data delivery ratio through (i) carrying messages by vehicles whose driving paths will pass through the messages' destination; (ii) increasing chance to find the vehicles that move toward the optimal expected road in intersection areas. The major contributions of this work may be listed as follows.

(a) A new method based on destination information gathering is proposed to predict the driving paths of taxis. The method takes the effect of destination information on prediction process into account, and thus the prediction accuracy and reliability is improved.

(b) We propose a novel data delivery scheme called PTDD to improve the data forwarding performance of VDTNs, which effectively takes advantage of the characteristics of public transportation traffic such as foregone driving path, long time running on roads, no privacy-protection need. Through extensive simulations, the effectiveness of our proposed PTDD scheme is evaluated.

The rest of the paper is organized as follows: Section 2 summarizes the related works for vehicular networking. In Section 3, we present a driving path prediction method based on destination gathering for taxis. In Section 4, we describe the design of our proposed PTDD scheme in detail. Section 5 shows the effectiveness of PTDD via simulation experiments. Section 6 concludes the paper.

2. Related Works

In recent years, data delivery and forwarding issues about vehicle-to-vehicle and vehicle-to-infrastructure in VANET have gained lots of attentions [1, 2, 5–8]. For the frequent network partition and mergence due to the high mobility of vehicles, the physically constrained nodal mobility resulted from the fixed roadways and the constrained nodal moving speed limited by the roadway conditions, the data forwarding in VANET is different from that in the traditional mobile adhoc networks (MANETs). These unique characteristics of the road networks make the MANET routing protocols ineffective in the VANET settings [17]. Thus, many routing protocols based on carry-and-forward thinking have being proposed in order to reach efficient and effective data forward performance in VDTNs.

Among these works, Epidemic [18] is an early approach to deal with the data forward issue in frequent network partition and mergence settings. It allows the random pair wise exchange of data packets among mobile nodes in order to maximize the possibility that data packets can be delivered to their destination node. Thus, a great number of copies of data packets are generated during the delivery, which weakens its performance to a great extent, especially when the resources of bandwidth and buffer are limited.

Some works mainly take advantage of geographic position information, such as GPSR [4], CAR [5], MMR [17], and VVR [19]. Similar to GPSR, both MMR and VVR use greedy forwarding strategy to find the next packet carrier based on the geographical proximity toward the packet destination. Through using the approach of "guard node," CAR forwards data packets through the connected path from the packet source to the packet destination. The performances of these protocols mainly depend on the network connectivity, and they are sensitive to the vehicle node density. Thus the geographic position based routing schemes cannot work well when the vehicular traffic is sparse and none-uniform-distribution.

Some works are based on the traffic statistics and network layout, such as SADV [6], VADD [7], and DBR [3]. SADV routing leverages on the stationary nodes to improve the network connectivity and the data forward performance. Through using a stochastic model based on vehicular traffic statistics, VADD tries to achieve as high delivery successful ratio as possible with low delivery delay from mobile vehicles to stationary packet destinations. DBR scheme focus on satisfying the user-defined delay bound rather than the lowest delivery delay, so that it can economize resource such as channel utilization and buffer space.

A few protocols such as TBD [8], TSF [9], and STDFS [10] are designed to utilize available vehicle trajectories to improve the data delivery performance. TBD utilizes the vehicle trajectory information along with vehicular traffic statistics in order to compute the accurate expected delivery delay for better forwarding decision making. However, to disseminate vehicle trajectory information, these protocols assume that numerous wireless access points (APs) need to be deployed along roads. That will undoubtedly request a large amount of investment.

Some researchers have made explorations on the prediction of vehicle driving paths, and take use of the character of anticipating vehicle routes to develop data delivery schemes. Several prediction model and scheme are proposed such as PBR [11], MOPR [12], PLR [13]. PBR exploits the location and velocity information of vehicles to predict route lifetime, and takes preemptive action to minimize route failure. MOPR improves the routing process by selecting the most stable route in terms of lifetime. PLR uses a location predictor to solve the problem of location inaccuracy and vehicle mobility. In [14], Jeung et al. propose a network

mobility model to predict the driving paths of vehicles. However, these works are based on either historic driving records or local and current vehicle running status, and thus there are prediction accurate problems. Moreover, the approaches mainly focus on short-term and short-distance prediction, and do not take into account how long-distance driving path prediction information is used to improve the data forwarding in VDTNs.

3. Driving Path Prediction for Taxis

In this section, we will firstly introduce the driving path prediction method for taxis based on destination information gathering. For our proposed driving path prediction method for taxis is a new thinking in the research of VDTNs, and thus we will discuss the feasibility and practicability of the prediction method.

3.1. Destination Information Gathering for Taxis. When a passenger takes on a taxi, the first sentence from the driver is always: "Hi, where are you going?" Naturally, the driving destination of the taxi can be caught from the answer of the passenger through phonetic recognition devices. This may be the most convenient way. Of course, other candidate input approaches include handwriting board, keyboard, touch panel, and so on.

3.2. Driving Path Prediction Method. According to taxi operation rule, the main approach of increasing revenues is to increase the carrying passenger times per day. So taxi drivers will always choose the paths with the shortest driving time to take passengers to their destinations as soon as possible. Based on the gathered destination information, GPS device, electronic map, and traffic statistic information at different times, a practicable prediction method is to utilize Dijkstra algorithm to look for the lowest cost path from the current position to the destination, where the cost means the average travel time for each road segment.

To improve prediction accuracy further, we will take advantage of the features that taxi drivers are relatively fixed and familiar with the road and traffic conditions. Through recording the drives' historical route and personal preferences at different times on each taxi, the prediction path can be adjusted and more consistent with the real situation.

3.3. Feasibility and Practicability. As to the approach of destination information gathering for taxis, there is no technology and cost problem because whether phonetic recognition or handwriting board, keyboard, touch panel are all very mature and reliable technology. The difficulty is to create an incentive measure that can prompt taxis drivers to carry out the information gathering activities. To solve the problem, the corresponding accounting mechanism is necessary, and senders or receivers should pay a fee to drivers for messages' success delivery.

When vehicle routes can be accurately predict and broadcast to neighbor vehicles in hello messages, that will bring benefits as follows: (a) increasing data delivery success ratio through carrying messages to their destinations by vehicles whose driving paths will pass the messages' destination; (b) increasing success ratio to find the vehicles that move toward the optimal expected road in intersection areas; (c) reducing wireless channel resources occupancy, and thus lessening wireless collision possibility and improve the quality of channel, which in turn can improve the delivery success rate.

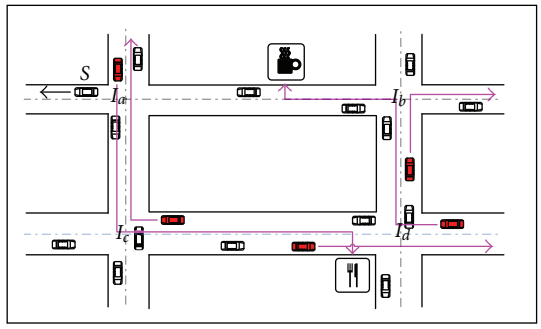


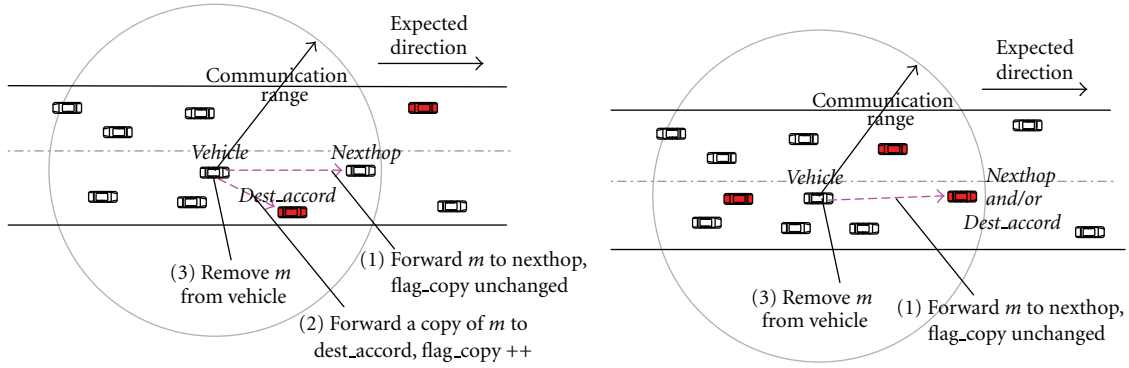
Figure 1: A sketch map of PTDD thinking.

4. Public-Transportation-Assisted Data Deliver Scheme

Through utilizing the proposed destination gathering based driving path prediction method in Section 2 each taxi can know its driving path in advance. Therefore, all public transportation vehicles, whether bus, tramway, light rail, or taxi, have such similar character, that is, their driving trajectories are of foreknowledge. Based on this character, we propose a new data delivery scheme called PTDD to improve the data delivery performance in VDTNs. In Figure 1, the sketch map of PTDD thinking is given out. In the rest of this section, our proposed PTDD will be described in detail.

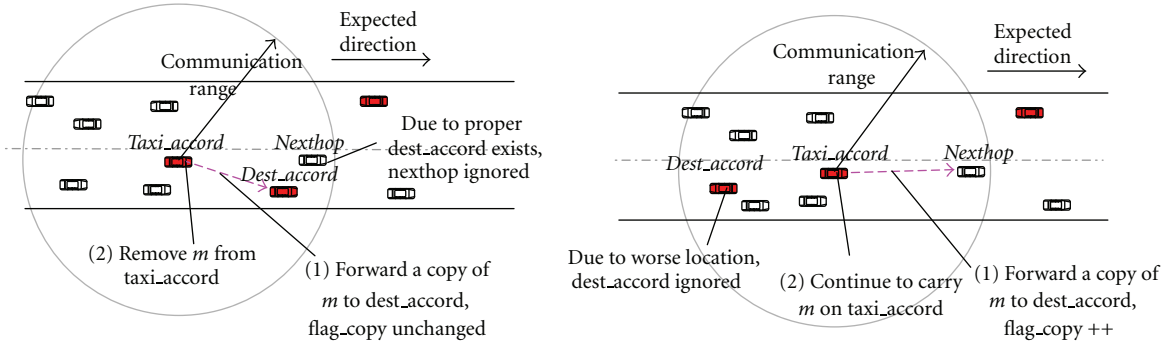
4.1. Assumption. Every vehicle can obtain its current location through GPS device, and be equipped with a preloaded street-level digital map, which not only describes road topology (see Figure 4) and traffic light period but also provides traffic statistics such as traffic density and average vehicle speed on roads at different times of the day. Such kind of digital map has already been commercialized [20], and more detailed traffic statistics will be integrated into digital map in the near future. Vehicles communicate with each other through short range wireless channel, and can find their neighbors through beacon messages. Each beacon provides vehicle's information such as its unique id, location, velocity, direction. What is more, each public transportation vehicle will announce its driving paths in beacon message.

4.2. Data Delivery for Ordinary Vehicles. In VDTNs, a vehicle needs to send messages in three cases as follows: (a) the vehicle itself produces messages; (b) it forwards the message what it received from other vehicles; (c) it periodically takes from its routing buffer to send. When an ordinary vehicle v_i have a message M to send, it firstly operates the specified routing algorithm to choose the optimal nexthop vehicle



(a) Both *Nexthop* and *Dest_accord* exist (b) Only one of *Nexthop* and *Dest_accord* exists

Figure 2: Message forwarding process for ordinary vehicles.



(a) Proper *Dest_accord* exists (b) Proper *Nexthop* exists

Figure 3: Message forwarding process for public transportation vehicles.

from his neighbor list, and the chosen one is indicated as *Nexthop*; at the same time, the vehicle also looks for the vehicles whose driving paths will pass the destination of the message. If there is more than a matching vehicle, then the one with the fastest average driving speed is selected, and its vehicle identification is denoted as *Dest_accord*. As shown in Figure 2, as to the ordinary vehicle v_i after completing the above query steps, there may appear the following cases.

Case 1. Both *Nexthop* and *Dest_accord* are found and different: the current vehicle v_i send M to *Nexthop*, and then generates a copy of M, denoted as M', increment flag_copy of M' and send it to *Dest_accord*. Lastly, v_i deletes M from its routing queue.

Case 2. Only one among *Nexthop* and *Dest_accord* is found, or *Nexthop* is equivalent to *Dest_accord*: v_i send M to *Nexthop*, and deletes the message from its queue.

Case 3. Neither *Nexthop* or *Dest_accord* is found: this means that v_i has no proper nexthop, it only puts M into its routing queue.

4.3. Data Delivery for Public Transportation Vehicles. When a public transportation vehicle pv_i has a message M to send, it firstly checks whether pv_i itself will pass through the destination of M. If pv_i does not pass through, then the forwarding process is similar with that of ordinary vehicles. Otherwise, as shown in Figure 3, pv_i will forward M according the steps as follows.

Step 1. Firstly, pv_i looks for the vehicles that will pass through the destination of M. If found none, then go to Step 3, else pv_i continues to check whether found more than a matching vehicle. If so, then the one with the fastest average driving speed is selected, and its vehicle identification is denoted as *Dest_accord*, and go to Step 2.

Step 2. The current public transportation vehicle pv_i executes the routing algorithm, and judges whether is better than *Dest_accord*, if not, then goes to Step 3, else sends M to and deletes the message from its routing queue. The message forwarding process ends.

Step 3. Vehicle pv_i operates the routing algorithm to choose the optimal nexthop vehicle from his neighbor list, and the chosen one is indicated as *Nexthop*. If *Nexthop* are not found then pv_i puts M into the routing queue; otherwise, the current vehicle pv_i generates a copy of M, denoted as M', increment flag_copy of M' and send it to *Nexthop*. M is puts into the routing queue. The message forwarding process ends.

When receiving a message from another vehicle, the current firstly checks whether it has keep a copy of the message in the routing queue. If so, it directly drops the message; else puts the message into its queue.

4.4. Queue Management Algorithm. For each mobile vehicle node in VDTNs, the size of its routing buffer queue is limited. Therefore, the queue management algorithm would greatly influence the data delivery performance. In the proposed PTDD, the flag_copy of a message indicates how many copies have been propagated, and the survival time shows how long a certain message has existed in the network. Therefore the flag_copy and survival time together denote the importance of a message, and the queue management is just based on the two parameters.

Messages are sorted in the routing queue based on a increasing order of their flag_copy. For those messages with the same flag_copy value, they are further sorted according to an increasing order of survival time. Thus messages with smaller tickets and shorter survival time are closer to the top of the queue, and can be transmitted with higher priorities. Moreover, messages will be dropped in the following two occasions: (a) when a message arrives and the queue is full, it is compared with that message at the end of the queue, and the one with bigger flag_copy is dropped among them. If the flag_copy values of the two messages are equal, then the one with a longer survival time is dropped; (b) whenever on, e message's survival time is longer than the delay tolerance of the network, it is dropped to avoid unnecessary resource occupation.

5. Performance Evaluations

In this section, we will evaluate the impact of the proposed PTDD scheme on the data transmission performance in VDTNs. We choose the classic wireless ad hoc network routing algorithm GPSR [4] as the referential. Since pure GPSR has no carry-and-forward ability, which will result in very poor performance in intermittently connected VDTNs, so that we extend it by adding buffers to make it have basic carry-and-forward ability. In the following simulation experiments, the data delivery performances of GPSR with simple carry-and-forward ability, denoted as GPSR (with buffer), and GPSR supported with PTDD scheme, denoted as PTDD (based on GPSR), will be analyzed and evaluated in terms of data delivery successful ratio and delivery delay.

5.1. Simulation Setting. As shown in Figure 3, the experiment is based on a 4000 m × 3000 m approximate rectangle street area, which is derived and normalized from a real street map of Beijing city in China. The rectangle contains 25 intersections and 40 bidirectional roads. We utilize NS-2.34 as the simulation tools. Since modeling of complex vehicle movement is important for accurately evaluating protocols, the open source software VanetMobiSim-1.1 [21] is used to generate the movement of vehicles. In our simulation, the critical vehicle is set as 6 neighbors per vehicles. Detailed simulation parameters are shown in Table 1.

Table 1: Simulation parameters.

Parameter	Value
Simulation area	4000 m × 3000 m
Number of intersections	25
Number of mobile vehicles	100, 200
Proportion of taxis	0–30%
Number of packet senders	20
Communication range	250 m
Vehicle velocity	40–80 km/h
CBR rate	0.1–1 packet per second
Data packet size	512 B
Buffer size	200 packets
Vehicle beacon interval	1 second

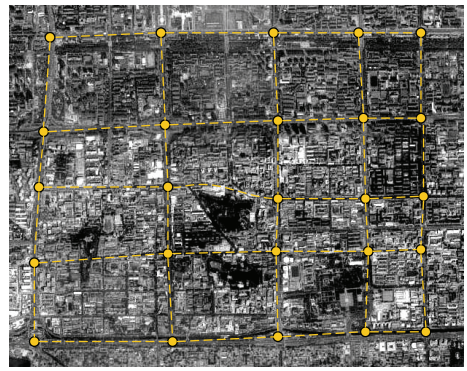


Figure 4: Road topology used in the simulation.

5.2. The Impact of Public Trasportaion Vehicle Proportion on Performance. Figure 5 shows the impact of public transportation vehicle proportion on data delivery performance as to PTDD (based on GPSR), where the data sending ratio is 0.1 packet/s, the total number of simulation vehicles is 100 and 200, respectively, and the average vehicle driving speed varies from 40 to 80 km which depends on road speed limit, vehicle density, and influence of traffic lights. From this figure, we can see that the data delivery ratio remarkably goes up with the proportion of public transportation vehicles increasing from 0 to 30% accounting for the whole vehicle number under the two cases of 100 and 200 simulation vehicles. This demonstrates that the bigger proportion the public transportation vehicles accounts for, the better performance PTDD scheme will reach. For the status that the vehicle total number is 100, the data delivery ratio is improved from 46 to 83% (the relative growth rate is 80.4%) as the proportion of public transportation increases from 0 to 30%. As to the status that the simulation vehicle number is 200, the data delivery ratio goes up from 55 to 89% (the relative growth rate is 61.8%) under the same growth proportion of public transportation. This reflects that the performance of DPPR scheme is improved more obviously in low vehicle density and sparse connection environment.

5.3. The Data Deliver Ratio. To evaluate the performance of PTDD scheme, we compare the data delivery ratios of

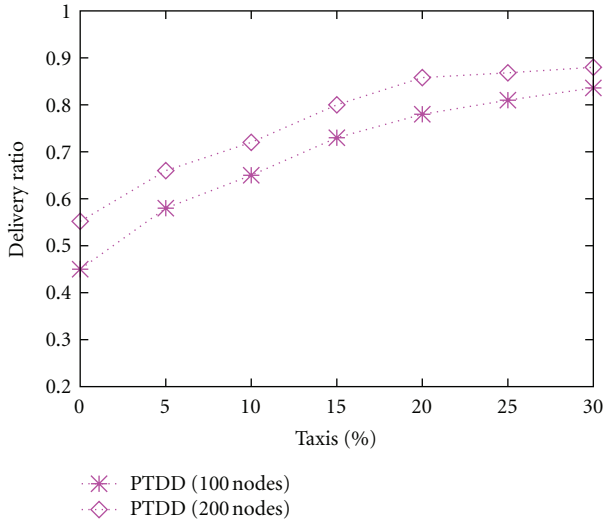


Figure 5: Impact of public transportation vehicle proportion on performance.

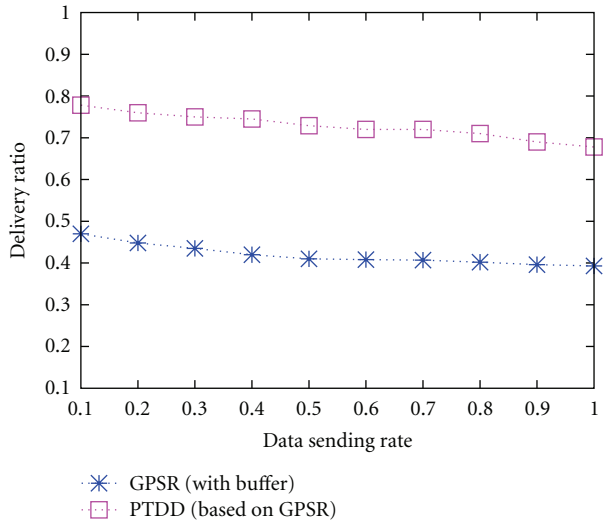


Figure 6: Data delivery ratio (100 vehicles).

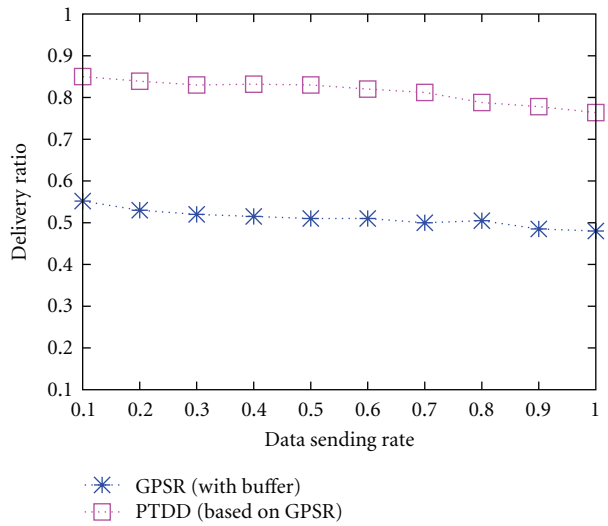


Figure 7: Data delivery ratio (200 vehicles).

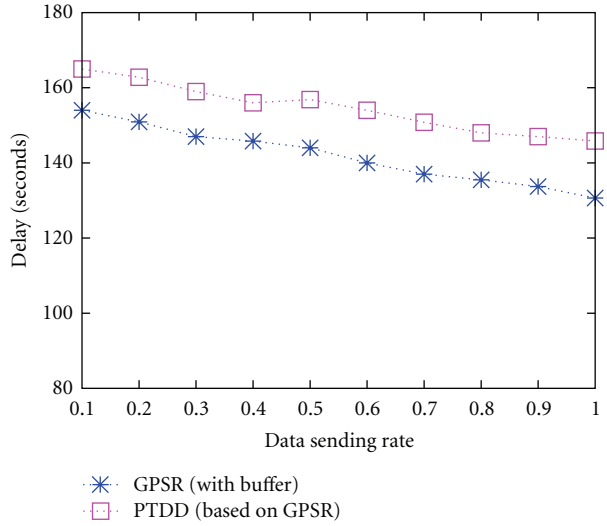


Figure 8: Data delivery delay (100 vehicles).

the two schemes, that is, GPSR (with buffer) and PTDD (based on GPSR), under different data sending rates. Here, the public transportation accounts for the total vehicles is fixed as 20%, and the CBR data sending rates is changed from 0.1 to 1 packet/s.

As shown in Figure 6, the data delivery ratio of PTDD (based on GPSR) is greatly higher than that of GPSR (with buffer) under the different data sending ratios, when the simulation vehicle number is 100. The average delivery ratio of PTDD (based on GPSR) reaches 0.74, which is far above that of GPSR (with buffer), that is, 0.43, and the relative growth rate is 72%.

In Figure 7 we can see the data delivery ratio of PTDD (based on GPSR) is also much higher than that of GPSR (with buffer) under all data sending ratios, when the vehicle number is 200. The average delivery ratio of PTDD reaches 0.82, which is far above that of GPSR, that is, 0.51, and the relative increase rate is 61%. The results show that PTDD thinking can play an important role in vehicular network environments with such features as high mobility, low vehicle density, sparse connection.

5.4. The Data Delivery Delay. From Figure 8, it can be seen that the data delivery delay in the two schemes, that is, PTDD (based on GPSR) and GPSR (with buffer), goes down with the increasing of the data sending rate in the case of 100 simulation vehicles. The reason is that the part of messages with too long survival time are dropped because the size of routing buffer is limited, when the increasing data sending rate results in the fast increase of transmission overhead.

From Figure 9, it can be seen that the data delivery delay goes down with the increasing of the data sending rate in

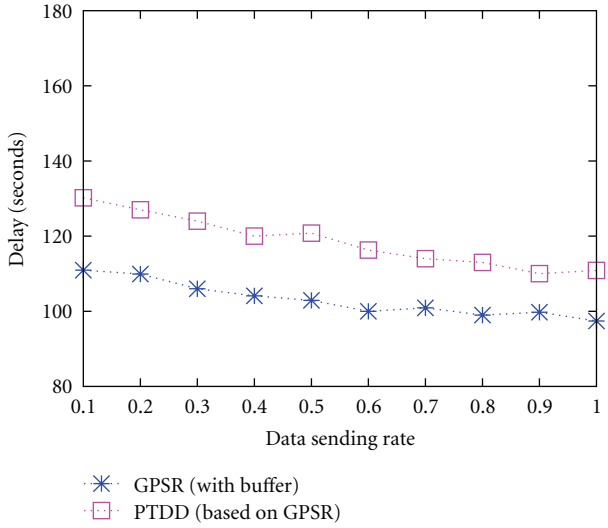


Figure 9: Data delivery delay (200 vehicles).

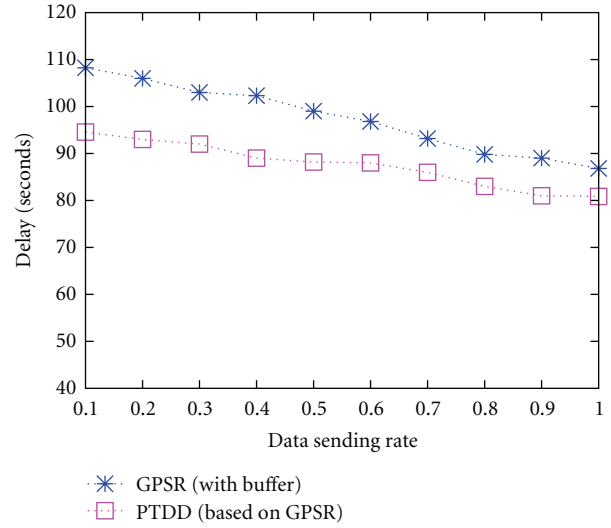


Figure 10: The lowest 75% data delivery delay (100 vehicles).

the case of 200 simulation vehicles. But the delay decrease speeds of PTDD and GPSR are different as the data sending rate increases, and PTDD shows a sharper decrease trend. The reason is that PTDD is a multiple copy scheme and it generates more data packet copies the GPSR, especially for the case of high vehicle density.

From Figures 8 and 9, we also notice the delivery delay of PTDD (based on GPSR) is longer than that of GPSR (with buffer). The reason is that there are many data messages which cannot find proper next hop to delivery and thus are dropped in GPSR routing, but in PTDD scheme the messages are likely to be successfully delivered by public transportation vehicles with carrying way when the public transportation vehicles pass through the messages' destinations. To better study the delivery delay, we only examine the "the lowest 75% delivery delay," which is the average delay of the lowest 75% packets according to the method proposed in literature [7]. As shown in Figure 10, the delivery delay of PTDD (based GPSR) scheme is clearly lower than that of GPSR (with buffer) in the case of 100 simulation vehicles. In Figure 11, the similar results also can be seen, that is the delivery delay of PTDD scheme is lower than that of GPSR when the vehicle number is 200.

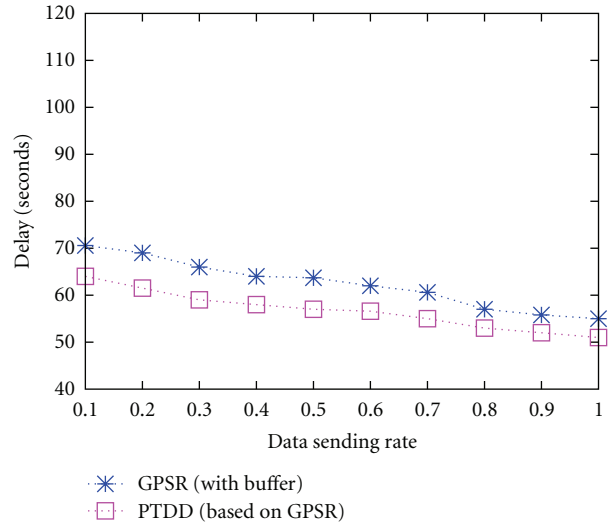


Figure 11: The lowest 75% data delivery delay (200 vehicles).

6. Conclusion

As an important constituent part of traffic system, public transportation vehicles have the features such as long running time, no privacy protection request. We think it is important and interesting research topic to effectively utilize the features of public transportation vehicles to improve the data delivery performance in VDTNs. In this paper, we firstly present a taxi driving path prediction method based destination information gathering. By doing, public transportation vehicles can foresee their driving paths. Through utilizing the features, we propose a new data delivery scheme call PTDD to improve the performance of data delivery of Vehicular Delay Tolerant Networks. Through simulation experiments, we analyze and evaluate the performance of GPSR (with buffer) and PTDD (based on GPSR) scheme. The experiment result shows that PTDD scheme can significantly improve the data delivery performance in VDTNs.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant no. 61262081; the Yunnan Provincial Applied Fundamental Research Project under Grant no. KKSY201203027.

References

[1] J. Ott and D. Kutscher, "Drive-thru internet: IEEE 802.lib for "automobile" users," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM '04)*, pp. 362–373, March 2004.

[2] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, 2008.

[3] Research and I. T. A. (RITA), "IntelliDrive: safer, smarter and greener," http://www.its.dot.gov/.

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

[5] V. Naumov and T. R. Gross, "Connectivity-aware routing (CAR) in vehicular ad hoc networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1919–1927, May 2007.

[6] Y. Ding and L. Xiao, "SADV: static-node-assisted adaptive data dissemination in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2445–2455, 2010.

[7] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular Ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.

[8] J. Jeong, S. Guo, Y. Gu, T. He, and D. H. C. Du, "Trajectory-based data forwarding for light-traffic vehicular Ad Hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 743–757, 2011.

[9] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "Trajectory-based statistical forwarding for multihop infrastructure-to-vehicle data delivery," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1523–1537, 2012.

[10] F. Xu, S. Guo, J. Jeong et al., "Utilizing shared vehicle trajectories for data forwarding in vehicular networks," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 441–445, April 2011.

[11] V. Namboodiri and L. Gaw, "Prediction based routing for Vehicular ad hoc networks," in *Proceedings of the Vehicular Ad Hoc Networks (VANET '04)*, 2004.

[12] H. Menouarand, M. Lenardi, and F. Filali, "Movement Prediction-based Routing (MOPR) concept for position-based routing in vehicular networks," in *Proceedings of IEEE 66th Vehicular Technology Conference (VTC '07-Fall)*, pp. 2101–2105, October 2007.

[13] Z. Li, Y. Zhu, and M. Li, "Practical location-based routing in vehicular ad hoc networks," in *Proceedings of IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (MASS '09)*, pp. 900–905, October 2009.

[14] H. Jeung, M. L. Yiu, X. Zhou, and C. S. Jensen, "Path prediction and predictive range querying in road network databases," *VLDB Journal*, vol. 19, no. 4, pp. 585–602, 2010.

[15] http://www.bjzx.gov.cn/html/meet5th/meet5th/yxta/0598.htm.

[16] http://www.china.com.cn/city/txt/2007-01/15/content_7655169.htm.

[17] Y. Lee, H. Lee, N. Choi, Y. Choi, and T. Kwon, "Macro-level and Micro-level Routing (MMR) for urban vehicular ad hoc networks," in *Proceedings of the 50th Annual IEEE Global Telecommunications Conference (GLOBECOM '07)*, pp. 715–719, November 2007.

[18] A. Vahdat and D. Becker, "Epidemic routing for partially-connected Ad Hoc networks," Technical Report, 2000, http://issg.cs.duke.edu/epidemic/epidemic.pdf.

[19] H. Lee, Y. Lee, T. Kwon, and Y. Choi, "Virtual Vertex Routing (VVR) for course-based vehicular ad hoc networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 4408–4413, March 2007.

[20] "Digital Map Data," 2012, http://www.mapmechanics.com/digital-map-data/.

[21] J. Härri, F. Filali, C. Bonnet, and M. Fiore, "VanetMobiSim: generating realistic mobility patterns for VANETs," in *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET '06)*, pp. 96–97, ACM, New York, NY, USA, September 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 290717, 12 pages
doi:10.1155/2012/290717

Research Article

Ant-Based Transmission Range Assignment Scheme for Energy Hole Problem in Wireless Sensor Networks

Ming Liu and Chao Song

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Sichuan, Chengdu 611731, China

Correspondence should be addressed to Ming Liu, csmliu@uestc.edu.cn

Received 13 September 2012; Accepted 9 October 2012

Academic Editor: Nianbo Liu

We investigate the problem of uneven energy consumption in large-scale many-to-one sensor networks (modeled as concentric coronas) with constant data reporting, which is known as an energy hole around the sink. We conclude that lifetime maximization and the energy hole problem can be solved by searching optimal transmission range for the sensors in each corona and then prove this is an NP-hard optimization problem. In view of the effectiveness of ant colony algorithms in solving combinatorial optimization problems, we propose an ant-based heuristic algorithm (ASTRL) to address the optimal transmission range assignment for the goal of achieving life maximization of sensor networks. Experimentation shows that the performance of ASTRL is very close to the optimal results obtained from exhaustive search method. Furthermore, extensive simulations have also been performed to evaluate the performance of ASTRL using various simulation parameters. The simulation results reveal that, with low communication cost, ASTRL can significantly mitigate the energy hole problem in wireless sensor networks with either uniform or nonuniform node distribution.

1. Introduction

Rapid technological advances in microelectromechanical systems (MEMS) and low-power wireless communications have enabled the deployment of large scale wireless sensor networks (WSNs). The potential applications of sensor networks are highly varied, such as environmental monitoring, target tracking, and battlefield surveillance [1, 2]. Due to limited and nonrechargeable energy provision, the energy resource of sensor networks should be managed wisely to extend the lifetime of sensors [3–7].

The sink node in a WSN receives the data from the sensor nodes and forwards these data to the applications over the WSN. Usually, the sensor nodes closest to the sink tend to deplete their energy budget more rapidly than others [8–10] because such nodes need to transmit more data than other nodes. This causes the problem of energy hole around the sink. A WSN suffering from the energy hole problem cannot deliver more data, and consequently the network lifetime has been greatly shortened, although most of the sensor nodes can still work properly.

Recently, there have been a number of studies done on the energy hole problem for improving the network lifetime. Generally, these studies aiming to mitigate or solve the energy hole problem can be divided into 3 categories: (i) assistant approaches, such as deployment assistance, traffic compression, and aggregation in [11]; (ii) node distribution strategies. Lian et al. in [9] propose a nonuniform sensor distribution strategy. The density of sensors increases when their distance to the sink decreases; (iii) adjustable transmission range. Jarry et al. [12] propose a mixed routing algorithm which allows each sensor node to send a message either to one of its immediate neighbors or to the base station directly.

Since adjusting transmission range of sensors is a promising way to be used for prolonging lifetime of sensor networks, we solve the energy hole problem by performing optimizing the transmission range assignment based on corona model in [8]. We prove that the problem of optimal transmission range assignment in coronas to achieve minimum energy consumption is an NP-hard problem, and therefore an approximation algorithm with low communication cost should be proposed for network lifetime

optimization. However, the existing researches [3, 12, 13] on addressing energy hole problem in category (iii) mainly work in a preplanned manner. The cooperation, communication, and management are deliberately modeled, designed, and tuned before the deployment of sensors. These methods usually ignore the requirements of self-adaptation and self-calibration and always produce complex protocols with higher overhead and just passable performance, which act dully to the change of environment. Fortunately, inspired by the ecosystem, some biologic models are applied in networks [14–16]. These biologic models exhibit swarm intelligence in pursuing a global optimal goal and throw new light on the energy hole problem in WSN.

Recently, the Ant Colony Algorithm (ACO) has been widely used in solving the combinatorial optimization problems. Through the simple cooperation of solo entity and the positive feedback mechanism, ACO outperforms tradition manners in terms of self-adaptation and self-calibration. In this paper, we propose an ant-based algorithm (ASTRL) for mitigating the energy hole problem in order to prolong the lifetime of networks with different node distributions. As far as we know, we are the first to use bioinspired methods to solve the energy hole problem in WSNs.

The remainder of the paper is organized as follows. Section 2 presents our literature review. Section 3 introduces the system assumption used throughout our work and then analyzes the energy hole problem and concludes that the problem of searching optimal transmission range list is a multiobjective problem. Section 4 gives the design details of ANT-based algorithm for mitigating energy hole problem in WSNs. Section 5 shows the effectiveness of the ASTRL via extensive simulations. Section 6 concludes this paper.

2. Related Works

Li and Mohapatra [17] investigate the problem of uneven energy consumption in a large class of many-to-one sensor networks. The authors describe the energy hole in a ring model (like corona model) and present the definitions of the per node traffic load and the per node energy consuming rate (ECR). In a many-to-one sensor network, all sensor nodes generate constant bit rate (CBR) data and send them to a single sink via multihop transmissions. Based on the observation that sensor nodes sitting around the sink need to relay more traffic compared to other nodes in outer subregions, their analysis verifies that nodes in inner rings suffer much faster energy consumption rates and thus have much shorter expected lifetime. The authors term this phenomenon of uneven energy consumption rates as the energy hole problem, which may result in serious consequences, for example, early dysfunction of the entire network. The authors present some approaches to the energy hole problem, including deployment assistance, traffic compression, and aggregation. Jarry et al. [12] propose an algorithm to resolve the energy hole problem, which uses mobile sensors to heal energy holes. The cost of these assistant approaches is a lot.

Lian et al. [9] argue that, in static situations, for large-scale networks, after the lifetime of the sensor network is over, there is still a great amount of energy left unused, which can be up to 90% of total initial energy. Thus, the static models with uniformly distributed homogenous sensors cannot efficiently utilize their energy. The authors propose a nonuniform sensor distribution strategy. The density of sensors increases when their distance to the sink decreases. Their simulation results show that, for networks with high density, the nonuniform sensor distribution strategy can increase the total data capacity by an order of magnitude. Wu et al. [18] propose a nonuniform node distribution strategy to achieve the subbalanced energy depletion. The authors state that, if the number of nodes in coronas increases from corona C_{R-1} to corona C_1 in a geometric progression with common ratio $q > 1$ and there are $N_{R-1}/(q-1)$ nodes in corona C_R, then the network can achieve subbalanced energy depletion. Here, Ni denotes the number of nodes in corona C_i. But the node distribution strategy can hardly work in the real world, because in most cases the node distribution is random and hence an uncontrollable node density in local area.

Olariu and Stojmenovic [8] discuss the relationship between the network lifetime and the width of each corona in concentric corona model. The authors prove that, in order to minimize the total amount of energy spent on routing along a path originating from a sensor in a corona and ending at the sink, all the coronas must have the same width. However, the authors assume that all nodes in corona C_i should forward data in corona C_{i-1}, and the transmission range in corona C_i is $r_i - r_{i-1}$ (here C_i is the subarea delimited by the circles of radii r_{i-1} and r_i). If each corona has different width and different transmission range, we think this assumption may lead to the waste of energy for transmission. For example, as shown in Figure 1, the width of corona C_i is larger than that of corona C_{i-1} and all nodes in C_i have the same transmission range of $r_i - r_{i-1}$ that is larger than the width of corona C_{i-1}. Divide the corona C_i into two subcoronas, namely, s_1 and s_2 (see in Figure 1). The width of subcorona s_1 is equal to that of corona C_{i-1}, so nodes in s_1 will transmit data to C_{i-1}. The nodes in subcorona s_2 which are close to corona C_{i-1} with transmission range larger than the width of corona C_{i-1} may transmit data across corona C_{i-1} to corona C_{i-2} that is closer to the sink node. Because of the authors assumption that the data transmitted from all nodes in corona C_i should be forwarded for the next hop in corona C_{i-1} rather than corona C_{i-2}, these nodes in s_2 with transmission range $r_i - r_{i-1}$ which can transmit data to C_{i-2} but should transmit to C_{i-1} will waste energy for transmission.

For balancing the energy load among sensors in the network, Jarry et al. [12] propose a mixed routing algorithm which allows each sensor node to send a message either to one of its immediate neighbors or to the base station directly. The decision about the next receiver is determined by a potential function depending on its remaining energy. However, when the network area radius is bigger than the sensors maximal transmission range, the proposed algorithm cannot be applicable.

Some biologically intelligent algorithms have been used to solve routing problems in networks. Di Caro and Dorigo [19] propose AntNet, an approach to the adaptive learning of routing tables in communications networks. AntNet,

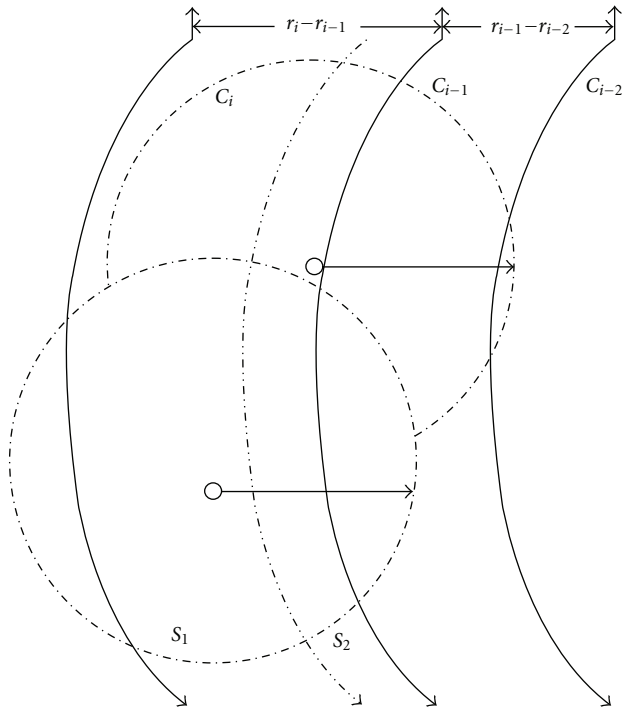


FIGURE 1: The energy problem caused by corona model with different width.

the routing algorithm they proposed, is a mobile agents system showing some essential features of parallel replicated Monte Carlo systems. The algorithm takes inspiration from previous work on artificial ant colonies techniques to solve combinatorial optimization problems and telephone network routing. The core ideas of these techniques are (i) the use of repeated and concurrent simulations carried out by a population of artificial agents called ants to generate new solutions to the problem, (ii) the use by the agents of stochastic local search to build the solutions in an incremental way, and (iii) the use of information collected during past simulations to direct future search for better solutions.

Ant routing has shown excellent performance for sensor networks. Aghaei et al. [20] present an AntNet-based routing network, which meets the enhanced sensor network requirements, including energy consumption, success rate, and time delay. Okdem and Karaboga [21] have developed a routing scheme and adapted ACO algorithm to this scheme to get a dynamic and reliable routing protocol. Their algorithm has also been implemented in router chip, which provides an easy handling of WSN routing operations for sensor node designers.

3. System Model

In this section, the network model, energy model and corona model will be presented, respectively.

3.1. Network Model. We assume our sensor network model as follows: (1) once deployed, the sensors must work unattended, and all sensor nodes are static. Each sensor has a nonrenewable energy budget, and the initial energy of each sensor is $\epsilon > 0$; (2) each sensor has a maximum transmission range, denoted as t_x, and assumed to be much smaller than R (the furthest possible distance from a sensor to the sink node); (3) sensors are required to send their sensed data constantly at a certain rate. For sake of simplicity, we assume that each sensor node generates and sends L bits of data per unit time; (4) we assume there is a perfect MAC layer in the network, that is, transmission scheduling is so perfect that there is no collision and retransmission. Initially the network is well connected. The issue that what node density can ensure network connectivity is investigated in [13]; (5) based on greedy forwarding approach, sensor nodes transmit data packets to the sink. In such greedy forwarding scheme, data packets are transmitted to the next-hop node which is closer to the destination.

Definition 1 (network lifetime). Li and Mohapatra in [17] present the definition of system lifetime, which is the time till a proportion of nodes die. A corona of sensor nodes in the network is said to be dead when it is unable to forward any data or send its own data. So the network lifetime in this paper is defined as the duration from the very beginning of the network until the first corona of sensor nodes dies.

3.2. Energy Model. A typical sensor node comprises three basic units: sensing unit, processing unit, and transceivers. Our energy model only involves the power for receiving and transmitting data without considering the energy consumed for sensing and processing data, which depends on the computation hardware architecture and the computation complexity. According to [17], the energy consumption formulas that we use in the analysis and simulations throughout the rest of this paper are as follows:

$$E_{\text{trans}} = (\beta_1 + \beta_2 d^{\alpha}) l, \tag{1}$$

$$E_{\text{rec}} = \beta_3 l. \tag{2}$$

Here E_{trans} denotes the energy consumption for transmitting and E_{rec} denotes the energy consumption for receiving. l (in bits/sec) is the data rate of each sensor node, and α is 2 or 4. The term d_{α} accounts for the path loss. According to [17], some typical values of the above parameters in current sensor technologies are as follows:

$$\begin{aligned} \beta_1 &= 45 \times \frac{10^{-9}\,\text{J}}{\text{bit}}, \\ \beta_2 &= 10 \times \frac{10^{-12}\,\text{J/bit}}{\text{m}^2} \quad (\text{when } \alpha = 2), \end{aligned} \tag{3}$$

or

$$\begin{aligned} \beta_2 &= 0.001 \times \frac{10^{-12}\,\text{J/bit}}{\text{m}^4} \quad (\text{when } \alpha = 4), \\ \beta_3 &= 139 \times \frac{10^{-9}\,\text{J}}{\text{bit}}. \end{aligned} \tag{4}$$

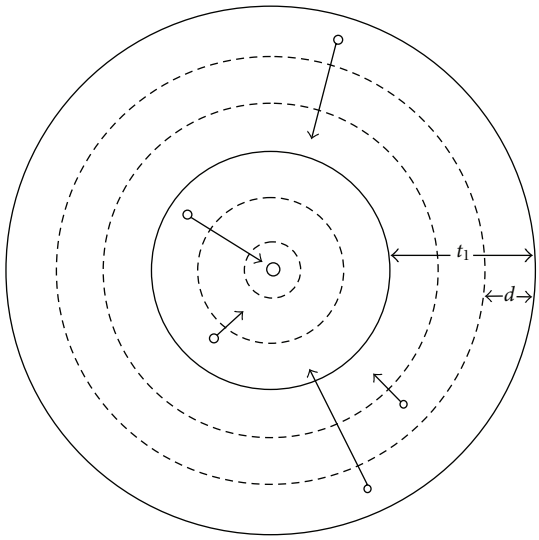


Figure 2: Illustration of adjustable transmission ranges.

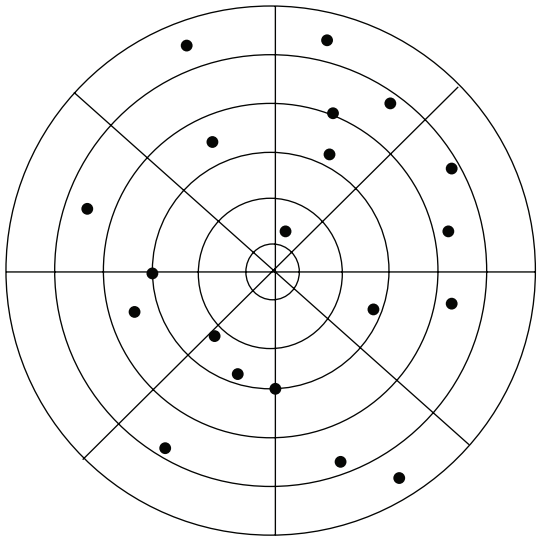


Figure 3: Concentric coronas.

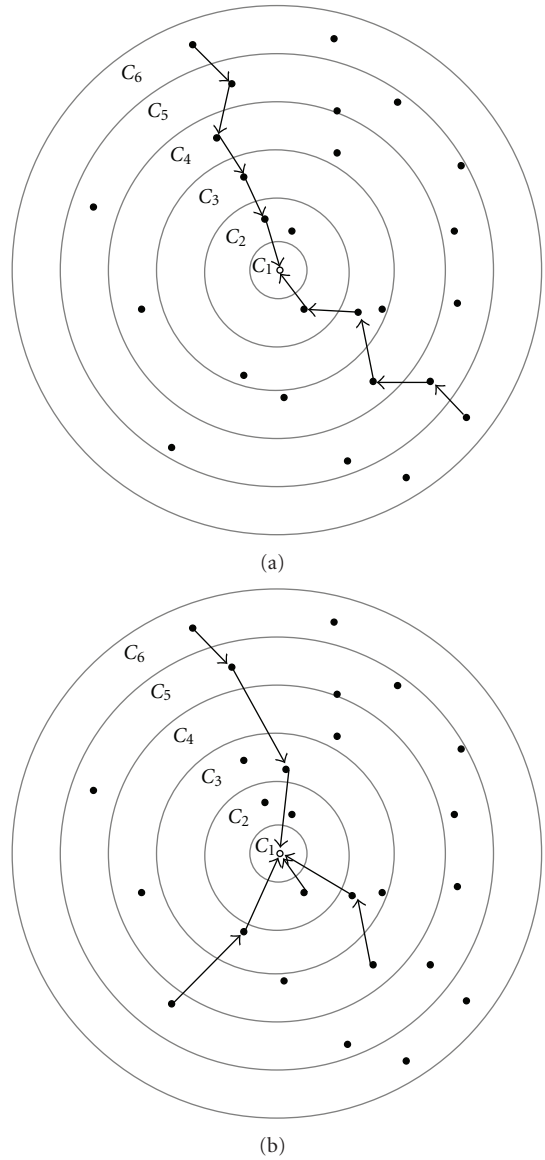


Figure 4: Two data forwarding patterns in corona model.

3.3. Corona Model with Adjustable Transmission Range. In order to save energy, sensors can adjust their transmission ranges. For simplicity, we divide the maximum transmission range into k levels, that is, $\{(1/k)t_x, (2/k)t_x, (3/k)t_x, \ldots, t_x\}$. As shown in Figure 2, each sensor has k levels of transmission range to choose. The Unit Length of Transmission Range (ULTR) is denoted by d:

$$d = \frac{t_x}{k}. \tag{5}$$

We partition the whole area with radius R into m adjacent concentric parts termed coronas. They are presented as follows (see Figure 3), which has been discussed in [22, 23]. The width of each corona is 1ULTR therefore,

$$m = \frac{R}{d}. \tag{6}$$

Assume that all the sensor nodes located in the same corona have the same transmission range, and the transmission ranges of the sensor nodes located in different coronas can be different. Therefore, if the transmission range of one node is i ULTR; its real transmission range can be calculated by the following equation:

$$i \times d = i \times \left(\frac{t_x}{k}\right). \tag{7}$$

Generally, there are the following two data forwarding patterns in corona model.

(1) $k = 1$. As shown in Figure 4(a), corona C_i relays all the data generated or forwarded by corona C_{i+1} to the sink node.

(2) $k > 1$. The next-hop corona of corona C_{i+1} is determined by its assigned transmission range, which

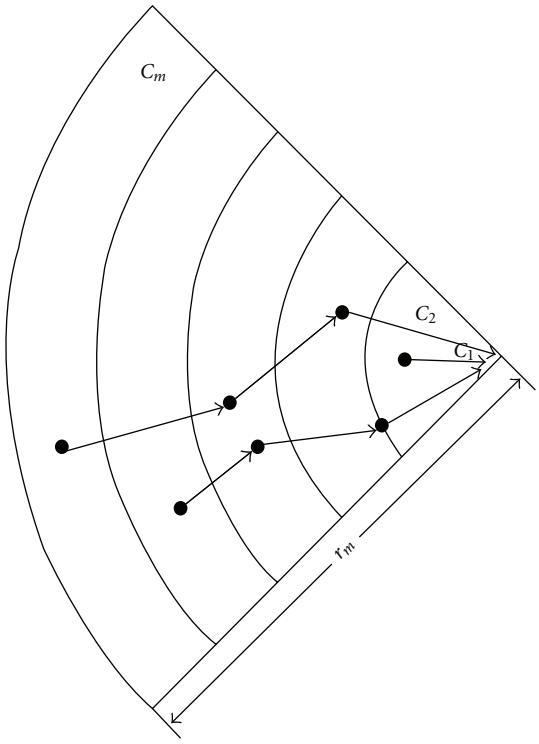


Figure 5: A sector W and the associated subcoronas.

has been illustrated in Figure 4(b). Here, we use TRL to indicate the transmission range list of all coronas. Intuitively, the second data forwarding pattern can do better in solving energy hole problem, and therefore it has been adopted in this paper.

4. Ant-Based Algorithm for Searching Transmission Range List (ASTRL)

Consider an arbitrary wedge W subtended by an angle of θ, and refer to Figure 5. W is partitioned into m sectors $C_1, C_2, \ldots, C_m$ by its intersection with m concentric circles, centered at the sink, and of monotonically increasing radii $r_1 < r_2 < \cdots < r_k = R$. For convenience of notation we write $r_0 = 0$ and interpret C_0 as the sink itself. Each sector C_i selects a node as a corona head denoted as H_i to determine the transmission range of all nodes in this sector.

4.1. Construction Graph. Each node has k transmission range levels to choose, so nodes in each corona have k coronas to be the possible next hop. Figure 6(a) shows all available routes with $k = 2$, termed as construction graph in ACO. In the construction graph, vertex denotes each subcorona. And if sector C_i has a transmission range which can transmit data to sector C_j, there is a directed edge (C_i, C_j) from C_i to C_j. The artificial ants can construct solutions in the construction graph by randomized walks.

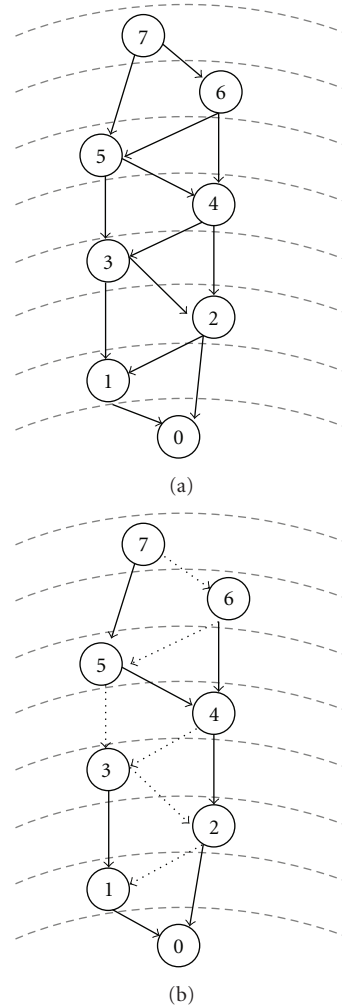


Figure 6: Construction graph and spanning tree.

The characters of the construction graph are as follows.

(1) Out-degree: vertexes with ID bigger than k have k out-degrees, and the out degree of each vertex whose ID is not-bigger than k is equal to its sector ID.

(2) In-degree: the vertexes of outmost k sectors have $(m-i)$ in-degrees (where m is the number of sectors and i is the sector ID), and each of other vertexes has k in-degrees, including the sink node.

(3) The edge (A_i, A_j) $(1 \le i,\ j \le m)$ should satisfy the condition $i > j$.

In Section 3, we have discussed that, in order to mitigate energy hole problem and maximize network lifetime, we need to search an optimal transmission range list. According to the list we can obtain a spanning tree from the construction graph (see Figure 6(b)). The root of the obtained spanning tree is sink node. The following are the characters of the spanning tree.

(1) Out-degree: each vertex has only one out-degree.

(2) In-degree: each vertex has not more than k in-degrees.

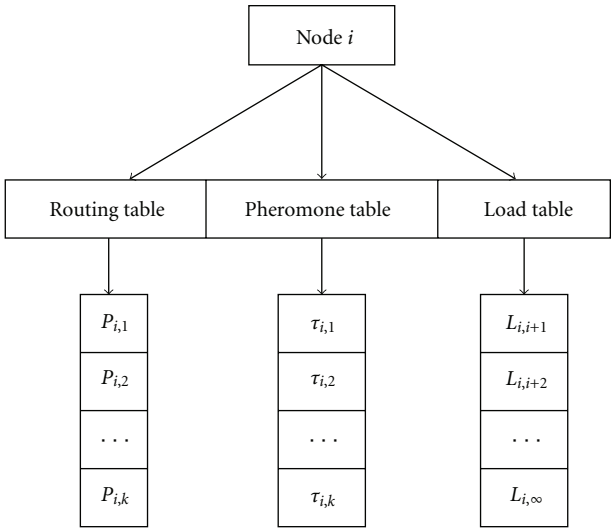


Figure 7: The energy problem caused by corona model with a different width.

(3) The edge (A_i, A_j) $(1 \le i,\ j \le m)$ should satisfy the condition $i > j$.

Because searching optimal transmission range lists is NP hard, we propose an ant-based algorithm to achieve optimizing using energy for transmission. In the construction graph, artificial ants generated from each vertexes travel toward the sink node to discover feasible paths with low energy consumption.

4.2. ASTRL Design. In ASTRL, we employ two types of ants: (i) forward ant (*Fant*), which travels from the source nodes to the sink node, and (ii) backward ant (*Bant*), which is generated by *Fant* when *Fant* reaches the sink node.

The artificial ants read and write in the following three data structures stored in each corona head (see in Figure 7).

(1) A pheromone table R_i, which records the pheromone of each edge started from the vertex of this corona in construction graph. Let $\tau_{i,j}$ denote the edge (i, r) with transmission range j, so $r = i - j$. The initial pheromone of edge (i, r) is obtained as follows:

$$\tau_{i,j} = \frac{1}{W_{i,j}(0)} = \frac{1}{L[\beta_1 + \beta_2(x_i d)^{\alpha}]}. \tag{8}$$

Here, $W_{i,j}(0)$ denotes the *ECR* value of sending data generated by itself with transmission range level j during unit time in C_i. After each interval δt, the pheromone of each edge in construction graph will be reduced, termed as evaporation. Let γ denote the evaporation coefficient, which can balance the ability of exploring new routes and that of storing energy-efficient routes:

$$\tau_{i,j} \longleftarrow \tau_{i,j} - \gamma\tau_{i,j}. \tag{9}$$

(2) A routing table T_i, with probabilistic entries. T_i defines the probabilistic routing policy currently adopted at corona head H_i for each transmission range j: for the destination C_0 (sink node) and for each neighbor vertex $(i - j)$, T_i stores a probability value $P_{i,j}$ expressing the goodness (desirability), under the current network routing policy, of choosing j as the next transmission range. Based on this, ants can explore better routing path. For each corona C_i, the probabilities stored should be satisfied as follows:

$$\sum_{j=1}^{k} P_{i,j} = 1, \quad i \in [1, m]. \tag{10}$$

(3) The per node traffic load of corona C_i generated from each outer corona, denoted as $L_{i,j}$, where i is the ID of corona C_i and j is the ID of outer corona. The values are obtained through an initializing process: during the initial unit time, each corona generates data at the rate of l, then sends them, and forward the data generated by outer coronas to the sink with transmission range d, and each data packet records the ID of corona which generates it. Therefore, the nodes of corona C_i can receive data generated from $C_{i+1}, C_{i+2}, \ldots, C_m$ and store the per node traffic load generated from these coronas, denoted as $L_{i,i+1}, L_{i,i+2}, \ldots, L_{i,m}$. In ATSRL, if an ant g_j arrives in corona head H_i, and the coronas which g_j has passed through can be a set $\text{path}_j = j_1, j_2, \ldots, j_n$, obviously $j_n \le m - i$. Therefore, the *ECR* value of forwarding g_j by the corona head H_i with transmission range level u is

$$W_j = L[\beta_1 + \beta_2(x_i d)^{\alpha}] + \left(\sum_{j=1}^{v} \sum_{jy \in \text{path}_j} L_{i,jy}\right) \times [\beta_1 + \beta_2(ud)^{\alpha} + \beta_3]. \tag{11}$$

The ant-based algorithm is described as follows.

(1) At regular intervals of δt from every corona head H_i, a forward ant *Fant* is launched toward sink node. Forward ants carry the same data packets, so that they experience the same traffic loads.

(2) While traveling toward their destination nodes, the ants keep memory of their paths and of the energy consumption found. The identifier of every visited node r and the energy consumption are pushed onto a memory stack $S_i(g)$ in each ant g.

(3) At each node r, each traveling ant heading towards sink node selects from the neighbors it has not visited yet the node t to move to; otherwise, the node t will be selected from all the neighbors in case all of them had been previously visited. The neighbor t is selected with a probability (goodness) $P_{i,j}$ that is computed with pheromone $\tau_{i,j}$ of each transmission range. Here, a and b are the coefficients for determining the influence of τ and η. Because all ants travel from

one sector with bigger ID to another with smaller ID, there will be no cycle in each path:

$$P_{i,j} = \frac{\tau_{i,j}^{a}\eta_{i,j}^{b}}{\sum_{l=1}^{k}\left(\tau_{i,l}^{a}\eta_{i,j}^{b}\right)}. \quad (12)$$

(4) When sink node is reached, the ant *Fant* generates a backward ant *Bant* and transfers to it all of its memory and then dies.

(5) The backward ant *Bant* takes the same path as that of its corresponding forward ant, but in the opposite direction. When the backward ant arrived at node r along the path mentioned before, it will update its own pheromone. Backward ants do not carry any data packet; they use higher priority queues because their task is to quickly propagate to the routing tables the information accumulated by the forward ants computed in

$$\delta\tau = \frac{1}{\sum_{(i,(i-j))\in \mathrm{path}_g} W_{i,j}}. \quad (13)$$

(6) Arriving at a node r from a neighbor node f, the backward ant updates the pheromone of edge (r, f) and the routing table T_r:

$$\tau_{r,(r-f)} \longleftarrow \tau_{r,(r-f)} + \delta\tau, \quad (r, f) \in \mathrm{path}_g. \quad (14)$$

The routing table T_r is changed by incrementing the probability $\tau_{i,j}$ (i.e., the pheromone of choosing neighbor f) and decrementing the other probabilities. The amount of the variation in the probabilities depends on a measure of goodness we associate with the energy consumption experienced by the forward ant and is given above.

(7) After the optimizing time t_0, each corona head chooses the transmission range with the maximal probability in T_i as the transmission range of the corona and then broadcasts it to all nodes in this corona.

5. Simulation Results

Based on the energy model described in Section 3, we simulate the ASTRL proposed in this paper with consideration of the two strategies of deploying nodes, namely, uniform and nonuniform node distributions. Here, we have to mention that it is the greed forwarding strategy that is adopted in simulations for data forwarding, which has been presented in Section 3.

5.1. Simulation Parameters. The initial energy of each node (denoted as ε) is 50 J. The sensor nodes maximum transmission distance (denoted as t_x) is 20 m; t_x is divided into k levels, and k is 4. Each sensor node generates and sends L bits of data per second, and L is set to be 4102 bits. The node density is 5/m^2. The parameter in the energy consumption formula (1) is set to be 4, and the setting of other parameters has been introduced in Section 3.2. In ASTRL, the optimizing time (denoted as t_0) is 1000 s, and the time interval δt is 0.1 s; the evaporation coefficient γ is 0.05, and the parameters a and b are set to be 1 and 0.3, respectively. For ease of reading, we have listed all parameters in Table 1.

Table 1: Simulation parameters.

Parameter	Value
Initial energy of each node (ε)	50 J
Maximum transmission range (t_x)	20 m
Number of transmission range levels (k)	4
Length of unit data (L/sec)	4×10^{-2}
Density (ρ)	5/m^2
α	4
β_1	45×10^{-9} J/bit
β_2	10^{15} J/bit/m^4
β_3	135×10^{9} J/bit
t_0	1000 sec
δt	0.1 sec
γ	0.05
a	1
b	3

Our simulation includes three parts: (1) using an ant colony optimizing process as an example to show the variation in probabilities of using different transmission ranges during the whole process; (2) compare ASTRL with other algorithms under different situations, which include optimal algorithm and the algorithms proposed in other papers. Under any of the situations the results are an average over 100 independent runs or more; (3) discuss the influence of the parameter settings in ASTRL on its performance.

5.2. Variation in the Probabilities of Using Certain Transmission Range. With the parameters listed in Table 1, the simulations examine the variation in the probabilities of using certain transmission range of the specific coronas selected from all coronas (12 coronas) under circumstance of uniform node distribution. Figure 8 shows how the variation in the probabilities that each of the 4 coronas (i.e., C_4, C_7, C_{10}, and C_{12}) uses certain transmission ranges within 1000 s. As it shows, during the ant colony optimizing process, the farther the corona is from the sink, the more slowly its probabilities of transmission range converge; particularly, for the corona C_{12}, the variation amplitude of the probabilities is the greatest. This is because inner coronas (like C_4), which are close to the sink, have fewer route options, and thus being able to determine their transmission range earlier than outer coronas via the study of network traffic. For outer coronas, since they have to adapt to the routing status of inner coronas, they are unable to determine their transmission ranges until inner coronas have finished the determination of their own transmission ranges.

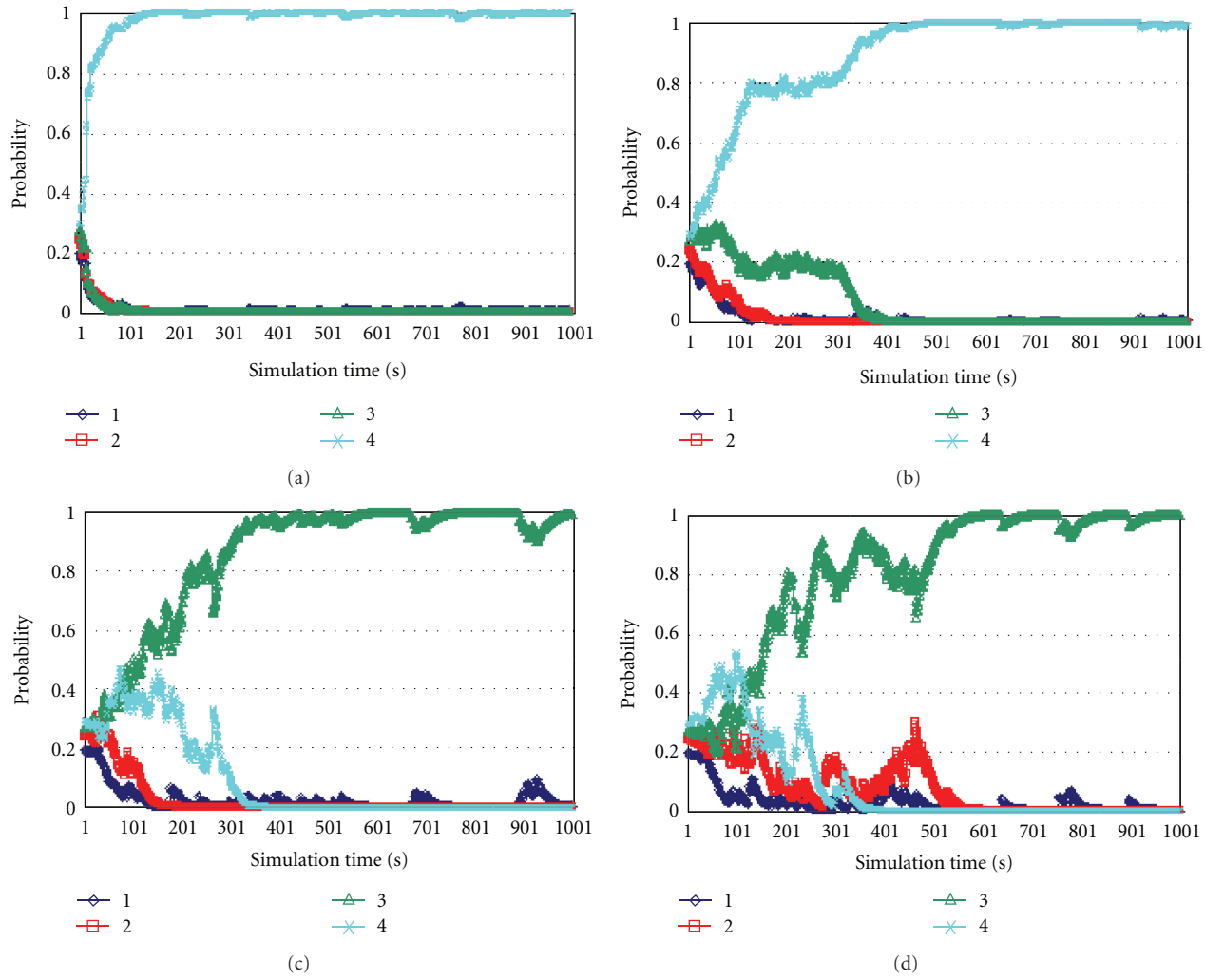


Figure 8: Variation in the probabilities of using certain transmission range of specific coronas.

5.3. Comparison with Other Algorithms. In this section, we evaluate the performance of the algorithm we proposed in this paper through comparing it with other existing algorithms. In the simulation, we compare ASTRL with two other algorithms.

(1) Optimal list (OL), namely, the optimal transmission range list. Since it is obtained by enumerating all available transmission range lists and selecting from them the list with maximal lifetime, it must be the optimal transmission range list compared to other transmission range lists of the network.

(2) t_x, the algorithm proposed in [8]. According to this algorithm, which is only capable of dealing with uniform node distribution, all the nodes in coronas adopt the maximal transmission range. We simulate the three algorithms with uniform node distribution and get their respective average network lifetime. In this simulation, in consideration of the requirements of t_x, the maximal transmission range of sensor nodes is set to be 5 m.

We experimentally compare the network lifetime with different network sizes (different coronas). As the results in Figure 9(a) show, in terms of the network lifetime, ASTRL is close to OL obtained by exhaustive search but significantly outperforms t_x. Figure 9(b) shows the average residual energy ratios of different algorithms, and the average residual energy ratio refers to the ratio of the remaining energy available when the network lifetime ends to the total initial energy of the network. As Figure 9(b) indicates, the performance of ASTRL is close to OL in terms of the average residual energy ratio. Compared with t_x, ASTRL has comparatively increased the energy efficiency so as to prolong the network lifetime.

The simulation is performed with 100 different uniform node distributions, and for each of them we run ASTRL 100 times. Figure 10 shows the ratios of the average network lifetime of ASTRL to that of OL in the networks with different uniform node distributions. As Figure 10 indicates the experimental results of ASTRL are close to the optimal network lifetime in the networks with whether 8 or 10 coronas. In addition, since there is only a minor fluctuation

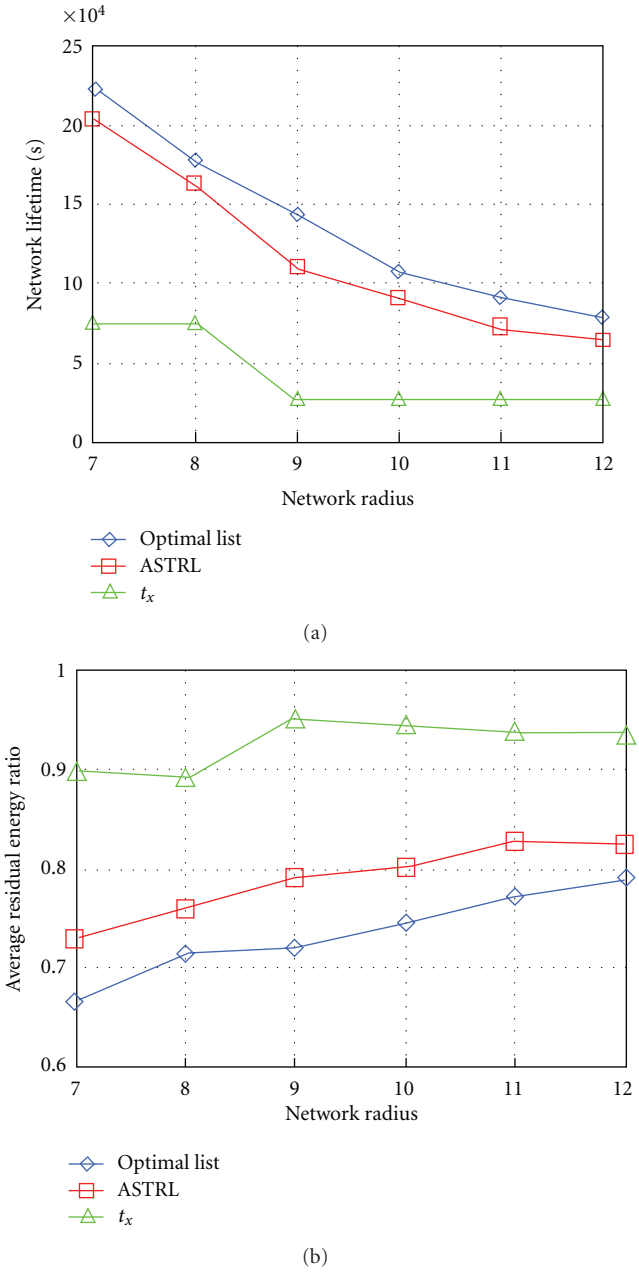


FIGURE 9: Network lifetime and average residual energy ratios of different algorithms.

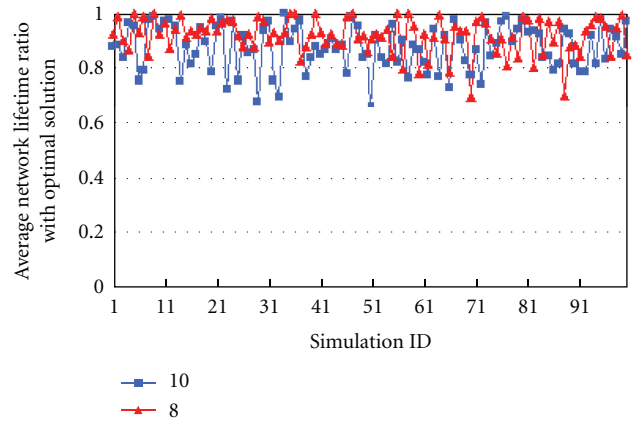


FIGURE 10: Network lifetime ratio with optimal solution in uniform random node distribution.

in the experimental results of ASTRL, 85% of which are larger than 0.8, we conclude that ASTRL can perform well in the networks with uniform random node distribution.

The simulations mentioned above are designed just for small-scale networks. This is because it is almost impossible to enumerate all potential transmission range lists in large-scale networks. For this reason, we have to compare ASTRL with t_x in a large-scale network (more than 20 coronas). Figure 11 shows how the network lifetime and the average residual energy ratios of the two algorithms vary with the increase of the number of the coronas in the network. Similar to Figures 9 and 11 indicates that even in large-scale networks ASTRL drastically outperforms t_x in terms of both network lifetime and average residual energy ratio.

ASTRL will take some optimizing time to determine the transmission range of each corona, and therefore we have to analyze the influence of the optimizing time of ASTRL on the lifetime of the whole network, and that will be done from the following four aspects. (1) After the optimizing time, the transmission range of the sensor nodes in each corona will be fixed without any change henceforth. (However, if the sensor network suffers from high node fault ratio, ASTRL can run again for adapting to the topology changes of the network. In this case, sink node can play the role of a decision-maker to decide whether to perform the optimizing process again or not.) (2) As shown in Figures 9(a) and 11(a), the ratio of the optimizing time to the lifetime of the whole network is very low. (3) Since the energy consumption for transmitting the ants can be ignored (see Section 3.2), the energy consumed by ASTRL itself is insignificant, which has little influence on the network lifetime accordingly. (4) As Figures 9(a) and 11(a) show, compared with t_x, ASTRL improves the network lifetime significantly at the cost of the energy consumption for the optimizing process. For this reason, we conclude that ASTRL outperforms t_x.

The authors in [18] propose to use a strategy of nonuniform node distribution to achieve maximum energy efficiency. According to the strategy, from the corona C_{R-1} to the innermost corona C_1, the number of the sensor nodes increases in a geometric ratio of $q > 1$, and in the corona C_R there are $N_{R-1}/(q-1)$ nodes. Besides the distribution strategy, they also propose a routing protocol called *q-Switch*, in which all the sensor nodes use the maximum transmission range. It is proved in [18] that the nonuniform node distribution strategy used in coordination with the routing protocol *q-Switch* will achieve the subbalanced energy depletion in the network and thus achieving the optimal network lifetime. Considering the decisive importance of the distribution strategy, we decide to evaluate the performance of ASTRL in large-scale networks with nonuniform node distribution by comparing it with *q-Switch*.

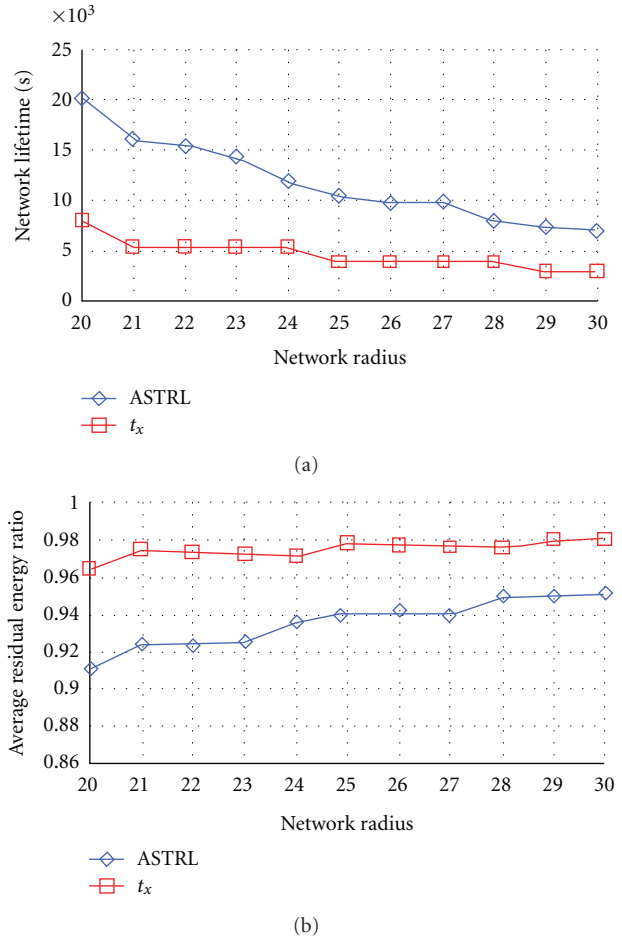


Figure 11: Network lifetime and average residual energy ratios of different algorithms.

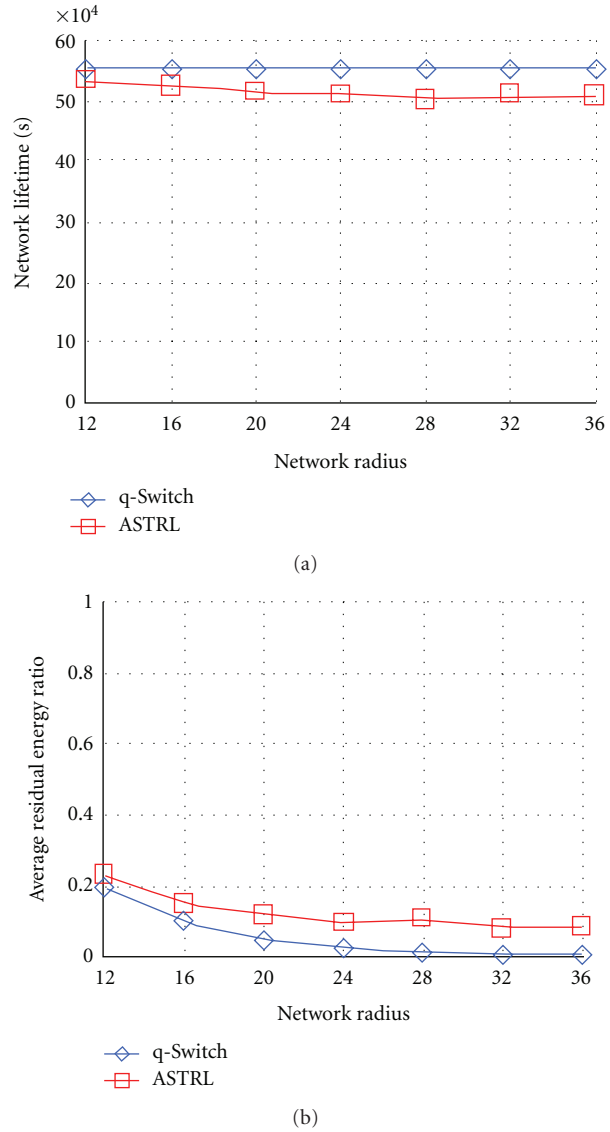


Figure 12: Network lifetime and average residual energy ratios with nonuniform deterministic node distribution.

Since the width of each corona in the corona model proposed in [18] is the maximum transmission range t_x while ASTRL is run on a kind of corona model with transmission range levels, to ensure a fair comparison, we have to covert the nonuniform node distribution strategy [18] into a corona model with transmission range levels. The corona with the width of t_x is divided into k subcoronas. From the outmost to the innermost every k subcoronas compose a group, in which each subcorona has the same number of nodes. The simulation parameters are as follows: $q = 2$, and there are 20 nodes in each of the four outmost coronas. Take a network with 12 coronas as an example. In the network, there are 20 nodes in each of the four coronas (from C_8 to C_5) and 40 in each of the other four coronas (from C_4 to C_1). The simulation results are shown in Figure 12. From Figure 12, it can be seen that the performance of ASTRL approaches the optimal values of *q-Switch* in terms of both network lifetime and average residual energy ratios. This indicates that ASTRL can adapt well to large-scale networks with nonuniform node distribution. Although *q-Switch* performs better, it will be determined by the nonuniform node distribution strategy, which, however, pays too much for node deployment. In contrast, ASTRL, as shown in Figures 11 and 12, is independent of the node distribution strategy and performs equally well in large-scale networks.

5.4. Effect of the Algorithm Parameters on the Performance. The simulation results in Figure 13 show the variation in the ratios of the average network lifetime of the ant colony algorithm with different evaporation coefficients to that of OL in the uniform node distribution. As Figure 13 shows, when the value of evaporation coefficient (denoted as γ) is 0.05, the algorithm has the best performance; when γ0.01, the evaporation coefficient of the pheromone is so low and the ants find it hard to search for other different routes during the search process and hence the poor search performance; when γ0.01, the evaporation coefficient is so high that the pheromone is unable to be accumulated quickly and hence affecting this positive feedback mechanism.

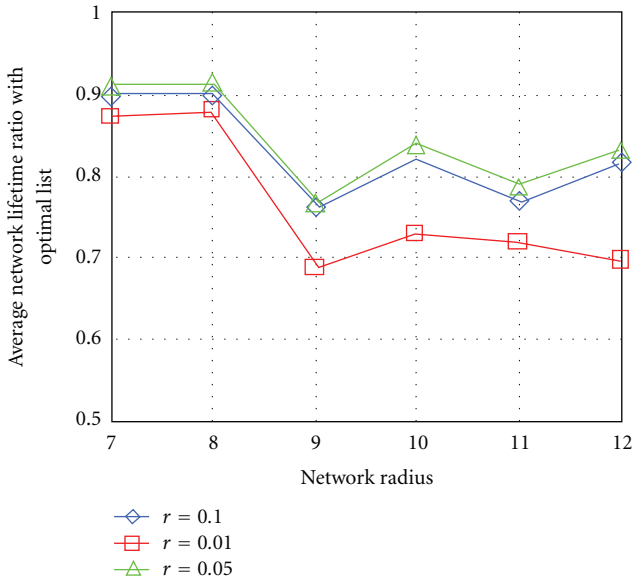


Figure 13: Average network lifetime ratio with optimal list in uniform node distribution.

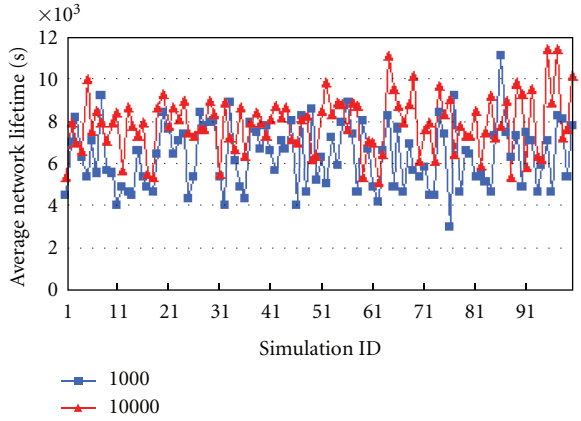


Figure 14: Network lifetime with different optimizing time.

We conduct simulations to evaluate the performance of a network with 30 coronas in uniform node distribution in terms of the network lifetime. In the simulation the parameter of optimizing time t_0 is set to be 102 s and 103 s, respectively, and for each of the parameter value we run the algorithm 100 times and use the average value of the results to evaluate the performance. As Figure 14 shows, the network lifetime is extended to some extent when t_0 increases, but there is not a major fluctuation in the simulation results when t_0 varies.

The ant generation rate refers to the number of the ants generated by each node during every time interval δt. Next we will discuss the relation between the ant generation rate and the convergence of ASTRL. We conduct simulations to examine how the minimum probability values in construction graph with different ant generation rates vary with the variation in the optimizing time in a network with 10

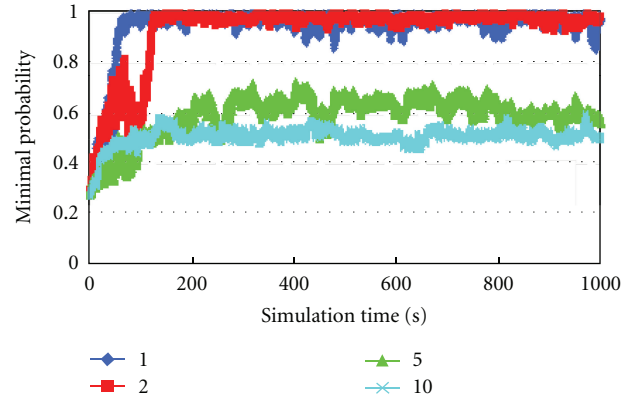


Figure 15: The changes of minimal probability with different ants generating rates.

coronas. In the simulation, the ant generation rates are set to be 1, 2, 5, and 10 ants, respectively, and each corona selects the maximum value of the probabilities of using different transmission ranges as its own maximum probability value; the smallest value of the maximum probability values of all the coronas is considered as the minimum probability value in construction graph. According to the simulation results in Figure 15, when the ant generation rate is 1, the algorithm achieves the best convergence, that is, the minimum probability value can converge approximately at 1 within the shortest time. In addition, we have noticed that the higher the ant generation rate is, the more feedback there is from the ants within the same time period. This kind of reciprocal impact makes the convergence of the algorithm decreased.

6. Conclusion

The energy hole problem is caused by uneven energy consumption. Because data transmission is achieved by forwarding scheme among coronas, the sensor nodes closest to the sink need to relay more traffic so as to deplete their energy budget faster than other sensors. Based on our analysis, the key factor of improving the network lifetime is that different transmission ranges should be adopted in different regions of the network. Therefore, in this paper, we start with finding the optimal transmission range list based on corona model and propose the ant-based distributed algorithm (ASTRL) which tends to prolong the network lifetime by assigning appropriate transmission range for each corona. The simulation results show our algorithm can significantly improve the network lifetime; actually, it is close to optimal list (OL) in terms of the network lifetime and can perform equally well in the nonuniform node distribution.

Acknowledgments

This work is supported by China National Science Foundation under Grants no. 60903158, 61170256, 61173172, and 61103226 and the Fundamental Research Funds of China

for the Central Universities under Grants no. ZYGX2010J074 and ZYGX2011J102.

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. 102–114, 2002.

[2] 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.

[3] G. Chen, C. Li, M. Ye, and J. Wu, "An unequal cluster-based routing protocol in wireless sensor networks," *Wireless Networks*, vol. 15, no. 2, pp. 193–207, 2009.

[4] J. Jia, X. Wu, J. Chen, and X. Wang, "Exploiting sensor redistribution for eliminating the energy hole problem in mobile sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, article 68, 2012.

[5] N. Bartolini, T. Calamoneri, A. Massini, and S. Silvestri, "On adaptive density deployment to mitigate the sink-hole problem in mobile sensor networks," *Mobile Networks and Applications*, vol. 16, no. 1, pp. 134–145, 2011.

[6] C. Song, M. Liu, J. Cao, Y. Zheng, H. Gong, and G. Chen, "Maximizing network lifetime based on transmission range adjustment in wireless sensor networks," *Computer Communications*, vol. 32, no. 11, pp. 1316–1325, 2009.

[7] D. Kandris, P. Tsioumas, A. Tzes, G. Nikolakopoulos, and D. D. Vergados, "Power conservation through energy efficient routing in wireless sensor networks," *Sensors*, vol. 9, no. 9, pp. 7320–7342, 2009.

[8] S. Olariu and I. Stojmenović, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, April 2006.

[9] J. Lian, K. Naik, and G. B. Agnew, "Data capacity improvement of wireless sensor networks using non-uniform sensor distribution," *International Journal of Distributed Sensor Networks*, vol. 2, no. 2, pp. 121–145, 2006.

[10] A. Wadaa, S. Olariu, L. Wilson, K. Jones, and M. Eltoweissy, "Training a sensor networks," in *Proceedings of the MONET*, January 2005.

[11] V. Mhatre and C. Rosenberg, "Design guidelines for wireless sensor networks: communication, clustering and aggregation," *Ad Hoc Networks*, vol. 2, no. 1, pp. 45–63, 2004.

[12] A. Jarry, P. Leone, O. Powell, and J. Rolim, "An optimal data propagation algorithm for maximizing the lifespan of sensor networks," in *Proceeding of the 2nd IEEE International Conference on distributed Computing in Sensor Systems (DCOSS '06)*, pp. 405–421, San Francisco, Calif, USA, 2006.

[13] G. Nico and M. Tesauro, "On the connectivity of ad hoc networks: the role of the power level distribution," in *Proceedings of the 15th International Conference on Software, Telecommunications and Computer Networks (SoftCOM '07)*, pp. 160–164, September 2007.

[14] F. Dressler, "Benefits of bio-inspired technologies for networked embedded systems: an overview," in *Proceedings of the Dagstuhl Seminar 06031 on Organic Computing-Controlled Emergence*, Schloss Dagstuhl, Wadern, Germany, January 2006.

[15] F. Dressler, "Efficient and scalable communication in autonomous networking using bio-inspired mechanisms—an overview," *Informatica*, vol. 29, no. 2, pp. 183–188, 2005.

[16] I. Wokoma, L. Sacks, and I. Marshall, "Biologically inspired models for sensor network," http://www.cs.kent.ac.uk/pubs/2002/1883, 2002.

[17] J. Li and P. Mohapatra, "Analytical modeling and mitigation techniques for the energy hole problem in sensor networks," *Pervasive and Mobile Computing*, vol. 3, no. 3, pp. 233–254, 2007.

[18] X. Wu, G. Chen, and S. K. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 5, pp. 710–720, 2008.

[19] G. Di Caro and M. Dorigo, "AntNet: Distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.

[20] R. G. Aghaei, M. A. Rahman, W. Gueaieb, and A. El Saddik, "Ant colony-based reinforcement learning algorithm for routing in wireless sensor networks," in *Proceedings of the 2007 IEEE Instrumentation and Measurement Technology on Synergy of Science and Technology in Instrumentation and Measurement (IMTC '07)*, May 2007.

[21] S. Okdem and D. Karaboga, "Routing in wireless sensor networks using ant colony optimization," in *Proceedings of the 1st NASA/ESA Conference on Adaptive Hardware and Systems (AHS '06)*, pp. 401–404, June 2006.

[22] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[23] F. Ingelrest, D. Simplot-Ryl, and I. Stojmenović, "Target transmission radius over LMST for energy-efficient broadcast protocol in ad hoc networks," in *Proceedings of the IEEE International Conference on Communications (ICC '04)*, pp. 4044–4049, Paris, france, June 2004.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 273792, 9 pages
doi:10.1155/2012/273792

Research Article

Application-Oriented Fault Detection and Recovery Algorithm for Wireless Sensor and Actor Networks

Jinglin Du,[1,2] Li Xie,[1] Xiaoyan Sun,[2] and Ruoqin Zheng[2]

[1] *Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China*
[2] *Department of Electronics and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China*

Correspondence should be addressed to Xiaoyan Sun, sun.xiaoyan.1989@163.com

Received 30 July 2012; Accepted 17 September 2012

Academic Editor: Wenzhong Li

Recent years have witnessed a growing interest in applications of wireless sensor and actor networks (WSANs). In WSANs, maintaining interactor connectivity is of vital concern in order to reach application level. Failure of a critical actor may partition the inter-actor network into disjoint segments. This paper proposed an application-oriented fault detection and recovery algorithm (AFDR), a novel distributed algorithm to reestablish connectivity. AFDR identifies critical actors and designates backups for them. A backup actor detects the critical node failure and initiates a recovery process via moving to the optimal position. The purpose of AFDR is to satisfy application requirements, reduce recovery overhead, and limit the impact of critical node failure on coverage and connectivity to the utmost. The effectiveness of AFDR is validated through simulation experiments.

1. Introduction

One of the most attractive and important parts of The Internet of Things is known as the wireless sensor and actor networks (WSANs) [1]. WSANs are finding applications in many domains such as disaster management, homeland security, battlefield reconnaissance, space exploration, search and rescue, and so forth. A WSAN consists of numerous miniaturized sensor nodes and fewer actor nodes [2]. The sensor nodes probe their surroundings, measure ambient conditions, and transmit the collected data to one or multiple actors for processing, making decisions, and responding to events of interest.

In WSANs, connectivity of the network is crucial throughout the lifetime of the network in order to meet the desired application-level requirements. As far as WSANs are concerned, in most application setups, actors need to coordinate with each other in order to share and process the sensors' data and plan an optimal response. In such connected WSANs, failure of one or multiple nodes may cause the loss of other nodes or communication links, partitioning of the network if alternate paths among the affected nodes are not available, and stopping the actuation capabilities of the node. Such a scenario will not only hinder the nodes' collaboration but also has very negative consequences on the considered applications. Therefore, WSANs should be able to tolerate the failure of mobile nodes and self-recover from them in a distributed, timely, and energy efficient manner: first, the recovery should be distributed since these networks usually operate autonomously and unattended. Second, rapid recovery is desirable in order to maintain the responsiveness to detected events. And finally, the energy overhead of the recovery process should be minimized to extend the lifetime of the network [3].

In this paper, we present an application-oriented fault detection and recovery algorithm (AFDR) to determine possible partitioning in advance and self-restore the connectivity in case of such failures with minimized node movement and message overhead. Since partitioning is caused by the failure of a node which is serving as a cut vertex (critical node) [4], each node determines whether it is a cut vertex or not in a distributed manner (Arrival Message Matrix (AMM)). Each node in the network periodically sends out heartbeat messages and decides whether it is a cut vertex based on the received feedback. Once such cut vertex nodes are determined, each node designates the appropriate neighbor called backup to handle its failure. The backup detects any failure of the cut vertex and chooses a suitable node (a leaf

node or itself) to replace the failed node. The alternative node need not move to the exact position directly where the failed actor is; it just moves to an optimal position to administer more sensors. The goal of AFDR is to ensure the coverage of actors and reduce the mobility and communication overhead caused in the connectivity recovery process.

2. System Model and Problem Statement

AFDR is applicable to WSANs that consist of sensors and actors. Sensors detect and report events of interest to one or multiple actors. Actors receive reports from sensors and process and collaborate with each other to plan an optimal coordinated response. The communication range of an actor refers to the maximum Euclidean distance that its radio can reach. An actor has two radios for sensor-actor and actor-actor communications. To simplify analysis, nodes are assumed to have the same communication range. Both sensors and actors are deployed randomly in an area of interest. After deployment, actors are assumed to discover each other and form a connected inter-actor network [3]. An actor is assumed to be able to move on demand and before moving it informs its backup so that it may not be wrongly perceived as faulty.

In order to provide qualified services, many applications require overlays [5] to guarantee reliability and avoid network failures [6]. For a large-scale distributed system, it is often not cost-effective to protect all the nodes [7]. While node failures may affect network coverage [8] and connectivity, this paper focuses on both, especially on maintaining the latter when a node is lost. The impact of an actor's failure depends on the position of that actor in the network topology. For example, losing a noncritical node does not affect inter-actor connectivity. Meanwhile, the failure of critical node will partition the network into disjoint segments. AFDR pursues actor relocation to recover from critical node failures. We consider one failure at a time and assume that no node fails during the recovery of another.

3. Related Work

In most WSAN applications, actors establish a connected inter-actor topology in order to coordinate with each other on an optimal response and synchronize their operations. However, the harsh environment which WSAN operates in makes actors vulnerable to physical damage and component malfunction. An actor failure may partition the inter-actor network into disjoint segments and consequently hinders inter-actor interaction. Numerous schemes have been pursued recently for repairing network connectivity in partitioned WSANs.

The main idea is to identify and relocate some of the nodes. Employing node mobility to repair damaged network topologies has started to attract attention. Most of the existing approaches in the literature are purely reactive [9], which can be categorized into block [10] and cascaded movements. Block movement often requires a high prefailure connectivity in order for the nodes to coordinate their response. It often becomes infeasible in the absence of higher level of connectivity. Therefore, cascaded node movement that can be further categorized based on network state information that nodes are assumed to maintain is exploited. Some approaches like DARA [2] and PADRA [3] require each actor to maintain two-hop neighbors. One of the neighbors of the failed node is picked to initiate the recovery process such that the movement overhead and the number of messages are minimized. While DARA designates the node with the least node degree as the recovery initiator and strives to restore connectivity lost due to failure of cut vertex, PADRA identifies a connected dominating set to determine a dominatee node in order to detect cut vertices. The dominatee does not directly move to the location of the failed node. Nonetheless, they use distributed algorithm and their solution still requires 2-hop neighbor's information that increases messaging overhead.

Some localized algorithms require only 1-hop neighbor's positional information at the expense of lower accuracy of cut vertices identification. Basically, some nodes are marked as critical while they are not cut vertices. However, no critical node will be missed. DCR [11] and RAM [12] employ a simple localized cut vertex detection procedure that runs on each node in a distributed manner to determine locally whether a node is critical or not.

Akkaya et al. introduced a mutual exclusion mechanism called MPADRA [13] in a localized manner. Both RAM and MPADRA can handle simultaneous failure of multiple actors but MPADRA differs from RAM in multiple aspects. First, MPADRA requires a mutual exclusion mechanism to avoid race conditions. Second, MPADRA reserves the nodes on the path in advance before actual relocation. On the other hand, RAM designates distinct backups and does not engage relocating nodes beforehand. Third, MPADRA maintains 2-hop network state information and requires primary and secondary failure handlers for each dominator.

Younis et al. proposed an algorithm called RIM [14]. When a node fails, its neighbors move inward toward its position so they can connect with each other. The rationale is that these neighbors are the ones directly impacted by the failure, and when they can reach each other again, the network connectivity is restored to its prefailure status. The relocation procedure is recursively applied to handle any nodes that get disconnected when one of their neighbors moves.

Another algorithm VCR [15] exploits the fact that some neighbors of the failed node are not using their full communication range and would thus be able to reach more distant nodes than the failed actor. In VCR, each actor maintains a list of 1-hop neighbors and monitors their heartbeats. The failure of an actor is detected through missing heartbeats. The recovery process consists of two phases. At first, volunteer actors are identified. Then the topology repair is performed through uncoordinated relocation of the volunteer actors while exploiting partially utilized transmission range and actor diffusion.

4. Connectivity Restoration Algorithm

4.1. Cut Vertex Identification. Every node in the system considers itself as a cut vertex candidate and initializes a cut

vertex detection process. The node that has zero or only one connection does not need to initialize this process since it can not be a cut vertex. At the beginning, the candidate assigns a unique numerical identifier for its connections which is called the connection number. For example, if a candidate has c connections, it will label them from 1 to c.

At first, the candidate sends a component probe message to each of its neighbors. The message contains the following elements: the candidate's ID, a timestamp, a TTL threshold, and the number of connections that connects this neighbor with the candidate. So each node in the system has a connection list. The format of the entry for a candidate in the connection list is "candidate ID, timestamp, TTL threshold, connection number 1, connection number 2, ...". Upon receiving a message, each node deals with it when one of the following situations happens.

(a) If the node has already received the message, or the message is old, the node simply just drops the message.

(b) If there is no entry for the candidate that issues this message, the node creates an entry for it.

(c) If the timestamp in the received message is newer than the one stored in the connection list, the node replaces the old timestamp and connection numbers stored in the connection list with the new ones.

(d) If the timestamp in the received message is the same as the one stored in the connection list but the connection number of the message differs from the old one, the node adds the new connection number to the entry and sends an arrival message back to the candidate.

So each candidate maintains an arrival list whose format is similar to the connection list: "node ID, timestamp, connection number 1, connection number 2, ...". That is to say, a node does not send any arrival messages until its probe messages contain at least two different connection numbers [3].

According to the arrival list, each candidate maintains a c-by-c binary matrix, where c is the number of connections the candidate has. The row or column numbers of the binary matrix represent the connection number and the matrix is called the candidate's AMM (Arrival Message Matrix). Even though the node with high degree has a big matrix, the storage overhead is so small when compared to the movement overhead in the recovery process that it can be ignored. For an entry (x, y) in the matrix, where x is the row number and y is the column number, if the corresponding connection number of x and y can be found in the same entry of the arrival list, the value of this matrix entry is set to 1. Otherwise, the value is set to 0. In other words, if any node has sent back an arrival message containing connection numbers x and y to the candidate, 1 is set in the (x, y) and (y, x) entries of the candidate's AMM. After waiting a defined period of time, the candidate forms an undirected graph coming from the matrix representation. The vertices of this graph are corresponding to the candidate's connections. We call this graph the candidate's AMM graph. Node x and node y are connected in the AMM graph if and only if the value of the AMM entry (x, y) is 1. If the candidate's AMM graph has more than 1 component (a connected graph), then we do a local search among the two-hop neighbors of the candidate. When this candidate is removed from the system, the rest of the actors still stay connected and the candidate is not a cut vertex. Instead, we determine that the candidate is a cut vertex.

4.2. Backup Selection and Failure Detection. Once the cut vertices are identified, the next step is to select and designate appropriate neighbors as backups for them. The purpose of the prenomination of backup nodes is to instantaneously react to the failure of cut vertex and avoid the possible network partitioning caused by such a failure.

4.2.1. Selection of Backup. The actors maintain minimum state information (i.e., 1-hop neighbors) to avoid extra overhead of messaging. Since neighbors become disconnected when a critical actor fails, backup actors are determined and notified before a failure of critical nodes takes place. A node can serve as backup for multiple actors. The selection of a backup among 1-hop neighbors is based on the following ordered criteria.

Neighbor Position (NP). As discussed above, each actor determines whether it is critical or noncritical depending on the position of that node in the topology. A noncritical neighbor actor is more suitable for backup because it will limit the scope of recovery that ultimately reduces the impact on application, coverage, connectivity, and incurred overhead.

Actor Flow (AF). If a noncritical node is available in the 1-hop neighborhood of the primary actor, we choose the actor with the smallest flow package from sensors charged by it and other actors; in other words, the one which receives the least data packets from other nodes, as the cut vertex's backup.

Flow package is divided into two categories.

(i) Actor-actor flow package (AAFP): this flow package is transferred between an actor and another actor.

(ii) Actor-sensor flow package (ASFP): this flow package is transferred between an actor and a sensor. If the number of these packages for an actor is large, it indicates that sensors under the jurisdiction of the actor is more, so the actor can play a greater role for the application system.

Actor Degree (AD). The impact of moving a node that has many neighbors will be significant. Thus, if a noncritical node is not available in the neighborhood, AFDR favors replacing the failed actor with the neighbor that is a strongly connected critical node (with high degree) because there is more probability to have noncritical nodes in the neighborhood. This will limit the scope of cascaded relocation and thus lower the recovery overhead.

Take the circumstance in Figure 1 as an example. Green pentagons represent the actors while yellow circles stand for sensors. Among the 1-hop neighbors of actor 6, actor 7 is

not a critical node, so it is best suited to be the backup. If we remove actor 7, in the rest of the neighbors of actor 6, actor 8 and actor 9 will have the same AD, which will be smaller than actor 3. At this point, we continue to compare their AF. The one whose AF is smaller will be chosen as a backup of actor 6. Figure 2 shows the pseudocode of backup selection of AFDR algorithm.

4.2.2. Failure Detection. Neighbor actors exchange heartbeat messages as part of their network operation to update their status. The chosen backup actors are notified via these messages. Once an actor receives backup notification, it starts monitoring the cut vertex through heartbeats. Missing a number of consecutive heartbeats is perceived by backup as a failure of the cut vertex.

4.3. Failure Recovery. Despite the fact that failure of a node which is not a cut vertex will not cause any problems to the inter-actor connectivity, it can create other problems such as forming coverage holes, disrupting the data collection from that particular region, and so forth. In such cases, depending on the application-level requirements, these problems need to be handled. We would like to note, however, that handling such problems is out of scope of this paper. We only focus on restoration of inter-actor connectivity when a cut vertex node fails.

4.3.1. Definition: Optimal Position (OP). AFDR defines an optimal position, when the actor failure, its replacement node does not need to move to the original location where the failure node is, the new location of this alternative node has the following characteristics: it can communicate with the surrounding sensors as many as possible, that is, the more ASFP it received, the better; the distance it moved from its original location to the optimal position should be as short as possible to reduce overhead. If this location meets the above conditions, we call it the optimal position.

4.3.2. OP Computation Model. We proposed a model to calculate the OP. For the convenience of description, we take the following circumstance as an example. In Figure 3, actor 3 serves as a backup for the node F. When F fails, it partitions the network into two parts. Actor 1 and actor 2 are respectively, the nearest nodes to F in the two subnets. Communication range of actor 1 is shown as circle A and simultaneously, circle B represents the communication range of actor 2. These two circles' intersections are M and N. C is the overlap area of actor 1 and actor 2's communication range. According to the location of the backup, we determine the optimal position.

(i) If actor 3 is located in A but not in C, do a connection between actor 3 and actor 2; the intersection of the line and circle B is the OP. If actor 3 is located in B but not in C, just use the same method to pursue the OP.

(ii) If actor 3 is located in C, the location of actor 3 itself is OP.

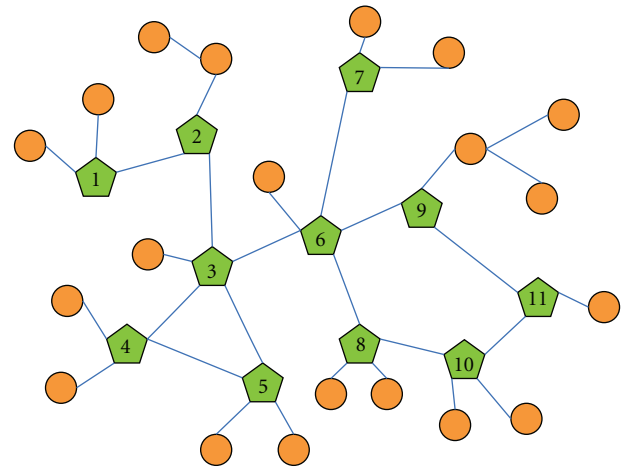


Figure 1: A connected interactor network with critical and noncritical actors.

```
F: Primary Node
/* Backup Selection */
If (isCritical(A) == false && isCritical(D) == true)
    EnableBackUp (A)
endif
If (isCritical(A) == false && isCritical(D) == false) then
    if(ActorFlow(A)<= ActorFlow (D)) then
        EnableBackUp (A)
    else
        EnableBackUp (D)
    endif
endif
If (isCritical(A) == true && isCritical(D) == true) then
    if(ActorDegree(A)>= ActorDegree (D)) then
        EnableBackUp (A)
    else
        EnableBackUp (D)
    endif
endif
```

Figure 2: Pseudocode of backup selection of AFDR algorithm.

(iii) If actor 3 is located neither in circle A nor circle B, either M or N will be the OP. Choose the one that is closer to actor 3.

The predesignated backup actor immediately initiates a recovery process once it detects the failure of the cut vertex. There are four scenarios that may happen. First, if the backup is noncritical then it simply replaces the cut vertex and moves to the OP so that recovery is complete. Second, if the backup actor is critical and has a leaf node in its neighborhood, the leaf node just replaces the cut vertex and moves to the OP. Third, if the backup actor is critical then it checks whether the failed node was also its backup or not; when

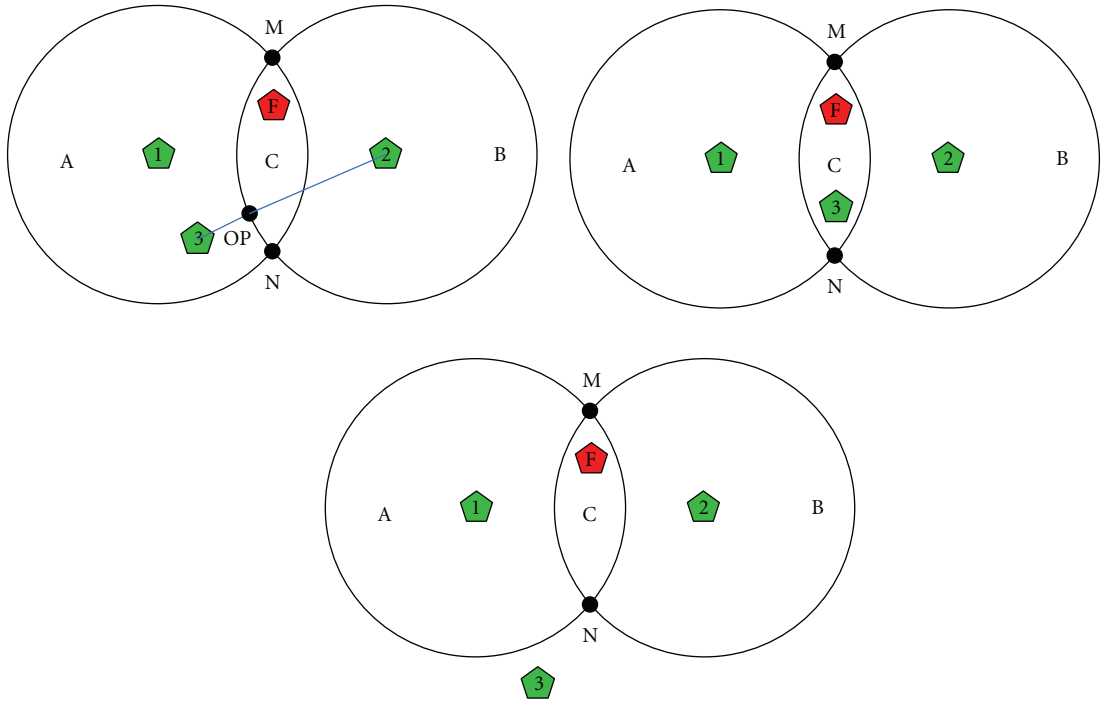


Figure 3: OP calculation.

the two nodes are backup for each other, the backup actor appoints another backup using the same criteria as specified in the preceding section and moves to the OP. Finally, if the predesignated backup actor is critical and its backup is alive then it just sends a movement notification message to the backup and moves to the OP. Since, in scenario 3 and scenario 4, moving a critical backup actor further partitions the network, a cascaded relocation may be triggered.

In Figure 4, the two circles, respectively, represent the communication range of actor 3 and actor 9. Obviously, actor 7 is the backup of actor 6. When actor 6 fails, based on the above OP calculation method, we can get the optimal position for actor 7. As long as actor 7 is a leaf node, it directly moves to the OP to ensure the connectivity of this network. Figure 5 indicates the location of actor 7 after the recovery progress is finished.

Figure 6 shows the pseudocode of the recovery process of AFDR algorithm.

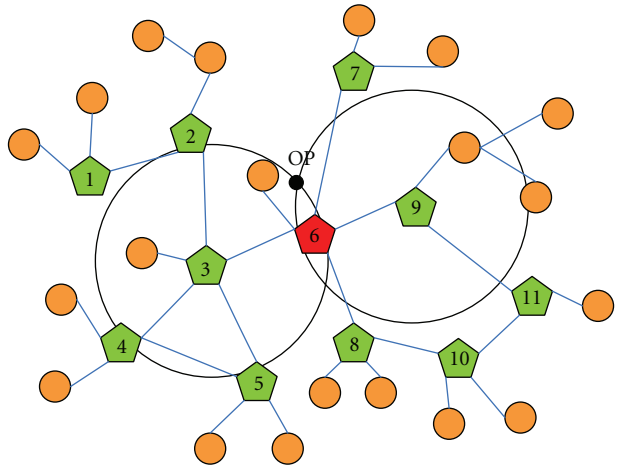


Figure 4: Original topology when an actor fails.

5. Algorithm Analysis

5.1. Simulation Setup and Performance Metrics. In the simulation experiments, we have created inter-actor topologies consisting of a varying number of nodes (20–100). Nodes are randomly placed in an area of 1000 m × 600 m with no obstacles that hinder a node from moving to a new position. We have varied the transmission range of actors from 50 to 125 m so that the topology becomes strongly connected. The following parameters were used to vary the WSAN configuration in the experiments.

(i) The number of deployed nodes (N) in the network affects the node density and the inter-actor connectivity.

(ii) The node communication range (r) influences the network connectivity and highly affects the recovery overhead in terms of the traveled distance and the number of involved actors.

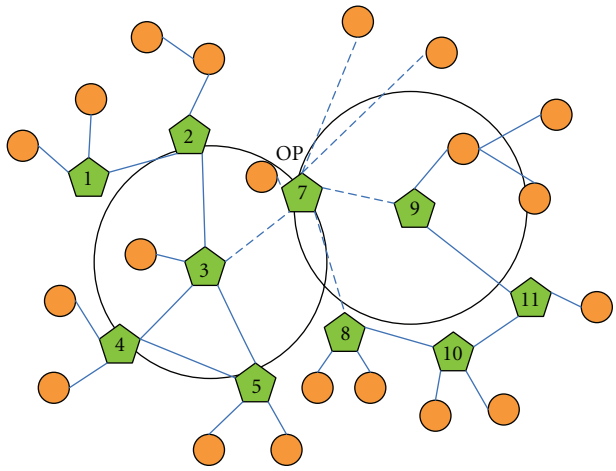


Figure 5: Recovery process when an actor fails.

We assessed the performance of AFDR by using the following metrics.

(i) The total traveled distance: it gives the total distance caused by all node movements involved in the recovery. This metric gauges how much energy will be consumed by the whole network due to the mechanical movements of nodes.

(ii) The number of messages exchanged among nodes: this metric also indicates the energy dissipation and recovery overhead.

(iii) The percentage of coverage reduction: although connectivity is the main objective of a recovery algorithm, node coverage is important for many setups. The loss of a node usually has a negative impact on coverage. This metric captures the loss of coverage resulting from the node movements.

(iv) Average node degree: it measures the level of interactor connectivity and availability of alternative paths after the recovery is complete.

The performance of AFDR is compared to DARA and RIM. Both DARA and RIM are reactive approaches and do not provision for recovery ahead of time. Like AFDR, DARA and RIM are distributed algorithms and exploit node relocation to recover from node failure. However, they have differences in the procedure. When a node fails, DARA selects a best candidate among its 1-hop neighbors and replaces it. This algorithm uses a recursive method to tolerate connectivity loss due to movement, that is, the chosen candidate will be replaced with one of its neighbors and so on. On the other hand, RIM moves all the 1-hop neighbors towards the failed node until they become connected again. As a result, when growing the communication range, the performance of RIM significantly worsens.

5.2. Results and Analysis. The number of nodes has been set to 20, 40, 60, 80, and 100. The communication range of actors is changed among 50, 75, 100, and 125. When changing the node count, "r" is fixed at 100 m; when changing the communication range, "N" is set to 100. The results of individual experiments are averaged over 50 trials.

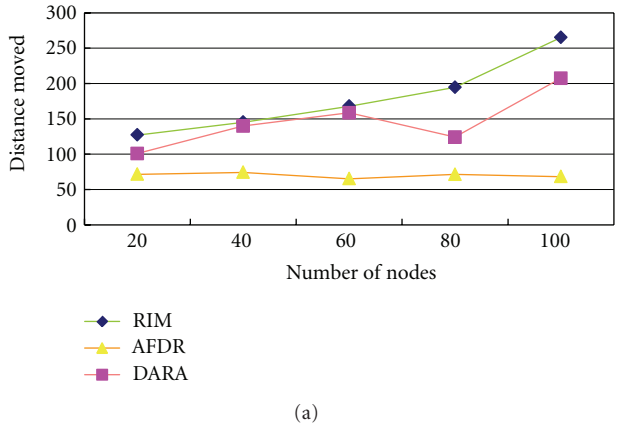
```
AFDR (A)
/* Recovery Process */
If(EnableBackUp(A)==true) then
  if (Primary actor F fails) then
    if(isCritical(A) == false) then
        MoveToOP(F, A)
    else if(HasLeafNode(A,B)==true) then
        MoveToOP(F, B)
        else if(SimPrimBackUp(F, A) == true) then
            AssignBackUp(A)
            NotifyBackUp(A)
            MoveToOP(F, A)
        else
            NotifyBackUp(A)
            MoveToOP(F, A)
        endif
    endif
  endif
endif
MoveToOP (F, A)
Move A to the optimal position of F
HasLeafNode(A, B)
B is the leaf node of A
SimPrimBackUp(F, A)
F and A are backup for each other
```

Figure 6: Pseudocode of the recovery process of AFDR algorithm.

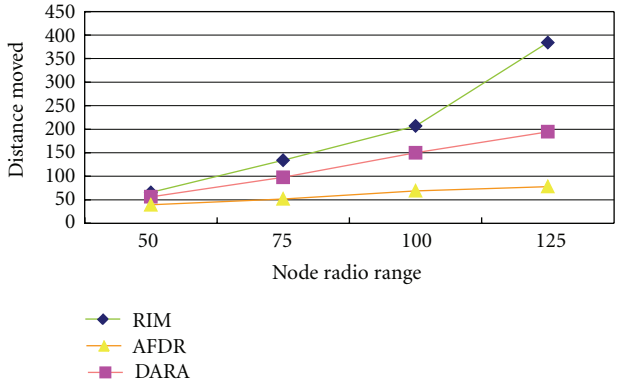
5.2.1. Total Traveled Distance. Figure 7 shows the distance traveled by all nodes until the connectivity is restored. AFDR obviously outperforms the existing approaches DARA and RIM.

As these two graphs in the figure indicate, the total node traveled distance of AFDR has not changed drastically even with higher node density and communication range. This is because AFDR strives to avoid moving critical nodes that causes further partitioning and requires cascaded relocations. It performs cascaded relocations only when noncritical nodes in the neighborhood of the failed actor are not available. RIM moves all the 1-hop neighbors towards the failed node until connectivity is reestablished. That is why its curve in the graph is steep.

5.2.2. Number of Messages Exchanged. Figure 8 presents the communication overhead when the network size and radio range take different values. As the figure indicates, AFDR begets less messaging overhead than DARA and RIM and

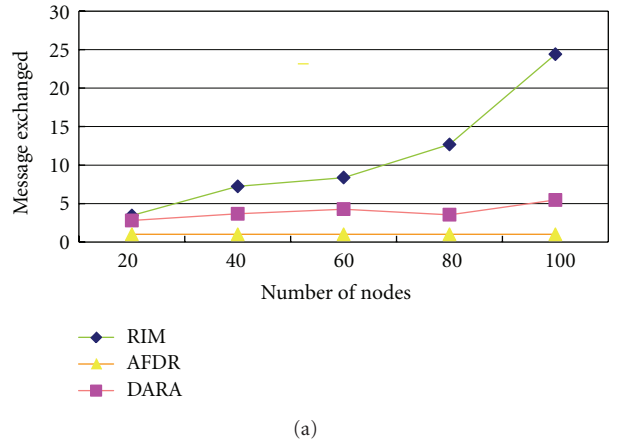


(a)

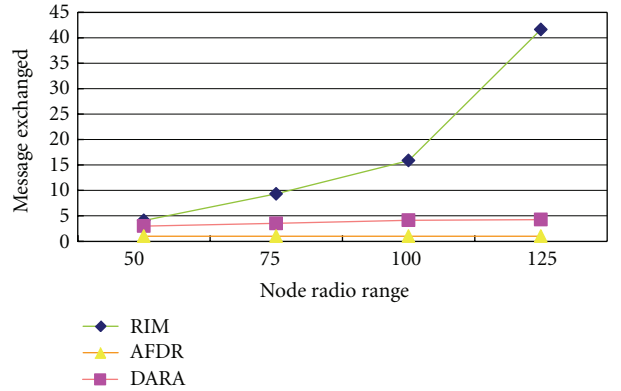


(b)

Figure 7



(a)



(b)

Figure 8

remains unchanged. Unlike DARA and RIM, AFDR strives to involve noncritical nodes in the recovery which limits the need of cascaded movement and thus reduces the number of notification messages. Therefore, AFDR makes best use of these messages, since most of the backup nodes are noncritical and they are not required to send any message. The average number of exchanged messages sent by AFDR in Figures 8(a) and 8(b) are less than 1. On the other hand, Figure 8 shows that the communication overhead in RIM grows rapidly for a higher actor density and long radio range.

5.2.3. Coverage Reduction Percentage. Figure 9 shows the impact of node failure on coverage, measured in terms of percentage of coverage reduction while changing N and r. Figure 9(a) indicates that the difference of curve trend among the three approaches is not too much as increasing the node number. Although increasing the node density helps, DARA and RIM still do not match AFDR's performance.

The advantage of AFDR in terms of coverage is obviously due to the limited scope of node relocation, which causes a coverage loss at the network periphery. In Figure 9(b), the advantage of AFDR highlights out as increasing the radio range. However, the performance of RIM worsens when growing the radio range above 100 m. With the increased value of r, the network becomes more connected and the number of neighbors of the failed node grows. RIM moves nodes inwards making the area around the failed node to be more crowded, thus leaves uncovered parts at the network periphery and causes a great loss of coverage.

5.2.4. Average Node Degree. Figure 10 shows the average actor degree and coverage in the network after connectivity is restored by all approaches. As showed in the figure, the data in the four graphs are basically the same. Figures 10(a)–10(d) confirm that AFDR does not have disadvantages in maintaining network connectivity.

Through the above analysis, we find that AFDR has advantages in reducing movement and communication overhead and keeping connectivity, which can make the life expectancy of the entire network longer. In other words, it will be easier to meet the application-level requirements.

6. Conclusion

In this paper, we have presented an application-oriented fault detection and recovery algorithm that focus on application-level concerns while recovering from critical node failure. The issued AFDR uses AMM to identify critical actors and

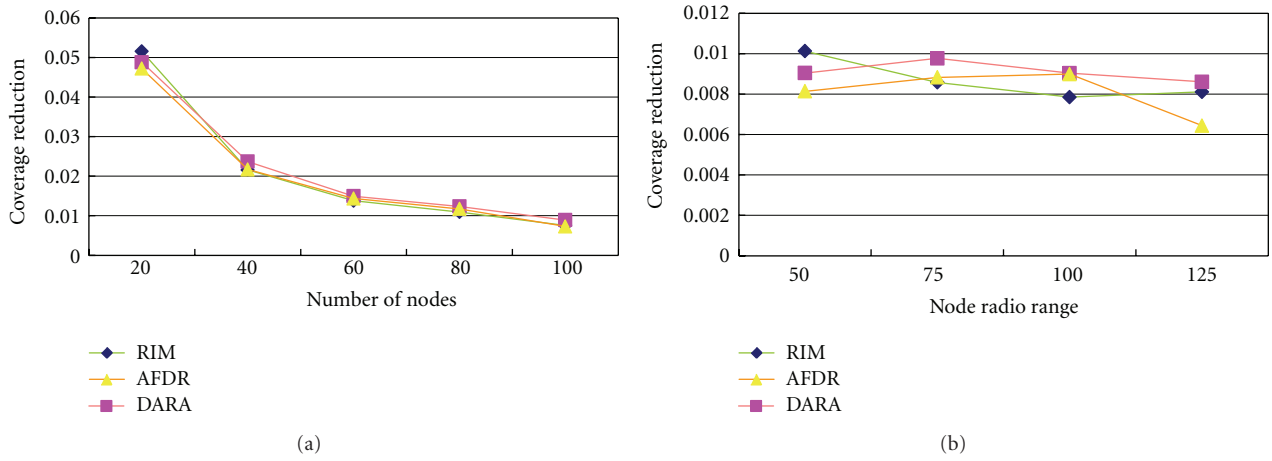


Figure 9

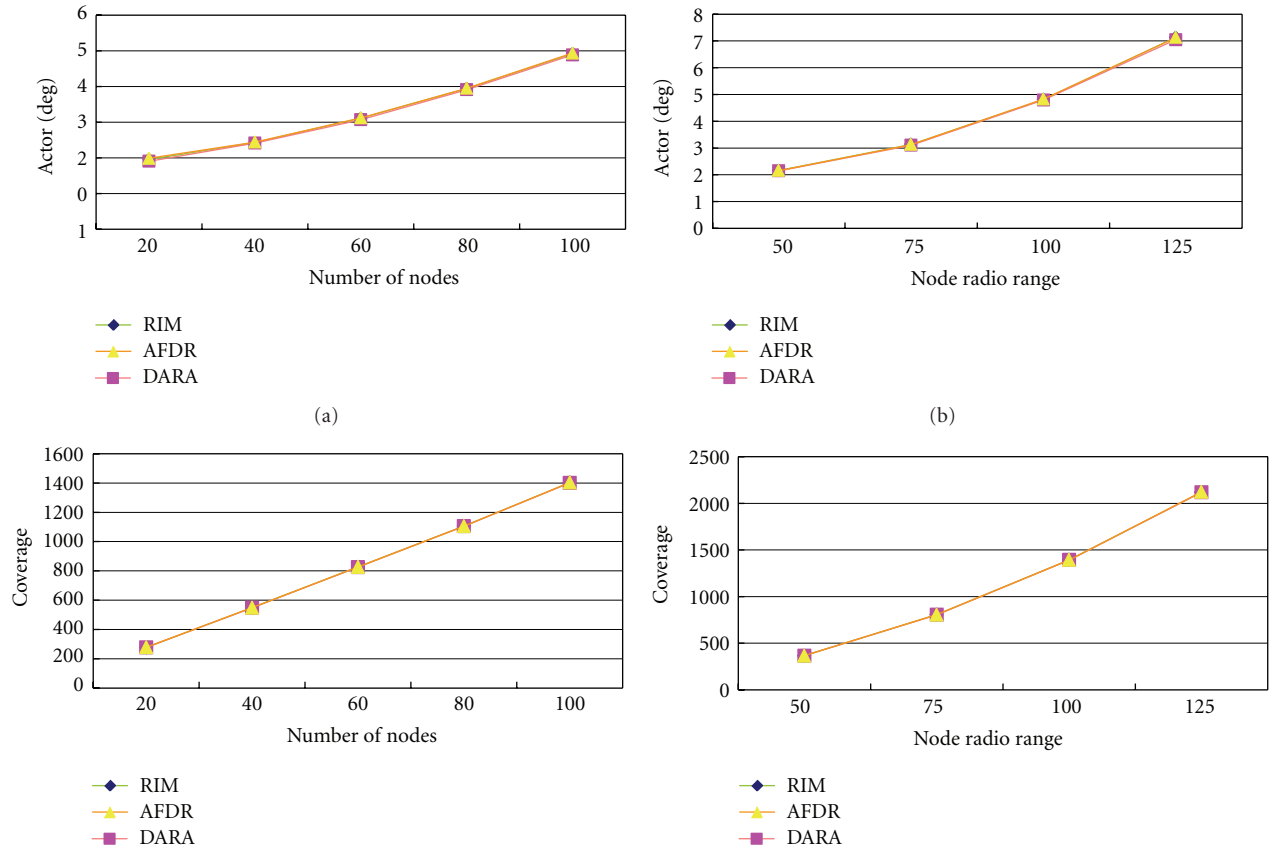


Figure 10

designates backup for them according to actor flow and actor degree. It tries to choose noncritical neighbors as backups to reduce the movement overhead and highly connected nodes as backups in order to limit the scope of cascaded relocation and nodes receiving smallest flow package as backups to minimize the communication overhead. In the recovery process, it finds an optimal position for backups to restore the inter-actor connectivity. The simulation results have proved the effectiveness of AFDR compared to contemporary recovery approaches in terms of satisfying application requirements, reducing recovery overhead and limiting the impact of critical node failure on coverage and connectivity.

In the future, our study can focus on how to identify cut vertex from a large amount of nodes in WSANs more quickly.

Acknowledgments

This work was supported by the 973 Program of China (nos. 2006CB303000 and 2009CB320-705), the National Natural Science Foundation of China (nos. 60873026 and 61021062), the National Science and Technology Support Program of China (no. 2012BAK26B02), the Science and Technology Support Program of Jiangsu Province (nos. BE2010178 and BE2011195), and the Industrialization of Science Program for University of Jiangsu Province (no. JH10-3).

References

[1] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, no. 4, pp. 351–367, 2004.

[2] A. Abbasi, K. Akkaya, and M. Younis, "A distributed connectivity restoration algorithm in wireless sensor and actor networks," in *Proceedings of the 32nd IEEE Conference on Local Computer Networks (LCN '07)*, pp. 496–503, Dublin, Ireland, October 2007.

[3] K. Akkaya, A. Thimrnapuram, F. Senel, and S. Uludag, "Distributed recovery of actor failures in wireless sensor and actor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08)*, pp. 2480–2485, Los Vegas, Nev, USA, April 2008.

[4] X. Liu, L. Xiao, A. Kreling, and Y. Liu, "Optimizing overlay topology by reducing cut vertices," in *Proceedings of the 16th Annual International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '06)*, May 2006.

[5] Y. Lin, B. Liang, and B. Li, "Data persistence in large-scale sensor networks with decentralized fountain codes," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1658–1666, May 2007.

[6] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Resilient peer-to-peer streaming," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP '03)*, 2003.

[7] Y. He, H. Ren, Y. Liu, and B. Yang, "On the reliability of large-scale distributed systems—a topological view," in *Proceedings of the 37th International Conference on Parallel Processing (ICPP '08)*, pp. 165–172, September 2008.

[8] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 2302–2312, March 2005.

[9] N. Tamboli and M. Younis, "Coverage-aware connectivity restoration in mobile sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, Dresden, Germany, June 2009.

[10] P. Basu and J. Redi, "Movement control algorithms for realization of fault-tolerant ad hoc robot networks," *IEEE Network*, vol. 18, no. 4, pp. 36–44, 2004.

[11] M. Imran, M. Younis, A. M. Said, and H. Hasbullah, "Partitioning detection and connectivity restoration algorithm for wireless sensor actor networks," in *Proceedings of the 8th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC '10)*, pp. 200–207, Hong Kong, China, December 2010.

[12] M. Imran, M. Younis, A. M. Said, and H. Hasbullah, "Localized motion-based connectivity restoration algorithms for wireless sensor and actor networks," *Journal of Network and Computer Applications*, vol. 35, pp. 844–856, 2012.

[13] K. Akkaya, F. Senel, A. Thimmapuram, and S. Uludag, "Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility," *IEEE Transactions on Computers*, vol. 59, no. 2, pp. 258–271, 2010.

[14] M. Younis, S. Lee, S. Gupta, and K. Fisher, "A localized self-healing algorithm for networks of moveable sensor nodes," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '08)*, pp. 1–5, December 2008.

[15] M. Imran, M. Younis, A. M. Said, and H. Hasbullah, "Volunteer-instigated connectivity restoration algorithm for wireless sensor and actor networks," in *Proceedings of the IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS '10)*, Beijing, China, June 2010.