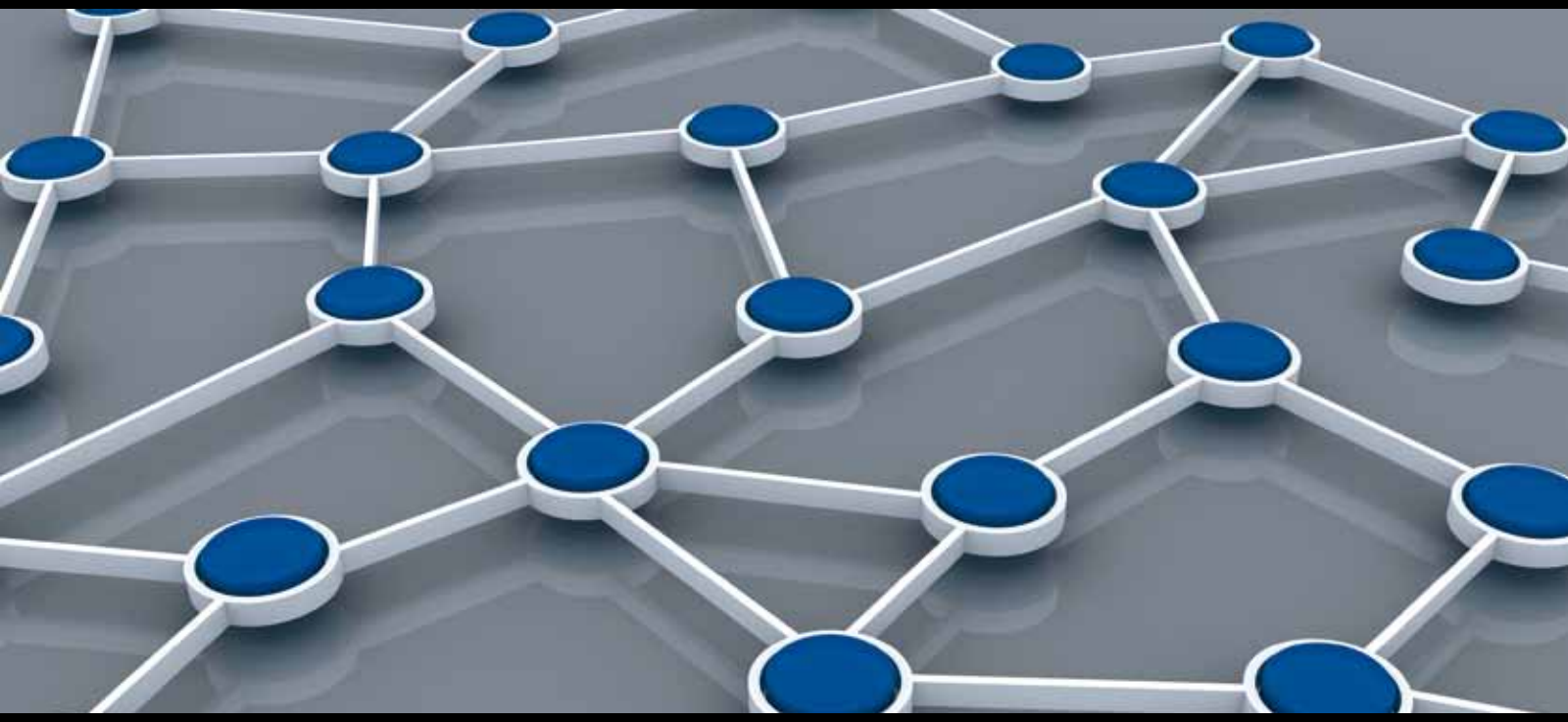
International Journal of Distributed Sensor Networks

Cooperative Communications for Wireless Ad Hoc and Sensor Networks

Guest Editors: Yong Sun, Shukui Zhang, Hongli Xu, and Shan Lin



Cooperative Communications for Wireless Ad Hoc and Sensor Networks

International Journal of Distributed Sensor Networks

Cooperative Communications for Wireless Ad Hoc and Sensor Networks

Guest Editors: Yong Sun, Shukui Zhang, Hongli Xu, and Shan Lin

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 161268, 2 pages
http://dx.doi.org/10.1155/2013/161268

Editorial

Cooperative Communications for Wireless Ad Hoc and Sensor Networks

Yong Sun,[1] Shukui Zhang,[1] Hongli Xu,[2] and Shan Lin[3]

[1] *Computer Science and Technology Institute, Soochow University, Suzhou, Jiangsu 215006, China*
[2] *School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230022, China*
[3] *Department of Computer and Information Sciences, Temple University, 324 Wachman Hall, 1805 N. Broad Street, Philadelphia, PA, USA*

Correspondence should be addressed to Yong Sun; suny@suda.edu.cn

Received 9 January 2013; Accepted 9 January 2013

The concept of cooperative communications for wireless ad hoc and sensor networks (WAHSNs) has recently attracted considerable attention. Different users or nodes in a WAHSN share resources to create collaboration through distributed transmission, which can significantly improve the performance of WANSNs. Thus, the problem that how relay nodes in the network cooperate with each other is the main subject in this special issue. The authors have focused on relay cooperation models based on the network coding, spectrum allocation, and models with space-time code, and so forth.

The paper "*An efficient reliable communication scheme in wireless sensor networks using linear network coding*," by J. Wang et al., addresses the modeling and design of linear network coding (LNC) for reliable communication against multiple failures in wireless sensor networks (WSNs). The proposed deterministic LNC scheme RDLC can significantly improve the network throughput. The authors also investigate the potential of random linear code RRLC for providing reliable communication in WSNs.

In the paper "*Network coded wireless cooperative multicast with minimum transmission cost*" by X. Wang et al., the authors explore the problem of minimum cost wireless cooperative multicast by using network coding. The authors propose a network coded hybrid source and cooperative exchange scheme to determine when to stop the source sending and start the exchange process, so as to minimize the total transmission cost.

The paper "*Cooperative transmission in cognitive radio ad hoc networks*," by J. Jia S. Zhang, investigates a cooperative transmission scheme to address the spectrum heterogeneity issue in cognitive radio ad hoc networks (CRAHNs) to improve the efficiency of spectrum utilization and the performance of cognitive radio networks. In particular, the authors describe several types of cooperative transmission and formulate a new resource allocation problem with joint relay selection and channel allocation.

The paper "*Towards aid by generate and solve methodology: application in the problem of coverage and connectivity in wireless sensor networks*," by P. R. Pinheiro et al., investigates the novel Generate and Solve (GS) methodology to solve the problem of coverage and connectivity in wireless sensor networks.

The paper "*Low-complexity decoding algorithms for distributed space-time coded regenerative relay systems*," by C. Zhang H. Yin, examines decoding structure for distributed space-time coded regenerative relay networks. Given the possible demodulation error at the regenerative relays, the authors provide a general framework of error aware decoder, where the receiver exploits the demodulation error probability of relays to improve the system performance. The authors also propose two low-complexity decoders.

The paper "*Integrated extensible simulation platform for vehicular sensor networks in smart cities*," by X. Tang et al., presents an integrated extensible simulation platform BHU-VSim for vehicular sensor networks (VSNs), which aims to support general simulation environment for typical vehicular applications in smart cities. And, as an initiate attempt, their platform provides significant improvement of VSNs' simulations.

The paper "*Efficient sensor localization method with classifying environmental sensor data*," by A.-c. Eun and Y.-g. Ha, proposes a novel localization method that uses environmental data recorded at each sensor location and a data classification technique to identify the location of sensor nodes.

The paper "*A path planning algorithm with a guaranteed distance cost in wireless sensor networks*," by Y. Liu et al., presents a distributed algorithm to obtain a path for the mobile node with minimum distance cost and effectively organize the network to ensure the availability of this path.

The paper "*Novel node localization algorithm based on nonlinear weighting least square for wireless sensor networks*," by F. Xiao et al., presents a new method for wireless sensor network node positioning based on nonlinear weighting least-square algorithm to explore the optimal solution and further reduce the positioning computational complexity by the simplification of the Taylor equation.

The paper "*A multiple-dimensional tree routing protocol for multisink wireless sensor networks based on ant colony optimization*," by H. Zhou et al., deals with the problem of a multiple-dimensional tree routing protocol for multisink wireless sensor networks based on ant colony optimization.

The paper "A Diagnosis-Based Clustering and Multipath Routing Protocol for Wireless Sensor Networks", by Wenjun Liu et al., proposes an energy-efficient data collection protocol which consists of clustering and multipath routing for fault diagnosis to ensure the gathering information accuracy and reduce energy additionally consumed by faulty nodes.

The paper "On guaranteed detectability for surveillance sensor networks," by Y. Zhu, proposes a fully distributed algorithm GAP for energy-efficient event detection for surveillance applications.

The paper "*ARQ protocols for two-way wireless relay systems: design and performance analysis*," by Z. Chen et al., proposes three basic automatic repeat-request (ARQ) protocols to improve the throughput of two-way relay systems, namely, relay-only ARQ (Ro-ARQ), terminal only ARQ (To-ARQ) and relay-terminal ARQ (RT-ARQ).

In the paper "*Distributed routing and spectrum allocation algorithm with cooperation in cognitive wireless mesh networks*," by Z. Chen et al., a distributed routing and spectrum allocation algorithm with cooperation (DRSAC-W) in cognitive wireless mesh networks is proposed, against the routing and spectrum allocation challenge in cognitive wireless mesh networks.

The paper "*NUNS: A nonuniform network split method for data-centric storage sensor networks*," by K.-Y. Lee et al., proposes a nonuniform network split(NUNS) method that distributes the load among sensor nodes in data-centric storage sensor networks and efficiently reduces the communication cost of expanding sensor networks.

The paper "*An efficient clustering algorithm in wireless sensor networks using cooperative communication*," by S. Zhang et al., constructs a minimum weakly connected dominating set (WCDS) as a clustering scheme for WSN.

The paper "*A hole-tolerant redundancy scheme for wireless sensor networks*," by J. Pu et al., introduces a new hole-tolerant redundancy scheme (HRS) which can prolong network lifetime while maintaining coverage and connectivity performance.

The paper "*CAC-MAC: a cross-layer adaptive cooperative MAC for wireless ad hoc networks*," by C. Shi et al., proposes a cross-layer adaptive data transmission algorithm considering both the length of data frame at the MAC layer and instantaneous wireless channel conditions. Under this algorithm, direct transmission mode or proper cooperative transmission mode will be adaptively selected for data packets according to both MAC layer and physical layer information.

The paper "*A mobile agent routing algorithm in dual-channel wireless sensor network*," by K. Liu et al., a mobile agent routing algorithm (MARA) is presented, and then based on the dual-channel communication model, the two-layer network combination optimization strategy is also proposed to make the energy of each nodes on the optimal route overall decline and hence improve the lifetime of network.

Yong Sun
Shukui Zhang
Hongli Xu
Shan Lin

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 795050, 11 pages
http://dx.doi.org/10.1155/2013/795050

Research Article

On the Performance of Quasiorthogonal STBC with Relay Selection and Phase Rotation Techniques for Decode and Forward Cooperative Communications

Nikorn Sutthisangiam,[1] Chaiyod Pirak,[1] and Gerd Ascheid[2]

[1] *The Sirindhorn International Thai-German Graduate School of Engineering (TGGS), King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand*
[2] *RWTH Aachen University, Aachen 52056, Germany*

Correspondence should be addressed to Nikorn Sutthisangiam; nikorns@gmail.com

Received 10 April 2012; Revised 31 October 2012; Accepted 12 November 2012

Academic Editor: Shan Lin

The performance of quasiorthogonal space-time block code with relay-selection and phase-rotation techniques applied to cooperative communications for four communication nodes is investigated. Specifically, by applying relay-selection and phase-rotation techniques, a diversity gain of four can be achieved. In addition, a symbol error rate (SER) performance analysis with closed-form expression and power allocation are investigated and compared with simulation results. The results show that theoretical SER curves are close to the simulation results. In addition, a code rate of the proposed scheme is two times higher than ordinary cooperative communications. The computer simulation results also show a significant probability of error improvement of about 2.8 dB over the conventional decode-and-forward protocol.

1. Introduction

In recent years, the utilization of multiple antennas at transmitters and receivers has gained popularity due to the potential of increasing the system capacity [1]. The increased spectral efficiency of such systems is also important because the bandwidth is a precious commodity, and by using multiple antennas at transmitters and receivers, the spectral efficiency can be drastically increased. Systems with multiple transmit and multiple receive antennas, more commonly known as multiple-input multiple-output (MIMO) systems, can provide a spatial diversity gain. This gain is obtained by transmitting or receiving copies of a signal through different antennas. This is an effective approach to combat fading in wireless channels and to improve the performance of the communication system.

Recently, a generalized MIMO system, called a cooperative communication, has been proposed for realizing the advantages of the conventional MIMO system, for example, the diversity gain [2]. By means of the cooperation of the active users equipped with a single antenna in the wireless networks, the generalized MIMO system can be established in a distributed fashion. In addition, the coverage range of such communication is also expanded, which results in lower power consumption for a particular user communicating with far-away destinations, and in turn prolongs the battery life.

Another approach for increasing the transmission rate is to employ a transmit diversity based on a space-time block code (STBC) technique [3]. However, the complex-valued STBC which provides a full code rate, and a full diversity gain does not exist for more than two transmit antennas [3]. In fact, orthogonal-STBC designed for more than two antennas can achieve full diversity gain, but its code rate is less than unity. On the other hand, quasi-orthogonal STBC (QO-STBC) [4], proposed for four transmit antennas, achieves the full code rate, but it suffers from a loss in diversity order due to a coupling effect between the symbols in the codeword.

Given the advantages of QO-STBC, it can be applied to cooperative communications for performance enhancement with relay-selection or phase-rotation techniques. The contributions of this paper are as follows.

(i) It can be shown that the proposed QO-STBC decode-and-forward cooperative communication (QO-DF) system can enhance the performance of the system such that the diversity of four is achieved. In addition, the code rate of the proposed scheme is two times higher than the ordinary cooperative communications.

(ii) The optimum and suboptimum power allocations are investigated. In addition, the system performance can be enhanced by adopting the optimum power allocation scheme.

(iii) We analyze the theoretical symbol error rate and compare the theoretical results with the simulation results. It turns out that the theoretical and simulation results are close to each other, which could confirm the validity of the theoretical result.

The rest of this paper is organized as follows. In Section 2, we present a conventional decode-and-forward (DF) protocol for four-node cooperative communications. In Section 3, we describe the proposed QO-DF cooperative communications with relay-selection and phase-rotation techniques. The maximum ratio combining (MRC) and the signal-to-noise ratio (SNR) of the proposed system are described in Section 4. The theoretical SER analysis and optimum power allocation of the proposed system are described in Section 5. The simulation results compared with the theoretical results are shown in Section 6. Finally, we conclude this paper in Section 7.

2. System Model and Conventional Decode-and-Forward Protocol for Wireless Ad hoc Networks

In cooperative wireless communications, for example, wireless ad hoc networks, wireless users can cooperate with neighbouring users to form a generalized MIMO system with a coding scheme, for example, STBC, for enhancing the system performance, for example, a probability of error. The conventional DF cooperative communication system model with a single relay is described in [5]. However, for the sake of exposition, we consider cooperative communications in the case of a wireless network with two phases and four communication nodes (i.e., one user acts as a source node and the other four users act as relay nodes), and one destination node as shown in Figure 1.

In phase I, node 1 transmits a modulated signal to its destination, while nodes 2, 3, and 4 receive this transmitted signal due to the broadcast nature of wireless channels. In phase II, nodes 2, 3, and 4 will retransmit the received signal to node 1's destination in a DF fashion. Likewise, in the next communication periods, node 2, 3, or 4 will act as the source node, and the other users will act as the relay nodes, respectively. In both phases, all nodes transmit the signal through orthogonal channels using time-division multiplexing (TDMA). In this paper, we employ an M-PSK modulation scheme.

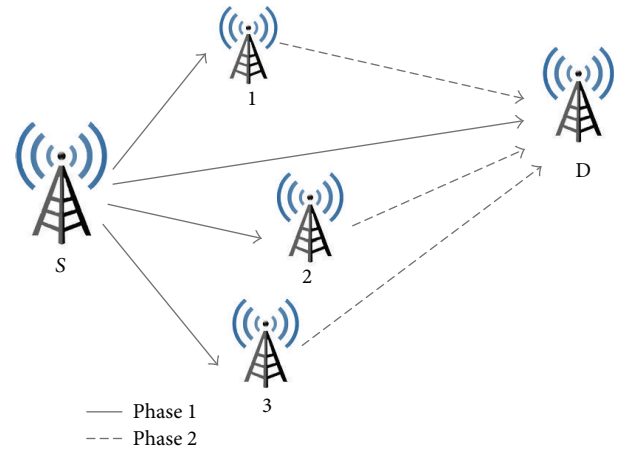


Figure 1: A system model of four-node cooperative communications.

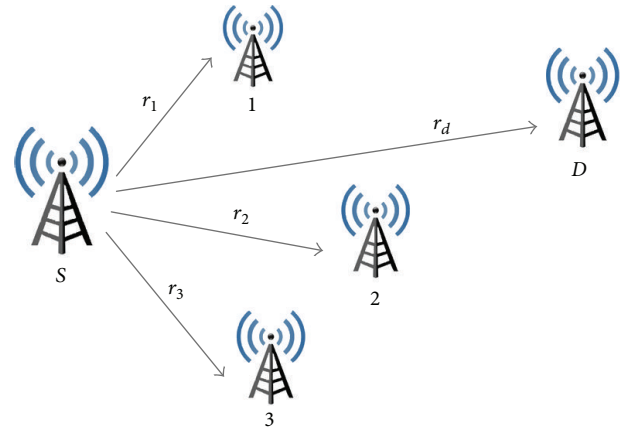


Figure 2: A source node broadcasts the transmit signal to relay nodes in phase I.

In phase I, the source node broadcasts its transmit signal to the destination and the relay nodes in the first time t_1, as shown in Figure 2. The received signal expressions for phase I can be expressed as follows:

$$\begin{aligned} r_1 &= \sqrt{P_d} h_{s1} x + n, \\ r_2 &= \sqrt{P_d} h_{s2} x + n, \\ r_3 &= \sqrt{P_d} h_{s3} x + n, \\ r_d &= \sqrt{P_d} h_{sd} x + n, \end{aligned} \tag{1}$$

where P_d is the transmit power of the source node, x is the transmitted signal from the source, r_1, r_2, and r_3 are the received signal at relays 1, 2, and 3, respectively, and r_d is the received signal at the destination. h_{ij} is the channel impulse response from node i to j, and n is the additive white Gaussian noise (AWGN).

After relays 1, 2, and 3 received broadcasting signals from the source node, and decoded these signals using a Maximum-Likelihood (ML) receiver, the decoded symbols

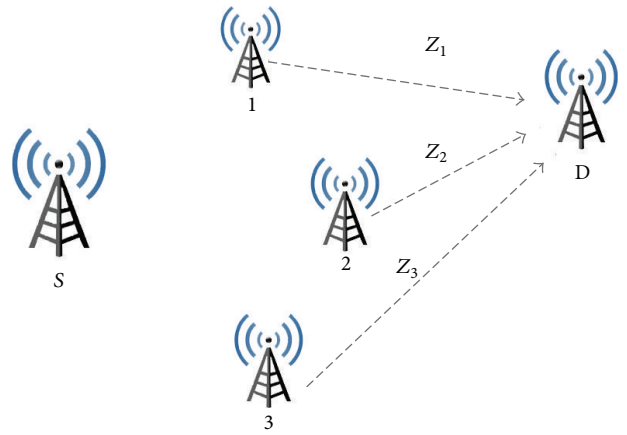


Figure 3: Each relay node sends the decoded signals to the destination node in phase II.

can be written as $\widetilde{x_1}$, $\widetilde{x_2}$, and $\widetilde{x_3}$, respectively. The relay nodes will modulate the decoded symbols and retransmit them to the destination node in phase II in the sequential time intervals t_2, t_3, and t_4, respectively, as shown in Figure 3.

The expressions are as follows:

$$\begin{aligned} z_1 &= \sqrt{\frac{P_q}{3}} h_{1d} \widetilde{x_1} + n, \\ z_2 &= \sqrt{\frac{P_q}{3}} h_{2d} \widetilde{x_2} + n, \\ z_3 &= \sqrt{\frac{P_q}{3}} h_{3d} \widetilde{x_3} + n, \end{aligned} \tag{2}$$

where P_q is the transmit power of the relay nodes, z_1, z_2, and z_3 are the received signals at the destination node in phase II, which are sent by the relays 1, 2, and 3, respectively. At the destination node, the MRC is performed as follows [6]:

$$y = \sqrt{P_d} \frac{h_{sd}^*}{N_0} r_d + \sqrt{\frac{P_q}{3}} \frac{h_{1d}^*}{N_0} z_1 + \sqrt{\frac{P_q}{3}} \frac{h_{2d}^*}{N_0} z_2 + \sqrt{\frac{P_q}{3}} \frac{h_{3d}^*}{N_0} z_3, \tag{3}$$

where y is the combined received signal at the destination node and N_0 is the variance of noise. In phases I and II, we can observe that the DF cooperative communication uses four time slots to send one symbol so that the code rate is equal to 1/4. Some improvement could be made by properly using a space-time coding scheme in phase II.

3. The Proposed Quasi-Orthogonal STBC Decode-and-Forward Cooperative Communications

Now, we consider four cooperative communication nodes as a multiple-input single-output (MISO) communication system, as shown in Figure 4. We also consider the source and three relays in the cooperative communications as four transmit antennas in the MISO communication, and apply QO-STBC [4], as shown in Figure 5.

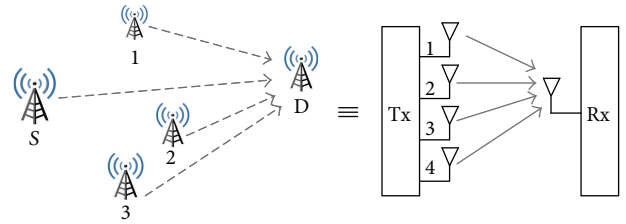


Figure 4: An equivalent system diagram of cooperative communications in phase II and a MISO system.

In phase I, a source node encodes four consecutive symbols to a QO-STBC codeword and broadcasts to the relay nodes and destination node in four time slots. Then the relay nodes will decode the received signals and send them individually to the destination node. We regard the cooperation in phase II of three relays and the source node as four transmit antennas for the QO-STBC scheme, in which relay 1, relay 2, relay 3, and the source act as the first antenna, second antenna, third antenna, and fourth antenna, respectively. Therefore, four data blocks are transmitted over four consecutive block intervals through four antennas using the following 4×4 QO-STBC code matrix [7],

$$C = \begin{matrix} \xrightarrow{\text{Space}} \\ \begin{bmatrix} s_1 & s_2 & s_3 & s_4 \\ -s_2^* & s_1^* & -s_4^* & s_3^* \\ -s_3^* & -s_4^* & s_1^* & s_2^* \\ s_4 & -s_3 & -s_2 & s_1 \end{bmatrix} \end{matrix} \begin{matrix} \\ \\ \\ \\ \downarrow \text{Time} \end{matrix} \tag{4}$$

where $\mathbf{C}$ is a QO-STBC code matrix. In addition, the channels can be modelled as a matrix of 4×1, whose coefficients are the same as the frequency response of the channels h_{ij}. In the first block interval, the blocks s_1, s_2, s_3, and s_4 are transmitted by transmitting a power of $P_q/4$ simultaneously from the first, second, third, and fourth antennas, respectively. The received signal corresponding to these blocks is expressed by r_1. In a similar way, the blocks of $-s_2^*$, s_1^*, $-s_4^*$, and s_3^* are transmitted during the second block interval simultaneously over four antennas, and the corresponding received block is expressed by r_2, and so on for the third and the fourth block intervals. The received signal blocks y_1, y_2, y_3, and y_4 in the first, second, third, and fourth block intervals, respectively, can be written as

$$\begin{aligned} y_1 &= \sqrt{\frac{P_q}{4}} \left(h_{1d} s_1 + h_{2d} s_2 + h_{3d} s_3 + h_{sd} s_4 \right) + n_1, \\ y_2 &= \sqrt{\frac{P_q}{4}} \left(-h_{1d} s_2^* + h_{2d} s_1^* - h_{3d} s_4^* + h_{sd} s_3^* \right) + n_2, \\ y_3 &= \sqrt{\frac{P_q}{4}} \left(-h_{1d} s_3^* - h_{2d} s_4^* + h_{3d} s_1^* + h_{sd} s_2^* \right) + n_3, \\ y_4 &= \sqrt{\frac{P_q}{4}} \left(h_{1d} s_4 - h_{2d} s_3 - h_{3d} s_2 + h_{sd} s_1 \right) + n_4. \end{aligned} \tag{5}$$

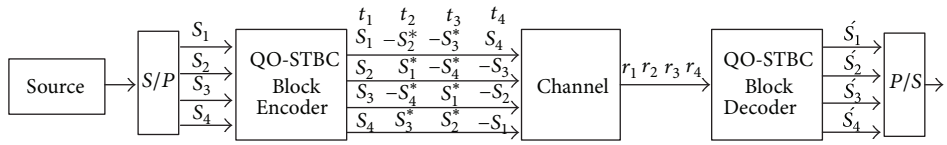


Figure 5: An equivalent block diagram of the proposed QO-STBC.

For the sake of simplicity, we replace h_{1d}, h_{2d}, h_{3d}, and h_{sd} by h_1, h_2, h_3, and h_4, respectively. Hence, we have

$$\begin{bmatrix} y_1 \\ y_2^* \\ y_3^* \\ y_4 \end{bmatrix} = \sqrt{\frac{P_q}{4}} \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \\ h_2^* & -h_1^* & h_4^* & -h_3^* \\ h_3^* & h_4^* & -h_1^* & -h_2^* \\ h_4 & -h_3 & -h_2 & h_1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix}. \quad (6)$$

We can derive (6) in a vector form as follows,

$$\mathbf{y}_k = \mathbf{h}_k \mathbf{s}_k + \mathbf{n}_k, \quad k = 1, 2, 3, 4. \quad (7)$$

At the receiver, the matched filtering is performed as follows [8]:

$$\widetilde{\mathbf{y}_k} = \mathbf{h}_k^{\mathbf{H}} \mathbf{y}_k = \mathbf{h}_k^{\mathbf{H}} \mathbf{h}_k \mathbf{s}_k + \mathbf{h}_k^{\mathbf{H}} \mathbf{n}_k, \quad (8)$$

where $\mathbf{h}_k^{\mathbf{H}}$ is a Hermitian matrix of $\mathbf{h}_k$.

For the orthogonal STBC, $\mathbf{h}_k$ would be a unitary matrix, and, hence, $\mathbf{h}_k^{\mathbf{H}} \mathbf{h}_k$ would be a diagonal matrix. In this case, $\widetilde{\mathbf{y}_k}$ would provide an estimate of $\mathbf{s_k}$. However, for the case of four antennas as in the QO-STBC scheme, the matrix $\mathbf{h}_k^{\mathbf{H}} \mathbf{h}_k$ is not diagonal, but it is in the following form:

$$\mathbf{h}_k^{\mathbf{H}} \mathbf{h}_k = \begin{bmatrix} \beta & 0 & 0 & \alpha \\ 0 & \beta & -\alpha & 0 \\ 0 & -\alpha & \beta & 0 \\ \alpha & 0 & 0 & \beta \end{bmatrix}, \quad (9)$$

where

$$\beta = \sum_{k=1}^{4} |h_k|^2, \qquad \alpha = 2 \operatorname{Re} \{h_1 h_4^* - h_2 h_3^*\}. \quad (10)$$

It can be seen that the signal-to-noise ratio (SNR) can be maximized if α is forced to zero. For an orthogonal STBC design, α is zero regardless of the channel coefficient values. However, for the case of four antennas, we require some feedbacks from the receiver in order to force α to zero. The techniques to force a coupling term α to zero, and to obtain a diversity order of four are relay-selection and phase-rotation techniques.

3.1. Relay-Selection Technique. Assuming that symbols from each relay are sent and multiplied by real-valued variables $\theta_k, k = 1, 2, 3, 4$, where the coefficients θ_k have a binary value $\{0, 1\}$, as explained below. The new coupling term becomes

$$\alpha = 2 \operatorname{Re} \{\theta_1 h_1 \theta_4 h_4^* - \theta_2 h_2 \theta_3 h_3^*\}. \quad (11)$$

The variables θ_k can be chosen such that

$$\begin{aligned} &\text{If } |h_1|^2 > |h_4|^2, \quad \text{then } \theta_1 = 1,\ \theta_4 = 0, \\ &\qquad\qquad\quad \text{else } \theta_1 = 0,\ \theta_4 = 1, \\ &\text{If } |h_2|^2 > |h_3|^2, \quad \text{then } \theta_2 = 1,\ \theta_3 = 0, \\ &\qquad\qquad\quad \text{else } \theta_2 = 0,\ \theta_3 = 1. \end{aligned} \quad (12)$$

The new received expressions can be expressed as follows:

$$\begin{aligned} y_1 &= \sqrt{\frac{P_q}{4}} \left(\theta_1 h_1 s_1 + \theta_2 h_2 s_2 + \theta_3 h_3 s_3 + \theta_4 h_4 s_4\right) + n_1, \\ y_2 &= \sqrt{\frac{P_q}{4}} \left(-\theta_1 h_1 s_2^* + \theta_2 h_2 s_1^* - \theta_3 h_3 s_4^* + \theta_4 h_4 s_3^*\right) + n_2, \\ y_3 &= \sqrt{\frac{P_q}{4}} \left(-\theta_1 h_1 s_3^* - \theta_2 h_2 s_4^* + \theta_3 h_3 s_1^* + \theta_4 h_4 s_2^*\right) + n_3, \\ y_4 &= \sqrt{\frac{P_q}{4}} \left(\theta_1 h_1 s_4 - \theta_2 h_2 s_3 - \theta_3 h_3 s_2 + \theta_4 h_4 s_1\right) + n_4. \end{aligned} \quad (13)$$

Then, the new matrix $\mathbf{h}_k^{\mathbf{H}}$ becomes

$$\mathbf{h}_k^{\mathbf{H}} = \begin{bmatrix} \theta_1 h_1^* & \theta_2 h_2 & \theta_3 h_3 & \theta_4 h_4^* \\ \theta_2 h_2^* & -\theta_1 h_1 & \theta_4 h_4 & -\theta_3 h_3^* \\ \theta_3 h_3^* & \theta_4 h_4 & -\theta_1 h_1 & -\theta_2 h_2^* \\ \theta_4 h_4^* & -\theta_3 h_3 & -\theta_2 h_2 & \theta_1 h_1^* \end{bmatrix}. \quad (14)$$

This technique would force α to zero, and we could obtain a diversity order of four, while preserving the total transmitted power. Basically, this technique chooses the best two channels in the antenna pairs (1, 4) and (2, 3) according to the channel quality indicated at the receiver. The QO-STBC scheme is then applied to use the best antenna in each pair. Therefore, it provides a diversity gain of four.

3.2. Phase-Rotation Technique. Another way to force α to be zero is to use a phase-rotation approach. We consider that the symbols transmitted from the third and the fourth antennas are rotated by a common phasor $e^{j\theta}$. Note that this operation does not change the transmitted power. Since the phase rotation on transmitted symbols is effectively equivalent to rotating the phases of the corresponding channel coefficients, the new coupling term can be written as

$$\alpha = 2 \operatorname{Re} \left\{\left(h_1 h_4^* - h_2 h_3^*\right) e^{-j\theta}\right\}. \quad (15)$$

Let $\rho = (h_1 h_4^* - h_2 h_3^*)$, in order to force α to zero, the product of ρ and $e^{-j\theta}$ should be a complete imaginary number. This can be achieved when angle $(\rho) - \theta$ is either $-\pi/2$ or $\pi/2$. Therefore, θ is determined by

$$\theta = \frac{\pi}{2} - \text{angle}\,(\rho) \quad \text{or} \quad \frac{3\pi}{2} - \text{angle}\,(\rho)\,. \tag{16}$$

Hence, the phase angle is limited to $\theta \in [-\pi/2, \pi/2)$. The new expressions for received signals in (13), respectively, can be expressed as follows:

$$\begin{aligned}
y_1 &= \sqrt{\frac{P_q}{4}}\left(h_1 s_1 + h_2 s_2 + e^{j\theta} h_3 s_3 + e^{j\theta} h_4 s_4\right) + n_1,\\
y_2 &= \sqrt{\frac{P_q}{4}}\left(-h_1 s_2^* + h_2 s_1^* - e^{j\theta} h_3 s_4^* + e^{j\theta} h_4 s_3^*\right) + n_2,\\
y_3 &= \sqrt{\frac{P_q}{4}}\left(-h_1 s_3^* - h_2 s_4^* + e^{j\theta} h_3 s_1^* + e^{j\theta} h_4 s_2^*\right) + n_3,\\
y_4 &= \sqrt{\frac{P_q}{4}}\left(h_1 s_4 - h_2 s_3 - e^{j\theta} h_3 s_2 + e^{j\theta} h_4 s_1\right) + n_3.
\end{aligned} \tag{17}$$

Then, the new matrix $\mathbf{h}_\mathbf{k}^\mathbf{H}$ becomes

$$\mathbf{h}_\mathbf{k}^\mathbf{H} = \begin{bmatrix} h_1^* & h_2 & e^{j\theta} h_3 & e^{-j\theta} h_4^* \\ h_2^* & -h_1 & e^{j\theta} h_4 & -e^{-j\theta} h_3^* \\ e^{-j\theta} h_3^* & e^{j\theta} h_4 & -h_1 & -h_2^* \\ e^{-j\theta} h_4^* & -e^{j\theta} h_3 & -h_2 & h_1^* \end{bmatrix}. \tag{18}$$

Furthermore, the relay-selection and phase-rotation techniques can be applied to eight or more nodes, as described in [9].

4. Maximum Ratio Combining and Signal to Noise Ratio Analysis

In DF cooperative communications, a destination jointly combines the signal received from a source in phase I and the signal received from the relays in phase II by using the maximum ratio combining (MRC) method and detects the combined received symbols by using the ML receiver.

4.1. Maximum Ratio Combining. In the proposed system, we can use an MRC combiner at the destination node by combining a direct signal from the source in phase I and retransmitted signals in phase II. The MRC combining expressions are as follows:

$$\begin{aligned}
u_1 &= w_d h_{\text{sd}}^* r_1 + w_q \left(y_1 h_1^* + y_2 h_2 + y_3 h_3 + y_4 h_4^*\right),\\
u_2 &= w_d h_{\text{sd}}^* r_2 + w_q \left(y_1 h_2^* - y_2 h_1 + y_3 h_4 - y_4 h_3^*\right),\\
u_3 &= w_d h_{\text{sd}}^* r_3 + w_q \left(y_1 h_3^* + y_2 h_4 - y_3 h_1 - y_4 h_2^*\right),\\
u_4 &= w_d h_{\text{sd}}^* r_4 + w_q \left(y_1 h_4^* - y_2 h_3 - y_3 h_2 + y_4 h_1^*\right),
\end{aligned} \tag{19}$$

where u_1, u_2, u_3, and u_4 are outputs of the MRC combiner to be used for decoding $\widetilde{s_1}, \widetilde{s_2}, \widetilde{s_3}$, and $\widetilde{s_4}$, respectively, and

$$w_d = \frac{\sqrt{P_d}}{N_0}, \qquad w_q = \frac{\sqrt{P_q/4}}{N_0}, \tag{20}$$

where w_d is a weighting coefficient of the direct signal from phase I and w_q is a weighting coefficient of the retransmit signal from phase II. We then employ an ML detector to detect $\widetilde{s_1}, \widetilde{s_2}, \widetilde{s_3}$, and $\widetilde{s_4}$, respectively.

However, it is worth noting that the MRC combining yields the maximum SNR to (19), given that the estimated symbols $\widetilde{s_1}, \widetilde{s_2}, \widetilde{s_3}$, and $\widetilde{s_4}$ at the relay nodes are correctly decoded. Specifically, in practical applications, the correctness of $\widetilde{s_1}, \widetilde{s_2}, \widetilde{s_3}$, and $\widetilde{s_4}$ depends solely on the quality of the channel links from the source-to-relay link. Hence, the MRC combining cannot guarantee the maximum SNR, as mentioned in [6]. The most useful method for improving the performance of the proposed system is to employ a power allocation scheme, which will be described later.

4.2. Signal-to-Noise Ratio Analysis. In this section, we derive an expression of the SNR for the proposed QO-DF cooperative communication system. The SNR output of the MRC combiner at the destination node consists of both direct and relay signals. It can be expressed as follows [6]:

$$\gamma_{\text{QO-DF}} = \gamma_d + \gamma_q, \tag{21}$$

where $\gamma_{\text{QO-DF}}$ is the received SNR at the destination node, γ_d is the SNR of the direct signal in phase I, and γ_q is the SNR of the retransmitted signal in phase II. Assuming that the transmitted symbol of the direct signal in phase I and retransmitted signal in phase II have an average energy of 1, we can derive the SNR of the received signal in each phase as follows:

$$\begin{aligned}
\gamma_d &= \frac{P_d |h_{sd}|^2}{N_0},\\
\gamma_q &= \frac{\left(P_q/4\right)|h_1|^2 + \left(P_q/4\right)|h_2|^2}{N_0}\\
&\quad + \frac{\left(P_q/4\right)|h_3|^2 + \left(P_q/4\right)|h_4|^2}{N_0}.
\end{aligned} \tag{22}$$

Hence, the total received SNR at the destination can be expressed as

$$\begin{aligned}
&\gamma_{\text{QO-DF}}\\
&= \frac{P_d |h_{sd}|^2 + \left(P_q/4\right)|h_1|^2 + \left(P_q/4\right)|h_2|^2}{N_0}\\
&\quad + \frac{\left(P_q/4\right)|h_3|^2 + \left(P_q/4\right)|h_4|^2}{N_0}
\end{aligned} \tag{23}$$

5. Symbol Error Rate Analysis and Optimum Power Allocation

In this section, we consider the symbol error rate (SER) performance analysis of the proposed QO-DF communication system with the M-PSK modulation scheme. First, we consider the SER of the M-PSK signal of the relays in phase I. Let Pe_1, Pe_2, and Pe_3 be incorrect decoding probabilities per a symbol of source to relay 1, source to relay 2, and source to relay 3, respectively. According to the SNR analysis in the previous section, we can obtain the SER expression of each relay node in phase I as follows [10]:

$$Pe_1 = \Psi(\gamma_{S1}) = \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\mathrm{PSK}}P_d|h_{S1}|^2}{N_0\sin^2\theta}\right)d\theta,$$

$$Pe_2 = \Psi(\gamma_{S2}) = \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\mathrm{PSK}}P_d|h_{s2}|^2}{N_0\sin^2\theta}\right)d\theta,$$

$$Pe_3 = \Psi(\gamma_{S3}) = \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\mathrm{PSK}}P_d|h_{s3}|^2}{N_0\sin^2\theta}\right)d\theta, \tag{24}$$

where $b_{\mathrm{PSK}} = \sin^2(\pi/M)$ and $M = 2^k$ with k even.

Over the Rayleigh fading, we average channels h_{s1}, h_{s2}, and h_{s3} with variances $\delta_{s1}^2, \delta_{s2}^2$, and δ_{s3}^2, respectively. Since the fading channels h_{s1}, h_{s2}, and h_{s3} are independent of each other, we can express the incorrect decoding probability of each relay as [5]

$$Pe_1 = F_1\left(1 + \frac{b_{\mathrm{PSK}}P_d\delta_{s1}^2}{N_0\sin^2\theta}\right),$$

$$Pe_2 = F_1\left(1 + \frac{b_{\mathrm{PSK}}P_d{\delta_{s2}}^2}{N_0\sin^2\theta}\right), \tag{25}$$

$$Pe_3 = F_1\left(1 + \frac{b_{\mathrm{PSK}}P_d{\delta_{s3}}^2}{N_0\sin^2\theta}\right),$$

where

$$F_1(x(\theta)) = \frac{1}{\pi}\int_0^{(M-1)\pi/M} \frac{1}{x(\theta)}d\theta. \tag{26}$$

5.1. The Proposed Cooperative Strategy. According to the received signals in phase I, in reality, the relay nodes do not always correctly decode the transmitted symbols. For setting the cooperative protocol strategy in phase II, the relay nodes are assumed to be capable of deciding whether or not it has decoded correctly. This could be achieved through cyclic redundancy check (CRC) codes or approaches by setting an SNR threshold at the relay nodes [11]. In addition, we also assume short-term statistics of the channels [6, 12], that is, channel variances, within a certain period of time to be known to the source node.

For the proposed QO-DF protocol, if no relay incorrectly decodes the symbols, all relays forward the decoded symbols to the destination by quasi-orthogonal space-time coding; otherwise, only the relays that correctly decode the symbols forward them to the destination by a conventional DF method. The total received SNR at the destination depends on the number of relays decoded whose symbols are correct. The cooperative protocol strategy is proposed in Table 1.

Table 1: Cooperative strategy.

Cooperation protocol	No. of incorrect relays	Total received SNR (γ_{total})
Direct signal only (noncooperative)	3	$\gamma_{\mathrm{noncooperative}}$
1-relay cooperative DF	2	$\gamma_{\text{1-relay}}$
2-relay cooperative DF	1	$\gamma_{\text{2-relay}}$
3-relay cooperative QO-DF	0	$\gamma_{\text{QO-DF}}$

This cooperative strategy is expected to achieve a performance diversity order of four. In order to achieve a diversity of order four, all relays have to decode the symbols correctly. The total SER of the proposed QO-DF system can be written as

$$Pe_{\mathrm{total}} = \Psi(\gamma_{\mathrm{total}}) = \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\mathrm{PSK}}\gamma_{\mathrm{total}}}{N_0\sin^2\theta}\right)d\theta, \tag{27}$$

in which Pe_{total} greatly depends on the SNR of the cooperative protocol strategy. We can readily express (27) as follows:

$$\begin{aligned} Pe_{\mathrm{total}} &= \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\mathrm{PSK}}\gamma_{\mathrm{noncooperative}}}{N_0\sin^2\theta}\right)d\theta \\ &\quad \times \text{probability of 3 relays error} \\ &\quad + \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\mathrm{PSK}}\gamma_{\text{1-relay}}}{N_0\sin^2\theta}\right)d\theta \\ &\quad \times \text{probability of 2 relays error} \\ &\quad + \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\mathrm{PSK}}\gamma_{\text{2-relay}}}{N_0\sin^2\theta}\right)d\theta \\ &\quad \times \text{probability of 1 relay error} \\ &\quad + \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\mathrm{PSK}}\gamma_{\text{QO-DF}}}{N_0\sin^2\theta}\right)d\theta \\ &\quad \times \text{probability of none relay error.} \end{aligned} \tag{28}$$

The first component shows the signal received at the destination which is a noncooperative scheme; the second component shows a 1-relay cooperative scheme; the third component shows a 2-relay cooperative scheme; the fourth component shows a 3-relay cooperative with quasi-orthogonal space-time coding.

According to the incorrect decoding probability of each relay in phase I, we can obtain the conditional SER of the proposed QO-DF protocol as follows:

$$
\begin{aligned}
Pe_{\text{total}} = {} & \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\text{PSK}}P_d|h_{sd}|^2}{N_0\sin^2\theta}\right)d\theta \\
& \times \left[1-(1-Pe_1)^4\right]\left[1-(1-Pe_2)^4\right] \\
& \times \left[1-(1-Pe_3)^4\right] \\
& + \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\text{PSK}}\left(P_d|h_{sd}|^2+P_q|h_1|^2\right)}{N_0\sin^2\theta}\right)d\theta \\
& \times (1-Pe_1)^4\left[1-(1-Pe_2)^4\right]\left[1-(1-Pe_3)^4\right] \\
& + \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\text{PSK}}\left(P_d|h_{sd}|^2+P_q|h_2|^2\right)}{N_0\sin^2\theta}\right)d\theta \\
& \times (1-Pe_2)^4\left[1-(1-Pe_1)^4\right]\left[1-(1-Pe_3)^4\right] \\
& + \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\text{PSK}}\left(P_d|h_{sd}|^2+P_q|h_3|^2\right)}{N_0\sin^2\theta}\right)d\theta \\
& \times (1-Pe_3)^4\left[1-(1-Pe_1)^4\right]\left[1-(1-Pe_2)^4\right] \\
& + \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\text{PSK}}\left(P_d|h_{sd}|^2}{N_0\sin^2\theta}\right.\right. \\
& \qquad \left.\left. + \frac{\left(P_q/2\right)|h_1|^2}{N_0\sin^2\theta} + \frac{\left(P_q/2\right)|h_2|^2\right)}{N_0\sin^2\theta}\right)d\theta \\
& \times (1-Pe_1)^4(1-Pe_2)^4\left[1-(1-Pe_3)^4\right] \\
& + \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\text{PSK}}\left(P_d|h_{sd}|^2}{N_0\sin^2\theta}\right.\right. \\
& \qquad \left.\left. + \frac{\left(P_q/2\right)|h_1|^2}{N_0\sin^2\theta} + \frac{\left(P_q/2\right)|h_3|^2\right)}{N_0\sin^2\theta}\right)d\theta \\
& \times (1-Pe_1)^4(1-Pe_3)^4\left[1-(1-Pe_2)^4\right] \\
& + \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\text{PSK}}\left(P_d|h_{sd}|^2}{N_0\sin^2\theta}\right.\right. \\
& \qquad \left.\left. + \frac{\left(P_q/2\right)|h_2|^2}{N_0\sin^2\theta} + \frac{\left(P_q/2\right)|h_3|^2\right)}{N_0\sin^2\theta}\right)d\theta \\
& \times (1-Pe_2)^4(1-Pe_3)^4\left[1-(1-Pe_1)^4\right] \\
& + \frac{1}{\pi}\int_0^{(M-1)\pi/M} \exp\left(\frac{-b_{\text{PSK}}\left(P_d|h_{sd}|^2+\left(P_q/4\right)|h_1|^2}{N_0\sin^2\theta}\right. \\
& \qquad + \frac{\left(P_q/4\right)|h_2|^2\left(P_q/4\right)|h_3|^2}{N_0\sin^2\theta} \\
& \qquad \left. + \frac{\left(P_q/4\right)|h_4|^2\right)}{N_0\sin^2\theta}\right)d\theta \\
& \times (1-Pe_1)^4(1-Pe_2)^4(1-Pe_3)^4,
\end{aligned}
\tag{29}
$$

where $(1-Pe_i)^4$ is a correctly decoding probability of QO-STBC codeword at relay ith. In a similar way, $(1-Pe_i)^4$ is a chance of incorrectly decoding. Furthermore, at higher SNR, we can approximate $1-(1-Pe_i)^4 \approx 4Pe_i$.

In the following analysis, we average the channels in phase II over the Rayleigh fading as in phase I. We set up h_{sd}, h_1, h_2, h_3, and h_4 having variance of $\delta_{sd}^2, \delta_1^2, \delta_2^2, \delta_3^2$, and δ_4^2, respectively. We are able to obtain the approximate SER equation of the proposed QO-DF cooperative communications with M-PSK modulation in a full closed-form expression as follows:

$$
\begin{aligned}
Pe_{\text{total}} = {} & 64F_1\left(1+\frac{b_{\text{PSK}}P_d\delta_{sd}^2}{N_0\sin^2\theta}\right)F_1\left(1+\frac{b_{\text{PSK}}P_d\delta_{s1}^2}{N_0\sin^2\theta}\right) \\
& \times F_1\left(1+\frac{b_{\text{PSK}}P_d\delta_{s2}^2}{N_0\sin^2\theta}\right)F_1\left(1+\frac{b_{\text{PSK}}P_d\delta_{s3}^2}{N_0\sin^2\theta}\right) \\
& + 16F_1\left[\left(1+\frac{b_{\text{PSK}}P_d\delta_{sd}^2}{N_0\sin^2\theta}\right)\left(1+\frac{b_{\text{PSK}}P_q\delta_1^2}{N_0\sin^2\theta}\right)\right] \\
& \times F_1\left(1+\frac{b_{\text{PSK}}P_d\delta_{s2}^2}{N_0\sin^2\theta}\right)F_1\left(1+\frac{b_{\text{PSK}}P_d\delta_{s3}^2}{N_0\sin^2\theta}\right) \\
& \times \left[1-4F_1\left(1+\frac{b_{\text{PSK}}P_d\delta_{s1}^2}{N_0\sin^2\theta}\right)\right] \\
& + 16F_1\left[\left(1+\frac{b_{\text{PSK}}P_d\delta_{sd}^2}{N_0\sin^2\theta}\right)\left(1+\frac{b_{\text{PSK}}P_q\delta_2^2}{N_0\sin^2\theta}\right)\right] \\
& \times F_1\left(1+\frac{b_{\text{PSK}}P_d\delta_{s1}^2}{N_0\sin^2\theta}\right)F_1\left(1+\frac{b_{\text{PSK}}P_d\delta_{s3}^2}{N_0\sin^2\theta}\right) \\
& \times \left[1-4F_1\left(1+\frac{b_{\text{PSK}}P_d\delta_{s2}^2}{N_0\sin^2\theta}\right)\right] \\
& + 16F_1\left[\left(1+\frac{b_{\text{PSK}}P_d\delta_{sd}^2}{N_0\sin^2\theta}\right)\left(1+\frac{b_{\text{PSK}}P_q\delta_3^2}{N_0\sin^2\theta}\right)\right]
\end{aligned}
$$

$$\times F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s1}^2}{N_0\sin^2\theta}\right)F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s2}^2}{N_0\sin^2\theta}\right)$$
$$\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s3}^2}{N_0\sin^2\theta}\right)\right]$$
$$+4F_1\left[\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{sd}^2}{N_0\sin^2\theta}\right)\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/2\right)\delta_1^2}{N_0\sin^2\theta}\right)\times\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/2\right)\delta_2^2}{N_0\sin^2\theta}\right)\right]$$
$$\times F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s3}^2}{N_0\sin^2\theta}\right)$$
$$\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s1}^2}{N_0\sin^2\theta}\right)\right]$$
$$\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s2}^2}{N_0\sin^2\theta}\right)\right]$$
$$+4F_1\left[\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{sd}^2}{N_0\sin^2\theta}\right)\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/2\right)\delta_1^2}{N_0\sin^2\theta}\right)\times\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/2\right)\delta_3^2}{N_0\sin^2\theta}\right)\right]$$
$$\times F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s2}^2}{N_0\sin^2\theta}\right)$$
$$\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s1}^2}{N_0\sin^2\theta}\right)\right]$$
$$\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s3}^2}{N_0\sin^2\theta}\right)\right]$$
$$+4F_1\left[\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{sd}^2}{N_0\sin^2\theta}\right)\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/2\right)\delta_2^2}{N_0\sin^2\theta}\right)\times\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/2\right)\delta_3^2}{N_0\sin^2\theta}\right)\right]$$
$$\times F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s1}^2}{N_0\sin^2\theta}\right)$$
$$\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s2}^2}{N_0\sin^2\theta}\right)\right]$$
$$\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s3}^2}{N_0\sin^2\theta}\right)\right]$$
$$+F_1\left[\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{sd}^2}{N_0\sin^2\theta}\right)\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/4\right)\delta_1^2}{N_0\sin^2\theta}\right)\times\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/4\right)\delta_2^2}{N_0\sin^2\theta}\right)\times\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/4\right)\delta_3^2}{N_0\sin^2\theta}\right)\times\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/4\right)\delta_4^2}{N_0\sin^2\theta}\right)\right]$$
$$\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s1}^2}{N_0\sin^2\theta}\right)\right]$$
$$\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s2}^2}{N_0\sin^2\theta}\right)\right]$$
$$\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{s3}^2}{N_0\sin^2\theta}\right)\right]. \tag{30}$$

If all channel link variances of source to relay (δ_{sr}^2) and relay to destination (δ_{rd}^2) are appropriately balanced, we can replace δ_{s1}^2, δ_{s2}^2, and δ_{s3}^2 by δ_{sr}^2, and δ_1^2, δ_2^2, δ_3^2, and δ_4^2 by δ_{rd}^2. We can obtain the SER expression as follows:

$$Pe_{\mathrm{total}} = 64F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{\mathrm{sd}}^2}{N_0\sin^2\theta}\right)\left[F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{\mathrm{sr}}^2}{N_0\sin^2\theta}\right)\right]^3$$
$$+48F_1\left[\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{\mathrm{sd}}^2}{N_0\sin^2\theta}\right)\left(1+\frac{b_{\mathrm{PSK}}P_q\delta_{\mathrm{rd}}^2}{N_0\sin^2\theta}\right)\right]\times\left[F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{\mathrm{sr}}^2}{N_0\sin^2\theta}\right)\right]^2\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{\mathrm{sr}}^2}{N_0\sin^2\theta}\right)\right]$$
$$+12F_1\left[\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{\mathrm{sd}}^2}{N_0\sin^2\theta}\right)\times\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/2\right)\delta_{\mathrm{rd}}^2}{N_0\sin^2\theta}\right)^2\right]\times F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{\mathrm{sr}}^2}{N_0\sin^2\theta}\right)\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{\mathrm{sr}}^2}{N_0\sin^2\theta}\right)\right]^2$$
$$+F_1\left[\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{\mathrm{sd}}^2}{N_0\sin^2\theta}\right)\times\left(1+\frac{b_{\mathrm{PSK}}\left(P_q/4\right)\delta_{\mathrm{rd}}^2}{N_0\sin^2\theta}\right)^4\right]\times\left[1-4F_1\left(1+\frac{b_{\mathrm{PSK}}P_d\delta_{\mathrm{sr}}^2}{N_0\sin^2\theta}\right)\right]^3. \tag{31}$$

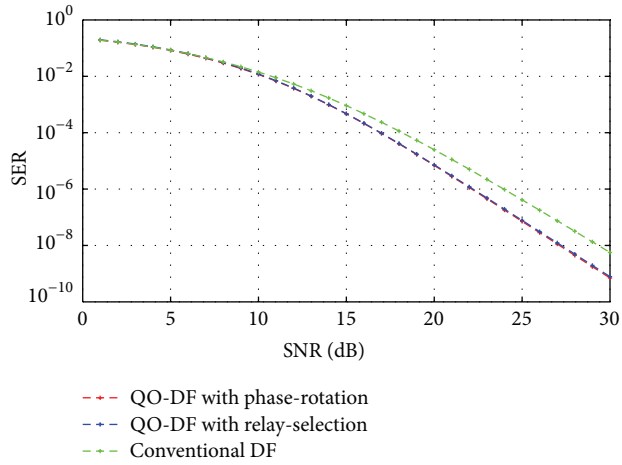


Figure 6: SER performance comparison between the proposed QO-DF system and the conventional DF system.

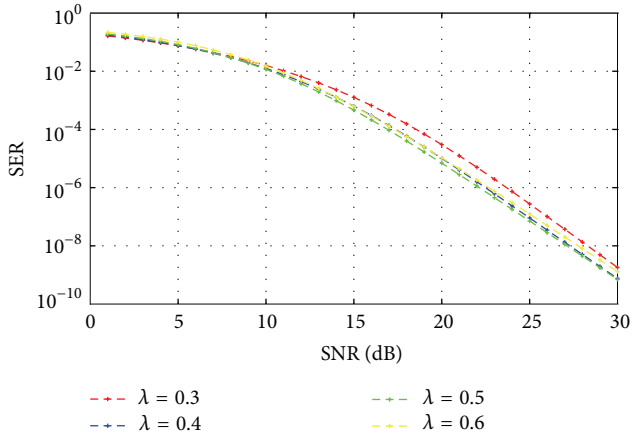


Figure 7: SER performance comparison of the proposed QO-DF systems with various values of $\lambda(\delta_{sr}^2 = \delta_{rd}^2 = 1)$.

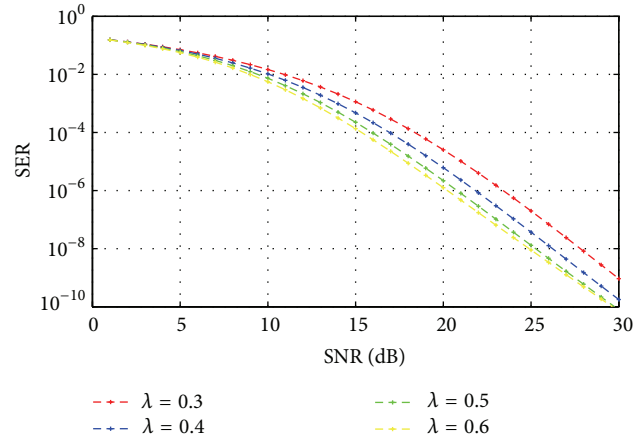


Figure 8: SER performance comparison of the proposed QO-DF systems with various values of $\lambda(\delta_{sr}^2 = 10, \delta_{rd}^2 = 1)$.

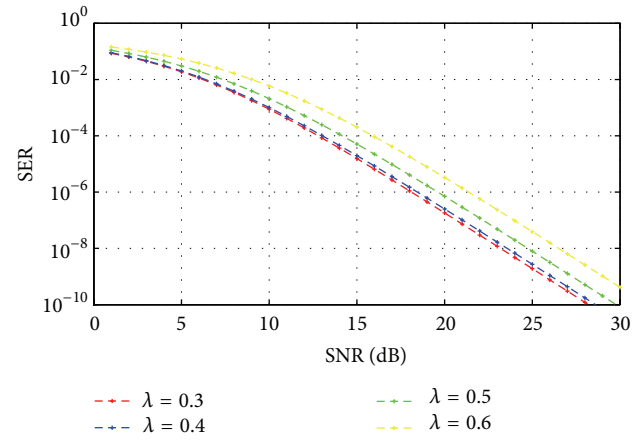


Figure 9: SER performance comparison of the proposed QO-DF systems with various values of $\lambda(\delta_{sr}^2 = 1, \delta_{rd}^2 = 10)$.

Specifically, from the SER approximation in (31), we observe that the link between source and destination contributes diversity order of one in the system performance. The cooperation strategy in the second phase also contributes diversity order of four in the system performance. In addition, it depends on the balance of the four channel links from the source to the relays and from the relays to the destination. Therefore, the proposed QO-DF cooperation systems show an overall performance of diversity order of four.

5.2. Optimum Power Allocation. It is common in cooperative communications that the channel variances between source to destination link, source to relay link, and relay to destination link are independent of each other. The MRC expressions in (19) cannot guarantee the maximum of total SNR at the destination. Hence, the power allocation objective is to minimize the approximated SER with respect to users' power, subject to a fixed total power constraint. The concept of power allocation is that the quality of the decoded symbols $\widetilde{s_1}, \widetilde{s_2}, \widetilde{s_3}$, and $\widetilde{s_4}$ greatly depends on the channel variances of both phase I and phase II. If we define the transmit power P_d for the source and P_q is the transmit power for the relays, for a fixed total transmission power of $P_d + P_q = P_t$, where P_t is the total transmit power, then we can write the power allocation condition as

$$P_t = (1 - \lambda) P_d + \lambda P_q, \quad \text{at } 0 < \lambda < 1, \tag{32}$$

where λ is a power allocation factor between P_d and P_q.

According to the SER expression in (30), we are able to heuristically search for optimum power allocation by replacing P_q with λP_d and P_d with $(1 - \lambda)P_d$. In our study, we use computer simulations to validate the optimum power allocation concept.

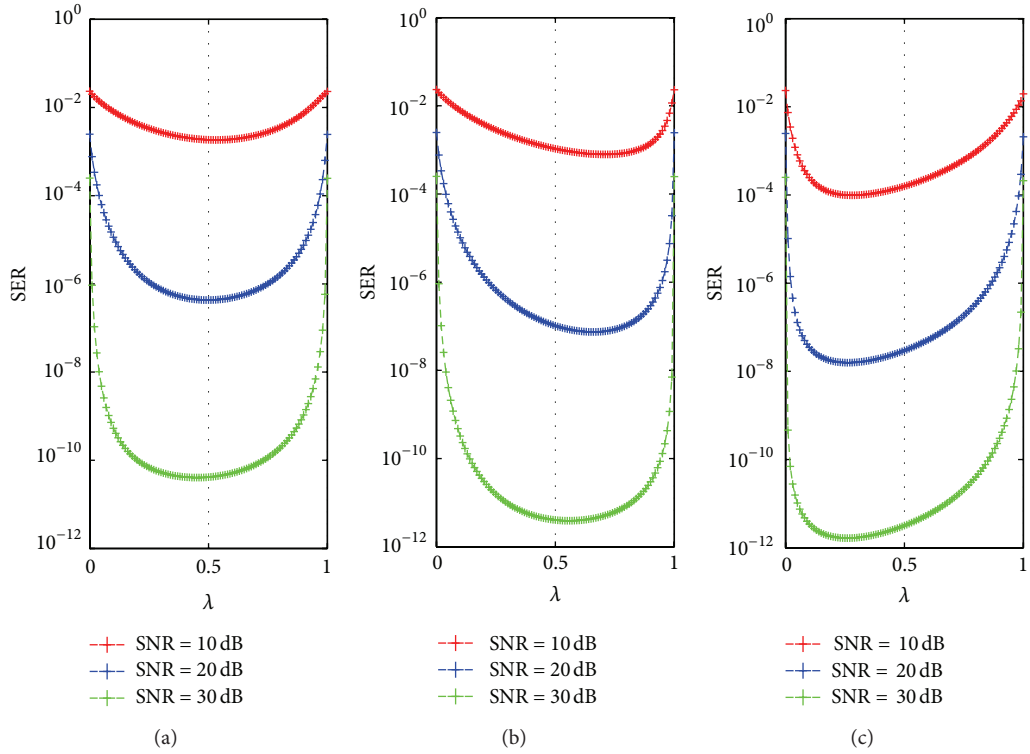


Figure 10: SER performance comparison of QO-DF system with various values of λ (a) $\delta_{sr}^2 = 1$, $\delta_{rd}^2 = 1$ (b) $\delta_{sr}^2 = 10$, $\delta_{rd}^2 = 1$ (c) $\delta_{sr}^2 = 1$, and $\delta_{rd}^2 = 10$.

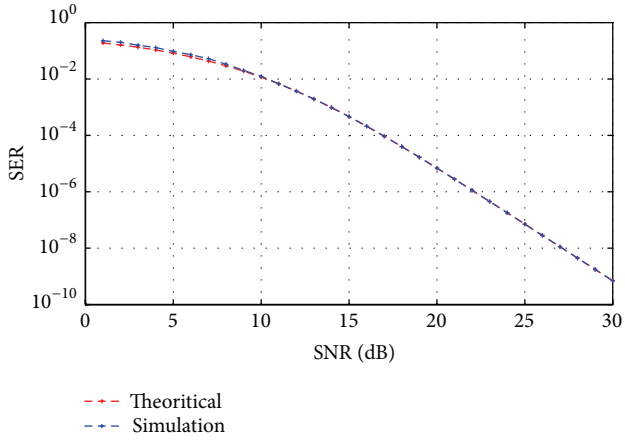


Figure 11: SER comparison of the proposed QO-DF system between theoretical analysis and simulation result.

6. Simulation and Results

In this section, based on computer simulations by MATLAB software, performance evaluations of the proposed QO-DF protocols are examined. For the sake of comparison, the conventional DF protocol for four communication nodes is also tested. The BPSK modulation with a total transmitted power has average energy 1, a variance of a noise is N_0 and bandwidth efficiency is 1 bit/s/Hz. In addition, Jake's model [13] is employed with a normalize doppler shift of 5,000 Hz for simulating Rayleigh fading channels, and we also assume all channel link variances in the system as appropriately balanced, that is, $\delta_{sr}^2 = \delta_{sd}^2 = \delta_{rd}^2 = 1$.

Table 2: Cooperative protocol code rate.

Cooperation protocol	Code rate
Direct signal only (noncooperative)	1
1-relay cooperative DF	1/2
2-relay cooperative DF	1/3
3-relay cooperative QO-DF	1/2

Figure 6 shows that SER performance of the proposed QO-DF system is better than the conventional four communication nodes DF system. Both of the proposed QO-DF systems, with relay-selection and phase-rotation techniques, have the SNR difference in comparison with the conventional DF system, specifically, 0.8 dB and 0.9 dB at BER of 10^{-5}, and then achieve 2.7 dB and 2.8 dB SNR difference at BER of 10^{-7}, respectively. The code rate of the proposed QO-DF system is shown in Table 2. In addition, the proposed QO-DF system will achieve a code rate two times higher than that of conventional DF in the case when no relay incorrectly decodes the symbols.

Next, we studied the effect of the channel qualities between source to relay and relay to destination for the optimum power allocation strategy. Figures 7–9 show the SER simulation results of the proposed QO-DF system with λ changing in the range of 0.3 to 0.6.

Figure 7 shows that, when channel variances of source to relay are equal to channel variances of relay to destination, that is, $\delta_{sr}^2 = \delta_{rd}^2$, at SNR < 7 dB, $\lambda = 0.3$ gives the lowest results of SER, and, at SNR > 7 dB, $\lambda = 0.5$ gives the lowest results of SER. The best average SER result for whole SNR range is at $\lambda = 0.5$.

Figure 8 shows that, when channel variances of source to relay are higher than channel variances of relay to destination, that is, $\delta_{sr}^2 \gg \delta_{rd}^2$, $\lambda = 0.6$ gives the lowest results of SER.

Figure 9 shows that, when channel variances of source to relay are lower than channel variances of relay to destination, that is, $\delta_{sr}^2 \ll \delta_{rd}^2$, $\lambda = 0.3$ gives the lowest results of SER.

We can summarize the strategy of power allocation to the proposed QO-DF system as follows. If the link qualities of source to relay are higher than the relay to destination, in (32), P_d goes to 0, and P_q goes to P_t. This implies that we should put more power at the relay nodes and less power at the source node. On the other hand, if the link qualities of source to relay are lower than those of the relay to destination link, P_d goes to P_t and P_q goes to 0. This implies that we should use almost all the power P_t at the source node, and use less power at the relay nodes. In addition, when the link qualities are approximately equal, we should put almost equal power at the source and the relay nodes.

For a high SNR case, we can observe the effect of the channel qualities on the power allocation strategy, as in Figure 10, by plotting the exact SER as a function of λ at SNR = 10 dB, 20 dB, and 30 dB with (a) $\delta_{sr}^2 = 1, \delta_{rd}^2 = 1$ (b) $\delta_{sr}^2 = 10, \delta_{rd}^2 = 10$, and (c) $\delta_{sr}^2 = 1, \delta_{rd}^2 = 10$. They show that the optimum BER results of all the different channel variances are not much different at $\lambda = 0.5$. Therefore, it is reasonable to adopt the equal power allocation scheme, that is, $\lambda = 0.5$, as a suboptimum power allocation, which in turn results in a simple power allocation strategy in the case of no available channel feedback.

In Figure 11, we present an SER comparison of the proposed QO-DF with phase-rotation technique between the theoretical SER and the simulation SER. These curves show that the theoretical result performs close to the simulation result.

7. Conclusion

In this paper, we have proposed a QO-DF cooperative communication system for four communication nodes in wireless cooperative communications, which could be well applied to wireless ad hoc networks. We also derived the theoretical SER, and compared the results with the simulation results. The theoretical SER shows a closed result to the simulated results. Furthermore, the proposed system achieves the full diversity of four by virtue of increasing several signal transmissions in the relaying phase. The optimum power allocation has also been investigated. In addition, it turns out that an equal power allocation could be used as a suboptimum power allocation for a slight SER degradation penalty. From simulation results, we can observe that the performance of the proposed schemes is significantly better than the conventional DF protocol. Another advantage of the proposed scheme is that it uses less time for signal transmission in the relaying phase so that the code rate is two times higher than the conventional DF system. Hence, this proposed protocol is suitable for future multimedia wireless communication.

References

[1] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.

[2] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[3] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, 1998.

[4] H. Jafarkhani, "A quasi-orthogonal space-time block code," *IEEE Transactions on Communications*, vol. 49, no. 1, pp. 1–4, 2001.

[5] K. J. Ray Liu, K. Ahmed Sadek, and W. Su, *Cooperative Communications and Networking*, Cambridge University Press, 2009.

[6] D. G. Brennan, "Linear diversity combining techniques," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 331–356, 2003.

[7] C. B. Papadias and G. J. Foschini, "A space-time coding approach for systems employing four transmit antennas," in *Proceedings of the IEEE Interntional Conference on Acoustics, Speech, and Signal Processing*, pp. 2481–2484, Salt Lake City, Utah, USA, May 2001.

[8] S. Lambotharan and C. Toker, "Closed-loop space time block coding techniques for OFDM based broadband wireless access systems," *IEEE Transactions on Consumer Electronics*, vol. 51, no. 3, pp. 765–769, 2005.

[9] P. Phenpakkool, N. Sutthisagiam, and C. Pirak, "Quasi-orthogonal space-time-coded protocol for eight-user cooperative communications," in *Proceedings of the 8th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON '11)*, pp. 397–400, Khon Kaen, Thailand, May 2011.

[10] M. K. Simon and M. S. Alouini, "A unified approach to the performance analysis of digital communication over generalized fading channels," *Proceedings of the IEEE*, pp. 1860–1877, 1998.

[11] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[12] C. Pirak, Z. J. Wang, and K. J. R. Liu, "An adaptive protocol for cooperative communications achieving asymptotic minimum symbol-error-rate," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, pp. IV53–IV56, Toulouse, France, May 2006.

[13] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, USA, 4th edition, 2000.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 860415, 10 pages
doi:10.1155/2012/860415

Research Article

Integrated Extensible Simulation Platform for Vehicular Sensor Networks in Smart Cities

Xiaolan Tang,[1,2] Juhua Pu,[1,2] Ke Cao,[1,2] Yi Zhang,[1,2] and Zhang Xiong[1,2]

[1] *State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China*
[2] *Research Institute of Beihang University in Shenzhen, Shenzhen 518057, China*

Correspondence should be addressed to Xiaolan Tang, tangxl@cse.buaa.edu.cn

Received 31 July 2012; Revised 20 November 2012; Accepted 22 November 2012

Academic Editor: Shan Lin

This paper presents an integrated extensible simulation platform BHU-VSim for vehicular sensor networks (VSNs), which aims to support general simulation environment for typical vehicular applications in smart cities. To deploy urban traffic scenario, we propose a hierarchical object structure to manage entities with different movement models in the network. Furthermore, we design a general data container to present different kinds of data transferred and provide two schemes to generate data packages. Regarding transmission control, the platform includes three components: task scheduling, storage management, and routing deployment. We support importing external routing protocols and configuring relevant parameters to satisfy various transmission requirements. Finally, an instance application, real-time traffic monitoring, and an example of statistical analysis are introduced to prove the practicality and accuracy of the simulator. In one word, as an initiate attempt, our platform provides significant improvement of VSNs' simulations.

1. Introduction

A vehicular sensor network (VSN) is a kind of network with sensors equipped on fast moving vehicles and provides ubiquitous connectivity among mobile users and efficient vehicle-to-vehicle (V2V) communications. It is widely used in intelligent transportation systems (ITSs) to support various applications, such as safe driving, real-time traffic monitoring, highway toll payment, multimedia resource sharing, and urban mobile surveillance [1]. As vehicles broaden and deepen the extent of data transmission, VSN has been considered as an important improvement in information collection by humans and has attracted increasing research in both academia and industry [2, 3]. Compared with traditional wireless sensor networks, VSN has its own characteristics, such as loose energy constraint, dynamic network topology, unstable connectivity, and geography-based communication patterns.

Nowadays, a brand new theory, interconnecting things together to form a ubiquitous internet of things (IoTs) [4], has been put forward. IoT initiates many new research domains, including smart city. One typical application in smart cities is smart traffic, which aims to avoid traffic jams, save energy resources, and reduce vehicles' emissions. Although there already exist ITSs in many metropolises to support real-time traffic monitoring, they require many infrastructures, like base stations, lengthy cables, reliable data center, and others. Considering the large overhead to establish such an ITS and its harsh demand about the environment, the concept of IoT motivates us to do research on internet of vehicles (IoVs) and implement traffic services by VSNs with few infrastructures, which is a key application in future smart cities.

In urban VSNs, because of high mobility and small communication radius, connectivity between vehicles cannot have a long duration and end-to-end paths seldom exist. Thus, we consider VSN as a special kind of delay tolerant networks (DTNs) [5] rather than vehicular ad hoc networks (VANETs) [6]. By providing DTN capabilities, some typical challenging situations of vehicular sensor networks may be overcome, such as intermittent connectivity, variable delays, high error rates, and nonexistence of an end-to-end path. That is because DTNs introduce a store-carry-forward paradigm that performs better and uses fewer resources than

those of end-to-end protocols, for each hop of DTNs is optimized individually [7].

As VSN is an emerging field with new variety and fast-moving features, detailed research work is urgent on data representation, data storage, routing protocols, and other aspects. Because experiments on real vehicles are expensive and dangerous, due to high resource consumptions and complicated traffic scenarios, a reliable and functional platform for VSNs' simulation is starved for. Such a simulation platform not only provides simulation scenario for VSNs' applications, but also helps to narrow the gap between research work and practical requirements.

However, to the best of our knowledge, little research had been done on a versatile simulation platform for VSNs to integrate DTN specialties. In this paper, we do research on the core problems in VSNs and construct an extensible simulation platform BHU-VSim. It provides a general environment integrated with scenario construction, data management, transmission control, and service optimization. It not only embeds several existing mobility models and routing protocols, but also supports interfaces to import external models and algorithms. Furthermore, to improve the practicality of the simulator, it supports flexible scenario control, data generation, task scheduling, and package deletion methods. Meanwhile, graphical user interface (GUI) is finely designed to improve user experience. Overall, different applications on VSNs can be easily realized on our platform with relevant configurations. In this paper, we take real-time traffic monitoring in urban scenario as an example, showing how to use our platform.

The rest of this paper is organized as follows. Section 2 presents related work in VSNs and existing simulation platforms. In Section 3, we introduce the design objectives and system framework of the platform. Sections 4, 5, and 6, respectively, discuss core issues in the platform design, including scenario construction, data management, and transmission control. An instance of real-time traffic monitoring and an example of statistical analysis are depicted in Section 7. Finally, with several improvements discussed for our future work, we conclude the paper in Section 8.

2. Related Work

As a rising hotspot in traffic management, VSN has attracted researchers in academia, industry, and even government. Well-known research institutes and programs include Car2Car-CC by BMW and other famous automobile corporations [8], CarTel by MIT [9], German's Network on Wheels (NOWs) [10], and Traffic Prediction by IBM. The Federal Communications Commission (FCC) of the US segments a dedicated short range of communication frequency (DSRC) for intervehicle communications. Research work on VSNs focuses on several aspects, such as network framework, mobility models, routing protocols, storage management, network security, and performance analysis [11, 12]. Now, a general and flexible platform to integrate these research domains and evaluate overall performance is highly required.

In VANETs, there already exist several typical simulators, such as GrooveSim [13], TraNS [14], ASH [15], VGSim [16], iTETRIS [17], NCTUns [18], and Veins [19]. Most of them are composed of network simulator, traffic simulator, and even environment simulator. Typical network simulators are ns-3 [20], ns-2 [21], OMNet++ [22], Jist/SWANS [23], V2X, OPNET, and so forth. Typical traffic simulators are SUMO [24], CanuMobiSim [25], VanetMobiSim [26], and so forth. These simulators import car-following, lane-changing, traffic-queuing, and vehicle interaction models to improve the fidelity of the simulation results. Besides, ASH and Veins utilize bidirectional interactions between mobility model and network model to support real-time path reselection during moving process. GrooveSim supports hybrid between simulated and real vehicles. NCTUns, from 2002 to 2010, has published six versions to gradually improve its simulation performance. However, each simulator has its drawbacks. For example, ns-2 has scalability problems, and GrooveSim just provides geographic routing protocols, and NCTUns is difficult to extend. Besides, a general issue of these simulators is that they all work for VANETs, which assume that end-to-end connectivity exists through some path, while VSNs do not always satisfy this assumption [27]. Therefore, we attempt to construct a simulator for VSNs based on DTNs, which also accepts sparse networks through its store-carry-forward paradigm.

Nowadays, most simulators for DTNs are still in progress. One well-known simulator is named opportunistic network environment simulator (ONE), which is constructed by a research group in Helsinki University of Technology [28, 29]. ONE supports embedded and imported routing protocols and mobility models and provides GUI and a set of reporting and analyzing modules. Although ONE is an excellent simulation platform for DTNs, there is still much improvement needed to increase the practicality of the platform, such as data representation and transmission control. In addition, it is not appropriate to simulate VSNs, due to VSNs' own characteristics, such as vehicular scenario constraints and highly dynamic mobility. Thus, based on ONE, we design our simulation platform to realize the expectations for actual applications in VSNs.

In this paper, we construct an innovative platform BHU-VSim for VSNs' simulations. It is integrated with several classic algorithms and also supports abundant interfaces for external algorithms, in the domains of mobility model, data representation, routing protocol, and so forth. Meanwhile, we design flexible methods to support system management, such as scenario control, data generation, task scheduling, and package deletion. Thus, our proposal can provide better performance, compared with previous simulators.

3. Platform Architecture

3.1. Design Objectives. From the perspective of applications, main tasks of VSNs are data collection, data management, transmission control, and finally providing timely and reliable services for smart traffic. In this way, VSN can be considered as a mobile network of massive distributed

heterogeneous data. The difficulties to design VSN simulator are complex scenario setting, various data representation, and flexible control mechanisms. In short, we list several objectives in the following for our platform to satisfy.

(1) Support complex traffic scenario construction, with well-managed hierarchy of traffic participants and fine scenario regulations.

(2) To simulate data collection and transmission in VSNs, the platform includes interfaces to import external algorithms.

(3) Users can neatly configure the control schemes according to specific application requirements.

(4) Have access to real traffic data from transportation statistics, to analyze and optimize the system design.

3.2. System Framework. In smart cities, as the basic unit, data cell has its life cycle, from data generation to data processing, then to data usage, until data death [30]. From data life cycle standpoint, we design the system framework of our simulator. First of all, we need an actual urban traffic scenario to support environment for data growing. We name this basis as scenario construction module. In data generation process, we use portable and stable sensors to collect information, so we set data collection layer at the bottom of the architecture. Besides, for specific rules about data formats and generation patterns required to guarantee expected data generation, the second lowest layer is data management layer, including data representation and data generation. After data generation, in order to successfully transmit data from one vehicle to another, vehicular transmission control mechanisms strictly influence data processing, including data storage, data transfer, and data update. Thus, this layer is named transmission control layer. When the data reaches its destination, it must be resolved and analyzed according to the requirements of a specific smart service, in order to provide appropriate data for the ultimate use. We name this layer smart service layer. Finally, the target terminal utilizes the data and decides on whether to delete or keep it, which is named terminal application layer. To sum up the previous analysis, VSN system can be segmented into five layers, data collection layer, data management layer, transmission control layer, smart service layer, and terminal application layer, as shown in Figure 1.

In BHU-VSim, we design a hierarchical object structure to construct scenarios with fast-moving vehicles, slow-moving pedestrians, static roadside infrastructure, and other entities. In data collection layer, sensors equipped on vehicles, roadside infrastructures, and mobile devices with pedestrians gather surrounding information. In data management layer, various data representation and flexible data generation are discussed. Transmission control layer contains task scheduling, storage management, and routing deployment components. Smart service layer supports basic VSNs' services, such as safe driving and traffic management. The top layer is terminal application layer, which satisfies different requirements of different terminals. Besides scenario setting and performance evaluation, our research mainly covers the data management, transmission control and smart service layers. In our platform, we focus on four modules as listed in the following.

(1) Scenario construction: to construct vehicular network scenarios, we introduce the digital map pattern to load instance map, and manage entities in a hierarchical structure to support flexible entity loading. Each entity has its characteristics and can adjust its movement states.

(2) Data management: this module consists of two parts, data representation and data generation. Firstly, we design a general data container to present different kinds of packages transferred in the network. Besides, the platform allows periodical or random generation of data packages.

(3) Transmission control: BHU-VSim provides task scheduling methods and package dropping methods to improve transmission and storage efficiency. Meanwhile, it is integrated with several typical routing protocols and has interfaces to import external protocols.

(4) Service optimization: analyzing basic information of the network and data received from traffic entities, vehicles compute relevant parameters to indicate service quality and adopt corresponding measures to optimize the quality.

To be noted, service optimization is used to prepare rough data well for terminal applications. Via computing service parameters which serve for the applications and defining optimization methods to optimize the simulation results of service parameters, this module ultimately realizes the value of data and provides high-quality service. Here, we see service optimization module always has strong association with specific applications. For an instance, a multimedia resource sharing application may require selection and integration of different data blocks, while a traffic monitoring application maybe needs the latest vehicular flow rate at each waypoint. Different requirements lead to a huge difference between the implementations of service optimization modules in these two applications. Therefore, to introduce general models in our simulator, we omit further discussion about service optimization in the next sections. However, an instance of service optimization module will be depicted in Section 7. In Sections 4–6, we describe core issues in the implementation of scenario construction, data management, and transmission control modules in detail.

4. Scenario Construction

As traffic status in urban areas is always complicated, exactly simulating an urban traffic scenario is often difficult. Even previous work on VANET simulators could not provide adequate models and convenient extensions to create various transportation participants. Additionally, as ONE is not specially designed for vehicular networks, it misses several

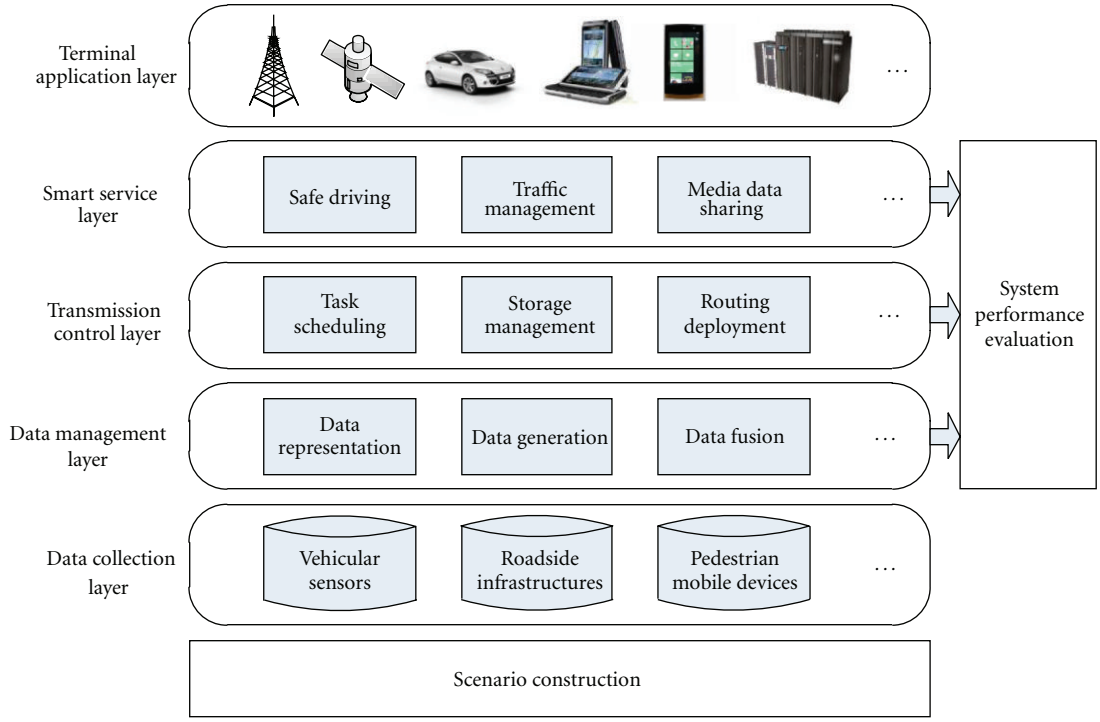


Figure 1: System framework.

important specialties about vehicular mobility, such as car-following and lane-changing. Considering the significance of an accurate scenario to achieve correct simulation results, great enhancements of scenario construction module are required in the simulator. In this paper, the scenario construction module has three parts, importing urban map, deploying traffic participants, and adjusting traffic situation.

4.1. Map Format. To simulate transportation in smart cities, we focus on the road information in the urban map rather than buildings, farmland, and so forth. Thus, each map is composed of many paths, and each path has a list of road waypoints. Similar with ONE, we select .wkt digital map file to be the standard map formats. To better simulate roads in cities, we add multilane and directions for .wkt file. Specifically, we allow users to set the numbers of lanes in two directions in the map file. Default numbers of lanes in two directions are both 1. The direction in line with the order of road waypoints is named positive direction, remarked "+." The other direction is called negative direction, remarked "−." The serial number of each lane is started from the roadside (1) to the middle of the path (n). Each lane has its lane number, such as "+1" and "−2." To be noted, real city maps in .wkt format can be imported into the system directly.

4.2. Entity Management. To support unified and extensible traffic entity management, we propose a hierarchical entity structure. Entities with similar characteristics comprise one group with the common attributes, and similar groups are also clustered together, in order for convenient management and deployment. The system automatically constructs entity groups in batch and then regulates entities with special features different from its group. Each entity has three kinds of attributes, including basic attributes (*ID*, *name*, *count*, etc.), mobility attributes (*mobilityModel*, *initialLocation*, *path*, *speed*, *waitTime*, *movingTime*, etc.), and communication attributes (*commRange*, *dataRate*, *bufferSize*, *router*, *packages*, etc.). Users can customize these attributes for a particular application in the class file *Entity*.

BHU-VSim is embedded with several movement models for traffic participants, such as *CarMovementModel* and *BusMovementModel*. To improve flexibility of the platform, external movement models are supported by extending *MobilityModel* class and overriding at least two methods *setInitialLocation()* and *setPath()*. Additionally, actual mobility files can be used in the system by calling methods with parameter *mobilityFile*.

Besides, in BHU-VSim, we implement IDM car-following model [31] and MOBIL lane-changing model [32] to simulate vehicular mobility constraints. With configuration sentences *carFollowingModel = IDM* and *laneChangingModel = MOBIL* in the mobility attributes, users can use these models.

4.3. Traffic Status Adjustment. To improve realness of BHU-VSim, we use traffic status adjustments to set jams in the scenario. We propose the configuration of influence sphere to improve the simulation accuracy, compared with obstacles defined in ASH [15]. For instance, a traffic control measure may just stop private cars passing at one direction. Relevant methods to set a block are shown in Table 1. Among

Table 1: Interface methods to set a block.

Usage	Class	Method	Parameter
Calling	*Block*	*setPaths()*	Path[]*p*
		setLane()	Lane[]ln
		setEntities()	Groups[] *entities*
		setSpeedRange()	Float *lowSpeed* Float *highSpeed*
Associating	*Scenario*	*addBlock()*	Block *b*

the methods, *setPaths()* and *setEntities()* are required. The default lanes are all lanes, and the default speed is 0.

The scenario construction module of BHU-VSim supports urban map with multilane paths, has a hierarchical structure to manage transportation participants, and also allows setting road blocks. Although these specialties help to construct more real traffic scenario, there are still many problems to be solved, such as the traffic light model, the overlapping issue due to overpass, and the building's interference with V2V communication. More actual 3D model may be helpful to overcome these problems in the future.

5. Data Management

Although previous network simulators seldom discuss the data management program, it is very significant to define data when simulating an actual application in VSNs, for data being the key element in the network communication. In BHU-VSim, we provide a general data container to represent different kinds of data packages for different applications. Moreover, we design two methods to support periodical generation and random generation of data packages.

5.1. Data Representation. To improve the flexibility and availability of BHU-VSim, we design a general data container to present and differentiate various data. We partition a package into header section and data section. For a specific application, the data package can be defined by inheriting class *DataPackage*, setting necessary attributes, and overriding the method *Entity.createPackage()* to assign values for the attributes in the package.

In our simulator, we have designed three kinds of data packages for typical services in VSNs, entity status data, digital data, and multimedia data. Entity status data presents this entity's basic information, moving state, and position relations with others, in order to monitor traffic and support safe driving. Digital data is always taken from different kinds of sensors to record temperature, humidity, and so forth. These sensors use digital to indicate the traffic states and road states. Multimedia data is designed for entertainment applications, including file format, minimum data rate, recovery rate, and content.

Besides data messages, control messages are required in almost every application. They always have particular requirements about transmission and analysis. Thus, we leave the discussions about control messages for our future work.

5.2. Package Generation. In BHU-VSim, we provide two interfaces to periodically or randomly generate data packages. By calling the method *setProdPkgGntTime(float start, float end, float interval)*, this entity will generate one data package every *interval* rounds from the round *start* to *end*. By calling methods *setRandPkgGntTime(float start, float end, int count)*, this entity will randomly generate *count* data packages from *start* to *end*. Although these methods are not complete, they are sufficient for usual data generation.

In BHU-VSim, we define the format of data package and present three typical instances. Besides, two data generation methods enrich the simulation scope. In this module, there are much more research needed on the control package formats and simplified data structure design, in order to improve the accuracy and efficiency of the simulation.

6. Transmission Control

How to successfully transfer a data package from its source to its destination is a key problem in network simulation. In VSNs, considering the unstable connection, when two entities meet with each other (they are in each other's communication range), they should grasp this valuable opportunity to transfer data between them. In previous work, although some simulators are embedded with classical routing protocols and storage management policy, they are incomplete and difficult to extend. After analyzing the influence factors in data transmission, we think that three basic problems should be solved: (1) which entity transfers first, and which package is transferred early; (2) what routing protocol to use for the package transmission; (3) if no enough storage left to receive the data, which package is dropped early to save space. To handle these issues in an extensible way, we introduce the task scheduling, storage management, and routing deployment policies.

6.1. Task Scheduling. To provide equal chance for each entity to transfer data, we propose random entity selection policy to select one meeting entity to send data first. Then, we use scheduling model to compute the transmission priority of each package in the sender entity, to improve the transmission efficiency.

Using the random entity selection policy, each meeting entity generates a random value in the interval (0, 1). Then, the entity with the largest value transfers first, and the entity having the second largest value transfers next, until each of them transfers one package. If there is still a transfer chance, the second round of transmission with the same sequence takes place. In this way, each entity has an equal chance to send data. Here we assume that all the nodes obey this policy, without mendacity. How to realize this assumption is left for research on network security.

When one entity has a chance to transmit data, firstly it computes scheduling priorities of all the packages in its storage and then selects the package having the highest priority to transfer. Here, we use basic scheduling attributes to stand for those attributes defined in the *ScheduleModel*

class. The method *computePriority()* in this class uses the values of these attributes to compute the scheduling priority.

The customized scheduling model should inherit from the class *ScheduleModel*, setting basic scheduling parameters and overriding the *computePriority()* method.

Traffic entities use the method *setScheduleModel()* to set scheduling model for their packages and assign values to basic scheduling attributes in the model. Here, we suggest the configuration to be set in batch if one or more groups of entities have the same policy.

6.2. Storage Management. Taking VSN as a distributed storage system with limited storage space in each vehicle, when the buffer is overfilled, the vehicle drops data according to package dropping rule, in order to make space for new coming data.

To increase the flexibility of BHU-VSim, we design a package dropping model based on selected attributes. The implementation is similar to the task scheduling policy. Basic dropping attributes are defined in the *DropModel* class, and the method *computePriority()* uses the values of these attributes to compute the dropping priority. Packages with higher priorities are dropped earlier. Traffic entities use the method *setDropModel()* to set drop model for their packages.

6.3. Routing Deployment. In urban traffic, each entity has its routing protocol to transfer data packages. For instance, the fast-moving vehicle may use *Prophet* protocol to transfer data to a specific destination, while road-side infrastructure may flood data to passing buses. Furthermore, to satisfy complicated requirements, we support flexible deployment according to traffic states. For example, in a crowded area, vehicles use *Spray and Wait* routing protocol to keep the number of package replica within a limit, while in some sparse area, *Epidemic* routing is selected to increase the probability of successful transmission. Thus, we support routing deployment based on different entity groups and traffic states.

Here, traffic state attributes are those attributes of the entity, which affect it to select routing protocol. Each entity sets its routing protocol with the method *setRouter()* by considering the values of these traffic state attributes.

BHU-VSim has been integrated with several typical routing protocols, including *EpidemicRouter*, *SprayWaitRouter*, and *ProphetRouter* classes. The interface to import new routing protocols is also provided. We use basic routing attributes to stand for those attributes defined in a new routing class, and the method *update()* in the class uses the values of these attributes to make routing decision, such as whether to transmit this message and how many replica to transmit (the usage of the number of replica can be obtained from *SprayWaitRouter* class). Users can create a new *Router* class inheriting the *BasicRouter* class, define the basic routing attributes, and override the method *update()*, to implement customized routing decision making process.

Besides the control methods described previously, we have concrete operation process in the core of the platform. In other words, user-defined control mechanisms are well integrated into BHU-VSim, if they are configured using the interface methods as defined previously. Although our improvement has great significance to simulate actual applications in VSNs, there are still more problems about transmission control to be handled, such as data inconsistency, data corruption, and data acknowledgment. Besides, to simplify and clarify the structure of the simulator, we omit the handshake process when establishing connection. In our future work, we should gradually improve the fidelity of the transmission process.

7. Performance Evaluation

7.1. An Instance: Real-Time Traffic Monitoring. As smart traffic is attracting more and more attentions from the government to individuals, we introduce one instance application, real-time traffic monitoring in VSNs, which aims to optimize vehicles' path selection to avoid traffic jams through V2V communication, without any roadside infrastructures.

In scenario construction module, to set urban traffic scenario for this application, we deploy two vehicle groups, *VGroup1* and *AGroup1*, in the 500×500 map zone during the simulation rounds $[0, 500]$. *VGroup1* has 100 cars moving with *CarMovementModel* movement model, and *AGroup1* has 12 buses with *BusMovementModel* mobility model. Here, buses in *AGroup1* gather traffic congestion information, because of their stable movement paths, fixed departure interval, and high network coverage. Cars in *VGroup1* just communicate with *AGoup1* to get the traffic state. In this way, the traffic state can be disseminated in the scenario with low transmission overhead. Here, we use our traffic status adjustment method to set one jam on the shortest path from the start point to the destination point. Configuration interfaces are listed in Table 2.

In data management module, we use the vehicle driving speed to indicate traffic congestion degree. Specifically, if the vehicle passing by one path runs fast, the traffic there is smooth and clear; if the vehicle can hardly move forward, there is possibly a traffic jam. In the implementation, we use entity status data to represent the data package. To save storage space, the entity status data just contains a quaternion, including recorder entity's *ID*, *location*, *velocity*, and *record time*. Concerning data generation pattern, each bus generates 1 package in every 5 rounds from the 0th round to the 400th round. Relevant methods are listed in Table 2.

In transmission control module, to improve transmission efficiency, we utilize task scheduling and package dropping policies. In the task scheduling policy, we use two attributes in data package *p*, *record time* and *location*, to be basic scheduling attributes. Packages with higher *record time* and more adjacent *location* are transferred earlier than others. In the package dropping policy, we use *record time* to be basic dropping attribute. Packages with lower *record time* are dropped earlier than others, to save space for new data. For the routing protocols, we use the embedded protocols *EpidemicRouter* and *ProphetRouter*. The *velocity* of entity is traffic state attribute. When the *velocity* is equal to or larger than 20, which means the traffic state is excellent, buses use

Table 2: System configurations for the instance application.

Module	Function	Method	Remark
Scenario construction	Set mobility models for entities	*VGroup1.setMobilityModel(Car-MovementModel)* *AGroup1.setMobilityModel(Bus-MovementModel)*	
	Set a traffic block *b*	*b.setPaths(path1, path2)* *b.setEntities(VGroup1, AGroup1)* *b.setSpeedRange(0, 5) scenario.addBlock(b)*	
Data management	Represent data package *p*	*AGroup1.createPackage()*	Data: *ID*, *location*, *velocity*, and *record time*
	Set generation time of data package *p*	*AGroup1.setProdPkgGntTime (0, 400, 5)*	
Transmission control	Set task scheduling policy *ts*	*VGroup1.createScheduleModel(ts)* *AGroup1.createScheduleModel(ts)* *ts.computePriority()*	Basic scheduling attributes: *record time* and *location*
	Set package dropping policy *pd*	*VGroup1.createScheduleModel(pd)* *AGroup1.createScheduleModel(pd)* *pd.computePriority()*	Basic dropping attributes: *record time*
	Set routing protocols for buses in *AGroup1*	*AGroup1.setRouter()*	Traffic state attributes: *velocity*

EpidemicRouter to propagate information; otherwise, when the traffic is jammed, they use *ProphetRouter* to decrease transmission overhead. Table 2 shows the configuration methods.

To avoid transmitting traffic states of irrelevant paths, buses just collect data about the 5 recommended shortest paths. Cars obtain the traffic information when they communicate with buses. Then, cars can optimize their path selection by analyzing the traffic states.

In service optimization module, with geographical data and traffic data, we put forward the path selection method. To avoid traffic jams, cars exclude the initial recommended paths including jams first and then select the one with the shortest path as the final path. However, maybe all the initial paths have jams. In this case, another 5 short paths are computed, and another round of jam rejection is implemented, until one path excluding jams is found. The GUI of BHU-VSim is designed to show the scenario and transmission state, as presented in Figure 2(a). The result of path selection to avoid jams is shown in Figure 2(b).

For other applications, users just need to configure the scenario, define the data, and override relevant control methods, in order to provide required services. Overall, we can conclude that the simulation platform BHU-VSim is easily configurable and extensible. It can satisfy different requirements of various applications in VSNs.

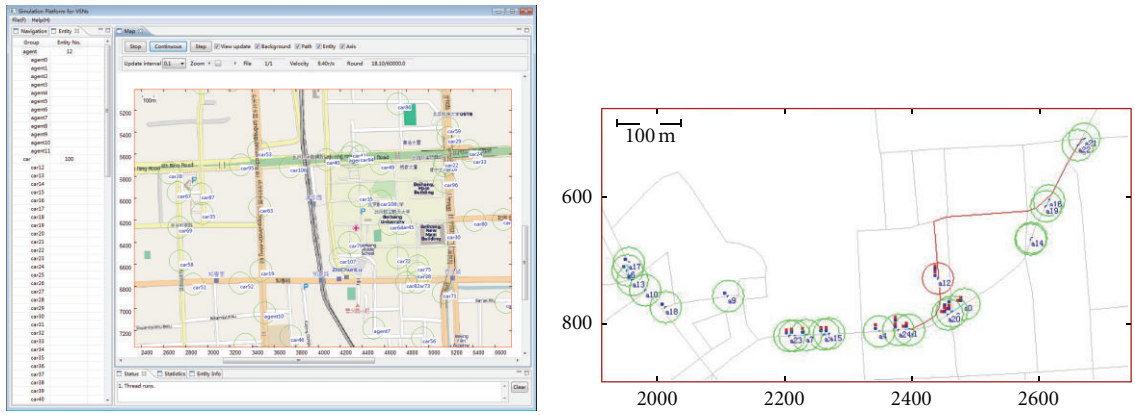
7.2. Statistical Analysis. In our simulator, we have data analysis module to compare the statistic results of different algorithms. Independent of the traffic monitoring instance, we test the performance of typical routing protocols under limited storage in VSNs, including *Epidemic*, *Prophet*, and *Spray and Wait* routing protocols. Configurations are listed in Table 3. We set the shortest path map-based movement model and FIFO package dropping model for vehicles.

Table 3: Configurations for the example of statistical analysis.

Parameter	Value
Network area	$5000 \times 5000\ m^2$
Number of vehicles	100
Simulation time	12 h (1 round stands for 1 second)
Communication radius	100 m
Mobility model	The shortest path map-based model
Mobility speed	Random in (10, 20) m/s
Data rate	2 Mbps
Size of data message	256 KB
Size of storage space	5 MB
Data generation	1 message every 15 min by each vehicle
Package dropping model	Fist in first out (FIFO)

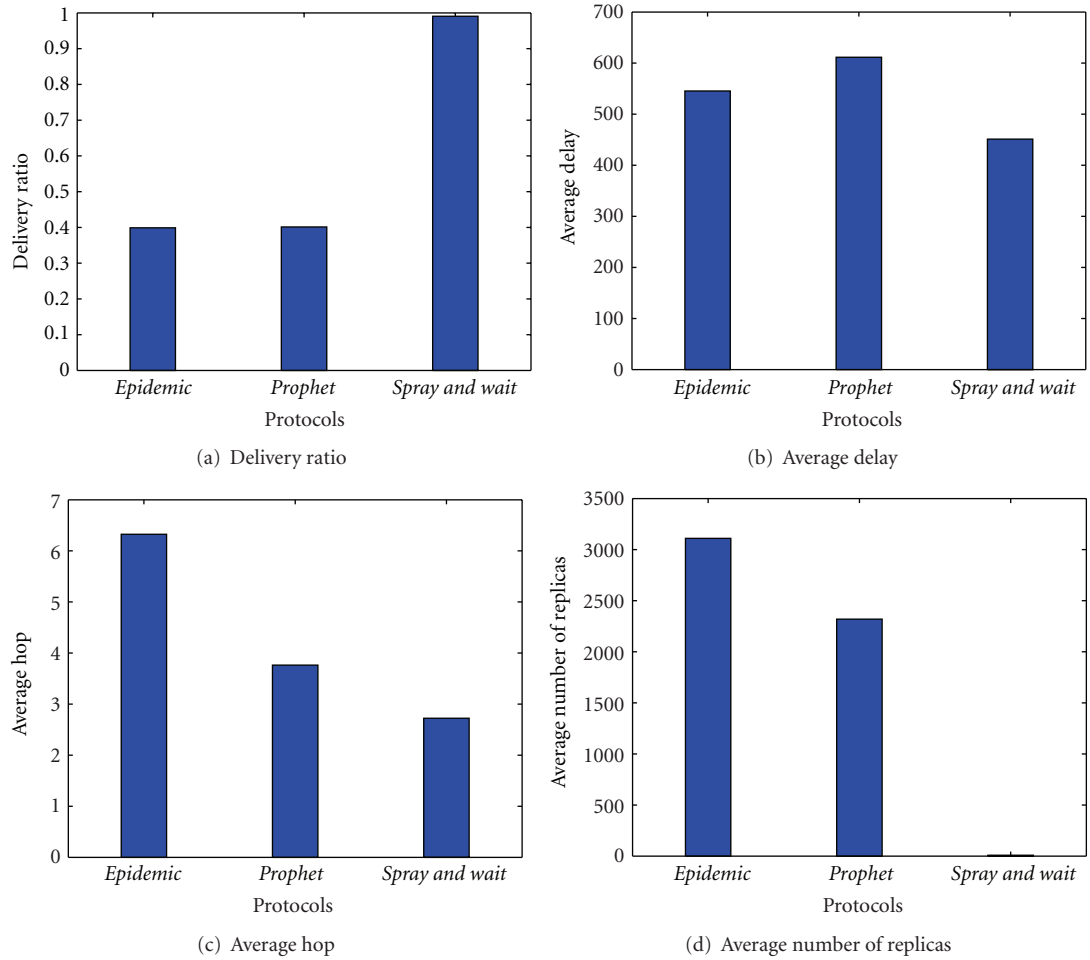
The evaluating indicators are the delivery ratio (ratio of the number of successful delivery messages to the number of total messages), average delay (average delay of successful delivery messages), average hop (average hop of successful delivery messages), and average number of replicas (ratio of the number of total replicas to the number of messages). The simulation results are shown in Figure 3.

From Figure 3, we see that in general, *Spray and Wait* routing protocol performs best among the three protocols. It has the highest delivery ratio, the lowest average delay, the lowest average hop, and the lowest average number of messages. The reasons lie in two parts. (1) As we use the shortest path map-based movement model, the advantages of *Prophet* cannot be shown out. This is because the mobility model causes irregular counter probabilities, which seriously interfere the forwarder selection in *Prophet* routing protocol. (2) The *Epidemic* routing protocol uses flooding method to disseminate data, so it leads to heavy resource consumption



(a) Main window (b) Path selection to avoid jams

Figure 2: GUI of the instance application.



(a) Delivery ratio (b) Average delay

(c) Average hop (d) Average number of replicas

Figure 3: Statistic results of different routing protocols.

of storage spaces. In this test, we set limited storage space at vehicles and use *FIFO* dropping rule to manage storage space. Thus, *Epidemic* leads to frequent change of the packages before they are forwarded. Thus, the performance of *Epidemic* routing protocol deteriorates.

From qualitative analysis, the simulation results are in line with theoretical derivation. Thus, it testifies the correctness of statistic results using our platform to some extent and further shows the practicality of our platform in algorithm performance comparison.

In this section, we introduce two typical instances. One is a real application, real-time traffic monitoring, in order to show the usage of our simulator and indicate the practicality and convenience of our platform. The other is statistical analysis of three routing protocols, in order to testify the accuracy of simulation results qualitatively. However, more actual experiments are required in our future work to further evaluate the performance of BHU-VSim quantitatively.

8. Conclusion

In this paper, we construct an integrated extensible platform named BHU-VSim, to support simulation environment for typical applications in VSNs. To deploy urban traffic scenario, we define standard map files with multilane paths, propose a hierarchical object structure to manage traffic entities, and provide traffic status adjustment to set jams in the scenario. Furthermore, to manage data, we define data package formats and support periodical and random data generation methods. Considering the key problems in transmission process, BHU-VSim supports flexible task scheduling policy, package dropping policy, and routing deployment method. Besides the embedded mobility models and routing protocols, it has abundant interfaces to import external models and protocols. Finally, an instance of real-time urban traffic monitoring application and an example of statistical analysis are introduced to illustrate the usage, practicality and accuracy of the simulator. In one word, BHU-VSim is a valuable attempt to construct an integrated and extensible simulation platform for VSNs.

However, there is still much profound research to be done to improve our simulator. As the channel model is crucial to simulate V2V communication, we will continue our research to integrate the models in physical layer and MAC layer into BHU-VSim. Additionally, more experiments in actual vehicular sensor networks are needed to test the simulator's performance and adjust it well for real-world scenarios.

Acknowledgments

The authors gratefully acknowledge the support from the Natural Science Foundation of China (61272350 and 61173009), National High Technology Research and Development Program of China (2011AA010502), International S&T Cooperation Program of China (2010DFB13350), Doctoral Fund of Ministry of Education of China (20091102110017), Science Foundation of Shenzhen City in China (JCYJ20120618170520900), and Fundamental Research Funds for the central universities.

References

[1] Y. Zhu and Y. Jian, "A game-theoretic approach to anti-jamming in sensor networks," in *Proceedings of the 16th IEEE International Conference on Parallel and Distributed Systems (ICPADS '10)*, pp. 617–624, December 2010.

[2] M. Gerla and L. Kleinrock, "Vehicular networks and the future of the mobile internet," *Computer Networks*, vol. 55, no. 2, pp. 457–469, 2011.

[3] X. Li, *Research on the key techniques of vehicular sensor networks and applications [Dissertation]*, Shanghai Jiao Tong University, 2009.

[4] L. Chen, M. Tseng, and X. Lian, "Development of foundation models for Internet of Things," *Frontiers of Computer Science in China*, vol. 4, no. 3, pp. 376–385, 2010.

[5] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: a comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 607–640, 2011.

[6] H. Hartenstein and K. P. Laberteaux, *VANET: Vehicular Applications and Inter-Networking Technologies*, John Wiley & Sons, 2010.

[7] P. R. Pereira, A. Casaca, J. J. P. C. Rodrigues, V. N. G. J. 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, 2012.

[8] L. Franck and F. Gil-Castiñeira, "Using delay tolerant networks for Car2Car communications," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '07)*, pp. 2573–2578, June 2007.

[9] The CarTel project, http://cartel.csail.mit.edu/.

[10] NOW: Network on Wheels, http://www.network-on-wheels.de.

[11] "UDel Models for Simulation of Urban Mobile Wireless Networks," http://udelmodels.eecis.udel.edu/.

[12] J. Harri, F. Filali, and C. Bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 19–41, 2009.

[13] R. Mangharam, D. S. Weller, D. D. Stancil, R. Rajkumar, and J. S. Parikh, "GrooveSim: a topography-accurate simulator for geographic routing in vehicular networks," in *Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET '05)*, pp. 59–68, September 2005.

[14] M. Piorkowski, M. Raya, A. Lugo, P. Papadimitratos, M. Grossglauser, and J. P. Hubaux, "TraNS: realistic joint traffic and network simulator for VANETs," in *Proceedings of the ACM SIGMOBILE Mobile Computing and Communications Review (MC2R '07)*, 2007.

[15] K. Ibrahim and M. C. Weigle, "ASH: application-aware SWANS with highway mobility," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 1–6, April 2008.

[16] B. Liu, B. Khorashadi, H. Du, D. Ghosal, C. N. Chuah, and M. Zhang, "VGSim: an integrated networking and microscopic vehicular mobility simulation platform," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 134–141, 2009.

[17] V. Kumar, L. Lin, D. Krajzewicz et al., "ITETRIS: adaptation of ITS technologies for large scale integrated simulation," in *Proceedings of the 71st IEEE Vehicular Technology Conference (VTC '10)*, May 2010.

[18] S. Wang and C. Lin, "NCTUns 6. 0: a simulator for advanced wireless vehicular network research," in *Proceedings of the IEEE Vehicular Technology Conference (VTC '10)*, 2010.

[19] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.

[20] Network Simulator ns-3, http://www.nsnam.org/.

[21] Network Simulator ns-2, http://isi.edu/nsnam/ns/.

[22] OMNet++, http://www.omnetpp.org/.

[23] R. Barr, Z. J. Haas, and R. van Renesse, "JiST: embedding simulation time into a virtual machine," in *Proceedings of the EuroSim Congress on Modeling and Simulation*, 2004.

[24] SUMO, http://sumo.sourceforge.net.

[25] CanuMobiSim, http://canu.informatik.uni-stuttgart.de.

[26] 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, Los Angeles, Calif, USA, September 2006.

[27] M. Zhang and R. S. Wolff, "A border node based routing protocol for partially connected vehicular ad hoc networks," *Journal of Communications*, vol. 5, no. 2, pp. 130–143, 2010.

[28] A. Keranen and J. Ott, "Increasing reality for DTN protocol simulations," Tech. Rep., Networking Laboratory, Helsinki University of Technology, July 2007.

[29] A. Keranen, J. Ott, and T. Karkkainen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTOOL '09)*, March 2009.

[30] Z. Xiong, W. Luo, L. Chen, and L. M. Ni, "Data vitalization: a new paradigm for large-scale dataset analysis," in *Proceedings of the 16th IEEE International Conference on Parallel and Distributed Systems (ICPADS '10)*, pp. 251–258, December 2010.

[31] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000.

[32] A. Kesting, M. Treiber, and D. Helbing, "MOBIL: general lane changing model for car-following models," in *Proceedings of the Transportation Research Board Annual Meeting*, January 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 417830, 8 pages
doi:10.1155/2012/417830

Research Article

Efficient Sensor Localization Method with Classifying Environmental Sensor Data

Ae-cheoun Eun and Young-guk Ha

Department of Computer Science and Engineering, Konkuk University, Seoul 143-701, Republic of Korea

Correspondence should be addressed to Young-guk Ha, ygha@konkuk.ac.kr

Received 31 July 2012; Revised 15 October 2012; Accepted 30 October 2012

Academic Editor: Shan Lin

Sensor location estimation is important for many location-based systems in ubiquitous environments. Sensor location is usually determined using a global positioning system. For indoor localization, methods that use the received signal strength (RSS) of wireless sensors are used instead of a global positioning system because of the lack of availability of a global positioning system for indoor environments. However, there is a problem in determining sensor locations from the RSS: radio signal interference occurs because of the presence of indoor obstacles. To avoid this problem, we propose a novel localization method that uses environmental data recorded at each sensor location and a data classification technique to identify the location of sensor nodes. In this study, we used a wireless sensor node to collect data on various environmental parameters—temperature, humidity, sound, and light. We then extracted some features from the collected data and trained the location data classifier to identify the location of the wireless sensor node.

1. Introduction

Location-aware services are an important application of ubiquitous computing. Therefore, in wireless sensor networks (WSNs), localization has become an essential functionality. Essentially, the localization of a wireless sensor node is achieved by measuring the received signal strength (RSS) of wireless links between the target node and multiple reference nodes and using the theory that the signal strength of the wireless link between two wireless nodes decreases as the distance between them increases. Measured RSS data are used to determine the location of the target node in methods such as triangulation [1], a centroid method [2], or fingerprinting [3, 4]. However, such a method has some limitations when used in indoor environments owing to the reflection, loss, and distortion of signals because of the presence of indoor obstacles. In addition, the RSS between two sensor nodes for a given distance decreases with the battery capacity of the sensor nodes.

In this paper, we propose a novel localization method for sensor nodes in indoor wireless sensor network environments [5]. The method involves the classification of environmental data, such as temperature, humidity, sound, and light, collected by the target nodes. To classify these environmental data according to the locations where they were recorded, we use a k-nearest neighbor (k-NN) classifier. In addition, we use a feature extraction method for the recognition through principal component analysis (PCA). We then perform localization experiments in an actual test environment to validate the proposed method.

The rest of this paper is organized as follows. In Section 2, the existing sensor localization methods and some problems that arise when using these methods in real-world applications are analyzed. In Section 3, we describe the design of the localization method proposed in this paper. In Section 4, the implementation of the method is explained and experimental results are discussed. Finally, in Section 5, the paper is summarized and future directions are given.

2. Related Work

2.1. Well-Known Localization Methods. Triangulation techniques include RSS indicator (RSSI) [6], time of arrival (ToA), time difference of arrival (TDoA), and angle of

arrival (AoA). RSSI measures the attenuation of the radio signal strength between a sender and a receiver. The power of the radio signal decreases exponentially with increasing distance, and the receiver can measure this attenuation and use it to estimate the distance from the sender. ToA [6–8] is based on the speed of radio wave propagation and the time that a radio signal takes to move between two objects. Combining these pieces of information allows a ToA system to estimate the distance between a sender and a receiver. TDoA [6, 9] measures the difference between arrival times. Beacon nodes are necessary to transmit both ultrasound and radio frequency (RF) signals simultaneously. A sensor measures the difference between the arrival times of the two signals and relays the range to the beacon node. Unlike the above techniques, which measure distance, AoA [10] techniques measure the angle at which a signal arrives. Angles can be combined with the estimated distance or other angle measurements to derive positions. AoA is an attractive method because of the simplicity of the subsequent calculations.

The use of triangulation methods for indoor environments is very problematic because they use the RSS; the drawback [11] of using the RSS has been described in Introduction. Thus, to avoid these problems, other methods should be used.

2.2. RF Fingerprinting. A fingerprinting [3, 4, 12] algorithm is usually the basis of a WLAN localization system. The proposed technique, based on the discriminant-adaptive neural network (DANN) [3] architecture, is implemented in a real-world WLAN environment, and realistic measurements of the signal strength are collected. This technique is used to extract useful information from available access points (APs) and transmit the information to the discriminative components (DCs). These components use this information for discriminating between different locations and rank it according to its quantity. Rank the locations according to the respective access point. The technique incrementally inserts DCs and recursively updates their weightings in the network until no further improvement is required. The network can accomplish learning intelligently using the information provided by the inserted DCs. Moreover, the weights of the input layer and the inserted components are determined using multiple discriminant analysis (MDA) [13] in order to maximize the useful information contained in the network. The RF fingerprinting technique also uses RSS values to determine the position of a sensor node. Thus, the problem explained in Section 2.1 is faced.

2.3. eWatch System. eWatch [14] is a wearable sensing, notifying, and computing platform that resembles a wristwatch, a factor that renders it very accessible, instantly viewable, ideally located for sensors, and unobtrusive to its users. Information transfer from eWatch to a cellular phone or stationary computer occurs through wireless bluetooth communication.

eWatch senses light, motion, sound, and temperature and provides visual, sound, and tactile notification. It has ample processing capabilities and a multiday battery life, which allows realistic user studies. This paper describes the motivation for developing a wearable computing platform, a description of power-aware hardware and software architectures and demonstrates the identification and recognition of a set of frequently visited locations via online nearest-neighbor classification.

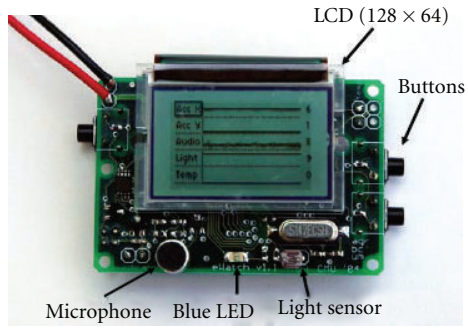


Figure 1: Top view of the eWatch board.

Figure 1 shows the board that was used for data collection and analysis in the eWatch project. eWatch finds a location using three environmental parameters: sound, temperature, and light. Note that the use of more parameters would increase the localization accuracy. In this paper, we discuss methods for measuring a user's location by using four parameters: sound, temperature, light, and humidity. In the present study, these sensing data were used in location-aware technology.

3. Design of the Proposed Method

In this section, we explain the design of the proposed system and describe the architecture and design concepts. In addition, details of the method for each module will be discussed.

3.1. System Architecture. Figure 2 shows the overall system architecture and data flow. The location data collection module (LDCM) periodically collects environmental data of each space and provides the data to the system. The environmental data of each space consists of temperature, humidity, light, and sound data.

The collected environmental data of each space is used for training the user location recognition module (ULRM). The location data feature extraction module (LFEM) provides a feature extraction function. This function is applied to the environmental data of the user location provided by the LDCM. The extracted features are input into the ULRM for the purpose of user location recognition. Primarily, feature extraction is used to decrease the amount of high-frequency data. In the LFEM, the data are converted from the format of the ULRM training module to the attribute-relation file format (ARFF) used by Weka [15]. Weka is a data mining tool. In addition, the LDCM module can sense the current environmental data communicated in the location test. Finally, the sensed and trained data will be used as test data to recognize a user's location.

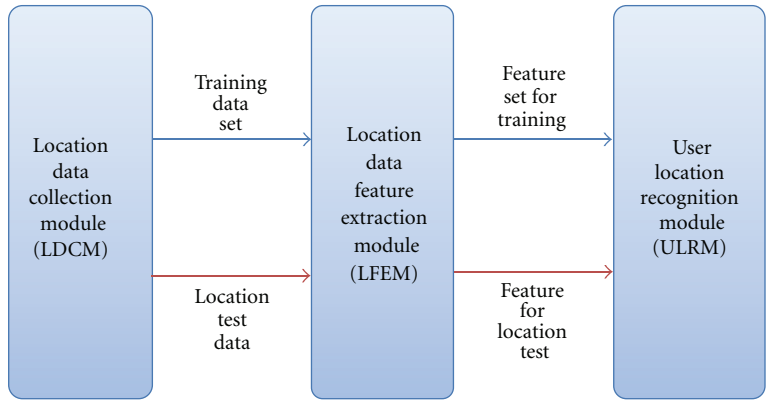


Figure 2: System architecture.

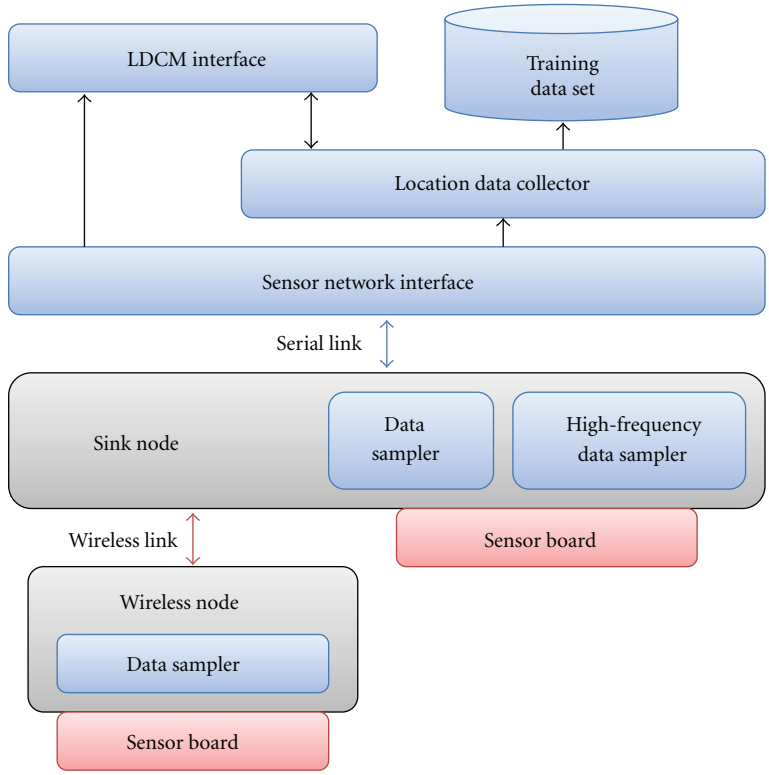


Figure 3: LDCM.

In addition, the LFEM uses a different extraction method for each feature. It uses PCA for feature extraction. In PCA, the number of principal components is less than or equal to the number of original variables. The ULRM uses a set of trained data for recognizing location. In this section, we discuss the data format for data training and that of the collected data. In addition, the ULRM shows the location recognition results based on real-time data extracted from the LFEM module.

3.2. LDCM. This section describes the elements of the LDCM. Figure 3 shows the structure of the LDCM. This module periodically senses and collects the environmental data of each space and provides it to the system. These data are then used for recognizing the user location. The WSN [16] consists of a wireless sensor node and sink nodes. A Hmote2420 sensor, which can sense temperature, humidity, light, and sound, is used in the sensor board.

The wireless sensor node loads data from the data sampler program and sensor board. Thus, the sensor nodes can acquire environmental data from the sensor board. While the data (temperature, humidity, light, and sound data) are being sent, the WSN can also send the data to the sink node through a wireless link by using a sampler program. The wireless link operates in the half-duplex transmission mode. The sink node delivers sensor data to the base station and the sensor network interface through a serial link. The sink node can also acquire environmental data directly from

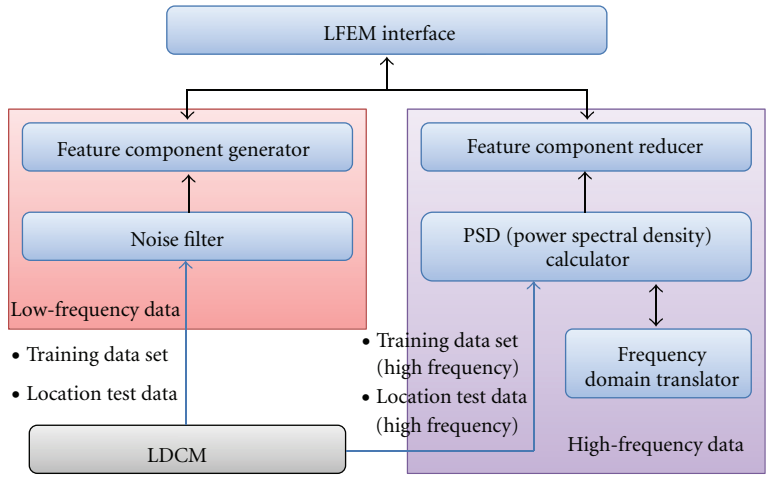


Figure 4: LFEM.

the installed data sampler and sensor board, but not through the wireless sensor node. The sink node has a high-frequency data sampler for sampling high-frequency data effectively. Two types of samplers, a high-frequency sampler and a low-frequency sampler, are used because of the very large amount of processing required for high-frequency data.

The sensor network interface links the sensor network to a base station. The hardware interface, such as USB or RS-232, uses a common serial link. On the other hand, the software interface has a device driver and a system application programming interface (API) for processing data received from the serial link. The location data collector saves environmental data in the data file of the training set.

This training set is created after the data file is given as the input to the LFEM, and it is used by the LFEM for training the ULRM with the feature extraction process. The LDCM interface provides an API, which can be used to obtain environmental data at the user's location. In the next section, the data extraction method will be explained.

3.3. LFEM. In our system, the LFEM performs data extraction. The structure of the module is shown in Figure 4. The extraction method used in the LFEM depends on the type of environmental data used. We perform noise filtering for low-frequency data and determine the power spectral density (PSD) for high-frequency data. Therefore, the collection of low-frequency data, such as temperature and humidity, involves noise filtering. Noise filtering helps distinguish between usable data and unusable data. Thus, our module acquires only usable data. However, high-frequency data, such as sound and light, are not subjected to noise filtering.

For collecting high-frequency data, the PSD should be used. Sound data and the top five principal component data are then extracted through frequency domain conversion. These real-time data are provided as input to the LFEM interface. They are used for feature extraction in the ULRM during user localization. The LFEM then creates a feature component on the basis of these data.

3.4. ULRM. Figure 5 shows the ULRM. The module is based on the space recognition features generated by the LFEM for training. This module also provides a user interface with an application level. The location data classifier classifies the current user's location features. To perform this task, the location data classifier is trained on a set of environmental data. The ULRM input is processed using the user location recognizer classification based on the received environmental data to provide an output. The ULRM performs a location test and training using the location data classifier.

In the first recognition test, the feature data can be sent to the location data classifier through the user location recognizer. The recognizer uses k-NN as the location data classifier. The k-NN classification was developed in view of the need for performing discriminant analysis when reliable parametric estimates of probability densities are not available. This classifier is traditionally based on the Euclidean distance between a test sample and specified training samples. k-NN is an algorithm for measuring the distance between bound objects from the value of K, which is the Euclidean distance. Finally, the result is returned to the user location recognizer through the ULRM interface and is displayed on the recognizer. ULRMs transfer training data from the user location trainer to the location data classifier. Finally, the data are displayed on the ULRM interface.

3.5. Location Feature Extraction and Recognition Procedure. Figure 6 shows location feature extraction and recognition procedure. The LDCM can sense environmental data and transfer them to the base station. The base station has the LFEM and the ULRM. The upper part of Figure 6 shows a method for feature extraction, which is the function of the LFEM (see Section 3.3).

The LFEM can extract features. For example, assume that we apply PCA to the collected sound and light data. The data are then analyzed using PSD. In spectrum analysis, PSD of data whose analysis element is limitless is used. Fourier transform is used to express limitless data as power per hertz. This representation is often simply called the

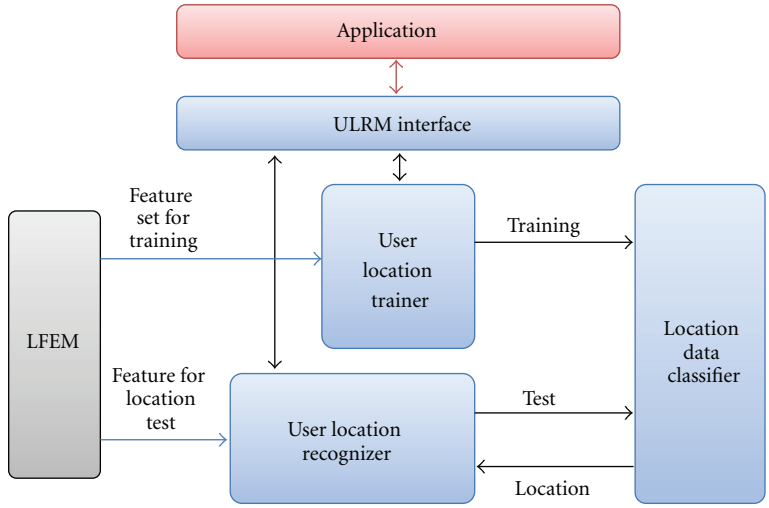
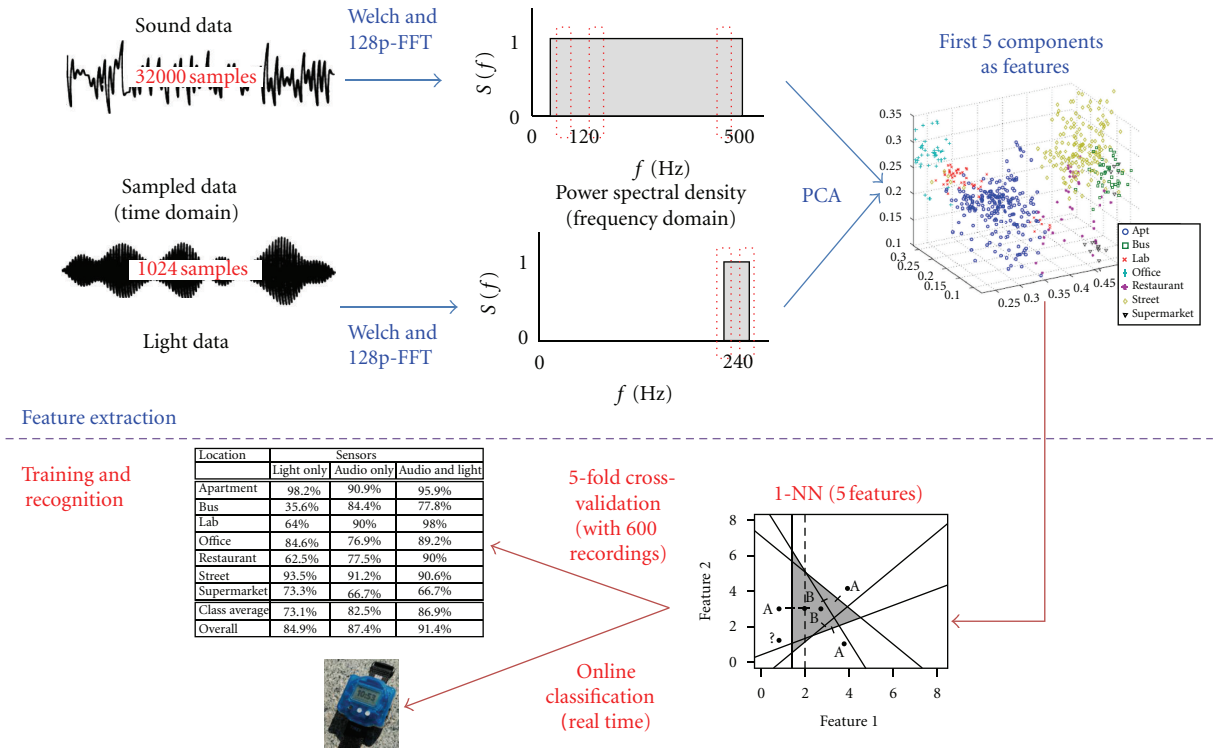


Figure 5: ULRM.



Location	Sensors		
	Light only	Audio only	Audio and light
Apartment	98.2%	90.9%	95.9%
Bus	35.6%	84.4%	77.8%
Lab	64%	90%	98%
Office	84.6%	76.9%	89.2%
Restaurant	62.5%	77.5%	90%
Street	93.5%	91.2%	90.6%
Supermarket	73.3%	66.7%	66.7%
Class average	73.1%	82.5%	86.9%
Overall	84.9%	87.4%	91.4%

Figure 6: Location feature extraction and recognition procedure.

power spectrum of the data. Intuitively, the spectral density measures the frequency content of a stochastic process and helps identify periodicities. Thus, different extraction methods are applied to different types of data. In addition, PCA is applied to data for high-speed analysis. PCA is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component has the highest variance possible under the constraint that it is orthogonal to the preceding components. The principal components are guaranteed to be independent only if the dataset is jointly and normally distributed. PCA is sensitive to the relative scaling of the original variables. We perform PCA on and partial characteristics from the sound and light data.

The lower part of Figure 6 shows the method used for location recognition, which is the function of the ULRM. The ULRM either recognizes a user location or trains

```
@relation usn

@attribute temp numeric
@attribute hum numeric
@attribute light numeric
@attribute sound numeric
@attribute class {lobby, lab, toilet, cafeteria, bank, bookstore}

@data
#temp value, #hum value, #light value, #sound value, #class name
```

Figure 7: ARFF format of dataset file.

Table 1: Implementation environments.

Operating system	(i) Location recognition system: Windows Vista (ii) Wireless sensor node: TinyOS
Programming language	(i) Location recognition system: Java (ii) Wireless sensor node: nesC
Software tools	(i) Location data feature extraction: MATLAB (ii) Location data classification: Weka
Hardware	(i) Location recognition system: Intel 2.0 Hz PC (ii) Wireless sensor node: Hmote2420

Figure 8: Environments considered in the experiments.

user location data. The training element uses K-fold cross-validation and k-NN methods.

4. Implementation and Experiments

4.1. Implementation Environments. Various software and hardware tools are used in our system. Table 1 shows the implementation environments. The operating system used for the location recognition system, which is coded in Java, is Microsoft Windows Vista. The wireless sensor is developed using TinyOS. We created a wireless sensor node using Hmote2420 and nesC. We used nesC in the TinyOS environment in order to use the Hmote2420 wireless network system. The operating systems and programming tools are described in the software section, while the hardware specifications of the sensor and the computer are presented in the hardware section.

The Hmote2420 sensor and TinyOS were used in the LCDM. Hmote2420 was used to collect environmental data and information at the base station. TinyOS was used to deliver the collected data into base station. In addition to the LFEM, we used a computer, a sensor node, a Java platform, and MATLAB to extract features from the collected data. The ULRM used the Java platform to show the recognized user's position, which was determined from the collected features. In addition, the k-NN algorithm was used for location recognition.

Table 2 shows the information related to sampling of environmental data. These sampled data were extracted using MATLAB, which was also used to convert the data to the ARFF format used by Weka.

4.2. Environmental Dataset Generation. The format of environmental datasets used in this study was ARFF. Temperature, humidity, light, and sound data were used to build training datasets, as explained in Section 4.2. The reason why we have used light, sound, temperature, and humidity is that they are the main physical parameters that characterize a place.

Feature extraction from a dataset involves different processes, depending on the sampling rate of the dataset (see Figure 4). High-frequency data, such as light and sound data, may lead to the training and classification process being slow, because the size of the dataset is too large. Therefore, to reduce the number of feature components, PCA was used to extract the most representative feature components for each location. Before the feature extraction procedure, high-frequency environmental datasets are transformed into the frequency domain using FFT.

On the other hand, environmental data sampled at a low frequency, such as temperature and humidity data, can be directly used as representative features for each location. Therefore, PCA need not be performed on these datasets. Figure 7 shows the format of ARFF training dataset files.

4.3. Experimental Method. In our experiments, data were collected from different places in Konkuk University (Figure 8): a laboratory, a toilet, the lobby of the New Millennium Hall, a bank, a bookstore, and a cafeteria (the last three are located in the student union building). The experiments are explained below.

First, we collected 100 datasets from each place by using the sensor. A total of 600 datasets were collected from

Table 2: Environmental data collection methods.

	Rec/sec	Sampling rate (Hz)	Duration (sec)	samples/rec	Type of sampler
Temperature	5/10	1	4	4	Std.
Humidity	5/10	1	4	4	Std.
Light	5/10	2048	0.5	1024	High Freq.
Sound	5/10	8000	4	32000	High Freq.

Table 3: Offline localization experimental results.

Test data	Classified					
	Lobby	Laboratory	Toilet	Cafeteria	Bank	Bookstore
Lobby	**91**	0	1	0	8	0
Laboratory	0	**99**	0	0	0	1
Toilet	1	0	**94**	0	5	0
Cafeteria	0	0	0	**99**	1	0
Bank	4	0	2	2	**92**	0
Bookstore	0	3	0	0	0	**97**

the six locations. Second, the collected data were classified into high- and low-frequency data. The classified data were extracted using the feature extraction method of MATLAB. The extracted data were then converted into formats compatible with Weka. Next, ten more datasets were collected at the same time and at the same locations. Finally, our system used the collected data to recognize user locations.

4.4. Results and Discussion. After training the localization classifier, we collected 10 additional feature datasets from different places at each location to test the classifier. The sensor's location was then identified using the 10 datasets.

The average localization accuracy (A_{ave}) was calculated with formula (1), where T_l denotes the set of all the datasets collected at location l, TC_l is a correctly classified dataset for location l ($TC_l \subset T_l$), and L is the number of locations considered in the localization experiments:

$$A_{\text{ave}} = \frac{\sum_{1 \le l \le L} |TC_l| / |T_l|}{L}. \quad (1)$$

Table 3 shows the confusion matrix for the test results. The 3-NN classification method with 20-fold cross-validation was used in the experiments. As shown in the matrix, the average localization accuracy was about 95.3%. This table shows that the highest levels of recognition were achieved for the laboratory and cafeteria.

In the table, the correct location data are shown in bold font. High localization accuracy is achieved for the laboratory and cafeteria data because of the correct classification of features. This implies that a high localization accuracy will be obtained in places where the features are well separated. Errors in recognition occasionally occur in the case of the lobby and bank. This implies that these two environments are similar in temperature, humidity, light, and sound.

Table 4 shows the real-time localization accuracy. In an experiment, the average localization accuracy of real-time location recognition was 82.2%. The highest localization accuracy was achieved for the toilet environment. On the other hand, the bookstore showed the lowest localization accuracy because the indoor light data for it are similar to those for the lobby.

Table 4: Real-time localization experimental results.

Location	localization accuracy
Laboratory	76.7%
Lobby	83.3%
Toilet	100%
Cafeteria	86.7%
Bank	93.3%
Bookstore	53.3%
Average	82.2%

The classifier confused the bookstore with the lobby. This occurred because both the locations have similar light and temperature conditions. However, in the case of the toilet, because of the high humidity, the recognition results showed high localization accuracy. Finally, we can improve the localization performance of our system further by using additional types of environmental data, especially for environments with similar conditions with regard to temperature, humidity, light, and sound.

5. Conclusion

In this paper, we have proposed a novel location recognition method for wireless sensor nodes. The method involves the classification of environmental data features using the k-NN localization data classifier. We performed localization experiments in an actual test environment by using the proposed method. The experimental results indicated high localization accuracy. In a real-time recognition experiment, the localization accuracy was found to be 82.2%. This value indicates that environmental data can be used for the purpose of location recognition. It also shows the importance of environmental data recognition in location recognition. Our future research will focus on combining the proposed

location recognition method and other localization methods, such as RSS pattern recognition methods. Furthermore, we intend using a modified version of PCA [17] and k-NN for location feature extraction and in the classification procedures of the proposed method to improve the overall localization performance.

Acknowledgment

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), which is funded by the Ministry of Education, Science and Technology (Grant no. 2012006817).

References

[1] Z. Xia and C. Chen, "A localization scheme with mobile beacon for wireless sensor networks," in *Proceedings of the 6th International Conference on ITS Telecommunications (ITST '06)*, pp. 1017–1020, June 2006.

[2] C. H. Lim, Y. Wan, B. P. Ng, and C. M. S. See, "A real-time indoor WiFi localization system utilizing smart antennas," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 618–622, 2007.

[3] S. H. Fang and T. N. Lin, "Indoor location system based on discriminant-adaptive neural network in IEEE 802.11 environments," *IEEE Transactions on Neural Networks*, vol. 19, no. 11, pp. 1973–1978, 2008.

[4] J. Yim, "Comparison between RSSI-based and TOF-based indoor positioning methods," *International Journal of Multimedia and Ubiquitous Engineering*, vol. 7, no. 2, 2012.

[5] Y.-g. Ha, "Dynamic integration of zigbee home networks into home gateways using OSGI service registry," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 470–476, 2009.

[6] C. C. Chen, D. C. Wang, and Y. M. Huang, "A novel method for unstable-signal sensor localization in smart home environments," *International Journal of Smart Home*, vol. 2, no. 3, 2008.

[7] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, pp. 775–784, March 2000.

[8] A. Savvides, C. C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 166–179, July 2001.

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

[10] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (IEEE INFOCOM '03)*, pp. 1734–1743, April 2003.

[11] Y.-g. Ha, H. Kim, and Y. Byun, "Energy-efficient fire monitoring over cluster-based wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 460754, 11 pages, 2012.

[12] R. Duda, P. Hart, and D. Strok, *Pattern Classification*, John Wiley & Sons, New York, NY, USA, 2001.

[13] A. K. Chandra-Sekaran, P. Dheenathayalan, P. Weisser, C. Kunze, and W. Stork, "Empirical analysis and ranging using environment and mobility adaptive RSSI filter for patient localization during disaster management," in *Proceedings of the 5th International Conference on Networking and Services (ICNS '09)*, pp. 276–281, April 2009.

[14] U. Maurer, A. Rowe, A. Smailagic, and D. P. Siewiorek, "eWatch: a wearable sensor and notification platform," in *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN '06)*, pp. 142–145, April 2006.

[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, 2009.

[16] Y.-T. Kim, Y.-S. Jeong, and G.-C. Park, "Design of RSSI signal based transmit-receiving device for preventing from wasting electric power of transmit in sensor network," in *Proceedings of the 2nd International Conference on Ubiquitous Computing and Multimedia Applications (UCMA '11)*, vol. 151 of *Communications in Computer and Information Science*, pp. 331–337, 2011.

[17] S. H. Fang and C. H. Wang, "A dynamic hybrid projection approach for improved Wi-Fi location fingerprinting," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 1037–1044, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 715261, 12 pages
doi:10.1155/2012/715261

Research Article

A Path Planning Algorithm with a Guaranteed Distance Cost in Wireless Sensor Networks

Yuanchao Liu,[1] Shukui Zhang,[1,2] Jianxi Fan,[1] and Juncheng Jia[1]

[1] *Institute of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

Correspondence should be addressed to Shukui Zhang, zhangsk2000@163.com

Received 11 July 2012; Accepted 28 August 2012

Academic Editor: Yong Sun

Navigation with wireless sensor networks (WSNs) is the key to provide an effective path for the mobile node. Without any location information, the path planning algorithm generates a big challenge. Many algorithms provided efficient paths based on tracking sensor nodes which forms a competitive method. However, most previous works have overlooked the distance cost of the path. In this paper, the problem is how to obtain a path with minimum distance cost and effectively organize the network to ensure the availability of this path. We first present a distributed algorithm to construct a path planning infrastructure by uniting the neighbors' information of each sensor node into an improved connected dominating set. Then, a path planning algorithm is proposed which could produce a path with its length at most c times the shortest Euclidean length from initial position to destination. We prove that the distributed algorithm has low time and message complexity and c is no more than a constant. Under different deployed environments, extensive simulations evaluate the effectiveness of our work. The results show that factor c is within the upper bound proved in this paper and our distributed algorithm achieves a smaller infrastructure size.

1. Introduction

Recently, as a large number of sensor nodes are deployed to monitor the environment and detect critical events [1–4], navigation has received wide attention in applications of WSNs. Usually, a mobile node is equipped with a device that can communicate with sensor nodes. After a WSN has been deployed in the monitoring area, relevant sensor nodes will send in situ data to the control center when they detect dangerous events happening in the area. Then, a part of sensor nodes would guide several mobile nodes which equip specific instruments to the destination and let them deal with the emergency event, such as navigating fire-fighting equipments automatically to exact areas to extinguish fire. Hence, how to design an effective path for the mobile node is a fundamental problem. The so-called navigation refers to the art of getting from one place to another in an efficient manner. Generally speaking, it could be described by three questions: "Where am I?," "Where am I going?" and "How should I get there?" [5], which need the localization methods, path planning algorithms and the moving control technology, respectively. In WSNs, the navigation of mobile node needs to communicate with sensor nodes to get the target data and correct its direction. While sensor nodes have finite energy and limited communication range and construct the network topology by self-organization, an efficient data routing infrastructure is necessary for updating rescue instructions periodically so as to guide the mobile node to its destination, for example, virtual backbone [6] and collection tree [7].

Up to now, a part of proposed navigation algorithms in WSNs rely on GPS and other modules to obtain locations of mobile nodes in real time [8, 9], which require a high hardware cost and energy consumption. In some particular environments such as mines, underwater environment, and underground tunnels, location information may not be able to achieve, these scenarios would limit the application of existing navigation algorithms using localization technique. To solve these drawbacks for emergency escape, some researchers [10] proposed artificial potential fields in which sensor nodes act as signposts for the mobile node to follow. Lately, some novel navigation algorithms for emergency

rescue in WSNs have been presented [11, 12]. As most of the above algorithms focus on providing safe paths in dangerous environment, they have overlooked the distance cost of paths. In [13], an idea of navigation overhead is denoted as a ratio between the Euclidean length of moving path and the shortest Euclidean length from initial position to destination which indicates the distance cost of algorithm.

In this paper, we characterize the navigation problem as a path planning problem. Firstly, based on the research of connected dominating set, we propose an improved distributed algorithm to construct a preliminary infrastructure for data routing. Then we construct a path planning infrastructure by combining the built infrastructure with neighbors' information of each node. At last, we introduce a path planning algorithm by tracking sensor nodes in the network based on our path planning infrastructure. We show that our infrastructure not only can serve as a backbone to send in suit data, but also can update and modify the path planning algorithm to ensure its availability. We also prove that the path planning algorithm provides a path which guarantees a constant distance cost compared with the shortest one from initial position to destination in the Euclidean plane.

The remainder of this paper is organized as follows. Section 2 describes the related work. Section 3 introduces some useful definitions and constructs the path planning infrastructure in the network. Section 4 proposes an effective path planning algorithm and analyses the performances of the proposed algorithms. Section 5 shows simulation results. Section 6 concludes this paper.

2. Related Work

A number of solutions have been proposed to solve the navigation problem in WSNs. In [8], an intelligent control architecture for mobile node has been proposed with environment sensing model. The architecture used clustering strategy by applying shared memory in the network and created control with a set of ultrasonic GPS modules. By using a triangular method, the algorithm provided the global position of mobile node in the environment. Therefore, the algorithm could guide the mobile node moving to the target effectively with a high precision, but it had a high cost and was difficult to implement in the environment which cannot obtain the location information of sensor nodes. Without using triangular localization technique, a navigation strategy based on planning reliable visual landmarks has been proposed in [9]. The method modeled landmarks within a directed graph and used the Markov decision process to compute the navigation path. The disadvantage of this strategy was that it needed to provide geographic information of surrounding environment beforehand and equip the mobile node with expensive detectors. To localize the mobile node quickly, it also needed to plan extra artificial visual landmarks in the environment. In [14], a protocol utilized the sensor network infrastructure for navigation has been proposed. Without the location information of network, the protocol constructed a road map system to provide navigating routes of the mobile node which consist of a sequence of sensor nodes to avoid the dangerous area. The mobile node tracked the target sensor node by measuring the strength and direction of wireless signals. When the dangerous areas have changed, the algorithm updated the navigating routes to ensure the safety of the mobile node. Without using any localization mechanism or requiring location information, the study in [10] also presented a distributed algorithm for dangerous area avoidance. During the motion of the mobile node, the algorithm combined the artificial potential field with the destination information to navigate the mobile node in real time. The dangerous area seemed to generate a repulsive potential which would push the mobile node away while the destination generated an attractive potential which would pull the mobile node towards the destination. Each sensor node calculated its potential value and tried to find a navigation path of the least total potential value to make mobile node bypass dangerous area. But the algorithm was prone to produce a local pole which would make the mobile node unable to reach the target. In [15], the algorithm set each sensor node with a weight based on the hop distance to the nearest safe region. Sensors were assigned smaller weight if they were closer to the safe exit. Otherwise, sensors were assigned greater weight. The mobile node chose the sensor node with the smallest weight in its communication range as its direction of movement to avoid the dangerous area. In [11], a novel distributed navigation algorithm has been proposed for individuals to escape from critical event region in WSNs. With no goal or exit as guidance, the navigation algorithm computed the convex hull of the event region by topological methods to make individuals get out of the event region. Because congestion may be caused by the individuals rushing for the safe exits, the study in [12] proposed an efficient navigation strategy by taking both pedestrian congestion and rescue force flexibility into account. The individuals navigation is treated as a network flows problem in the graph which is modeled by the emergency regions. In [13], a navigation algorithm using the metric calculated from neighbor's hop count has been proposed in WSNs. This algorithm did not require predefined maps or GPS modules. By interacting with neighboring sensor nodes, the mobile node moved towards the target where the hop count becomes smaller and finally reached the destination by periodically measuring the value. But the mobile node has not considered selecting a proper sensor node from its neighbors as a local target which would decrease the deviation between current direction and optimal moving direction and there was no theoretic analysis for the distance cost. In [16], a novel method which relied on the heat diffusion equation has been proposed to finish the navigation process conveniently. The method guided the mobile node by establishing a high density of the information field.

In summary, although using a localization technique had more precision, but the algorithms without requiring locations could apply into more scenarios. And most of them modeled a WSN as a graph and let a planning path in the environment correspond to a directed vertex path in the graph. While all of the above algorithms adopted existing protocols for data routing, they have overlooked that

an efficient routing infrastructure may not only send in suit data quickly, but also update and modify the path planning algorithm to achieve a guaranteed distance cost.

3. Network Model

In WSNs, we assume that sensor nodes are randomly deployed in the Euclidean plane. Each sensor node u is assigned a global unique identifier which denoted as id_u. For simplicity, let all sensor nodes have the same communication and sensing ranges which are referred to R_C and r_s, respectively. The maximum communication range $R_{\max}$ can be obtained by adjusting the transmitting power. We use an unweighted graph $G = (V, E)$ to model the WSN. The vertex set V represents sensor nodes and the edge set E represents communication links if any two vertices u and v satisfy $d(u, v) \leq R_{\max}$, where $d(u, v)$ is the Euclidean length between u and v. Let n denote the number of vertices in V. Without any confusion, we assume that the terminologies of vertex and node are interchangeable. Furthermore, we can use a UDG to abstract the original sensor network by scaling each edge length with $R_{\max}$. That is, for any two vertices u and v in UDG, an edge exists between u and v if the distance $d(u, v) \leq 1$. In order to make a WSN monitor the whole area entirely, we also assume that there are sensor nodes as many as possible which would build a quite dense network.

In order to construct an efficient path planning infrastructure for routing data and providing an available path, we introduce some useful definitions and properties.

Definition 1. Given a graph G and a subset $V_C \subseteq V$, for any vertex $v \in V - V_C$, if there is at least one adjacent vertex in V_C, then V_C is referred to a dominating set (DS). If the vertex induced graph $G[V_C]$ is connected, then V_C is a connected dominating set (CDS).

A CDS has been recommended to serve as a virtual backbone for WSNs to dramatically reduce routing overhead. In this paper, we focus on a special CDS proposed by Du et al. in [19]. Because not only the CDS can provide a guaranteed routing overhead for any pair of nodes which will be shown in Lemma 3, but also we can implement it to build an effective path planning infrastructure by uniting neighbors' information of each sensor node into CDS.

Definition 2. Given a graph G and a subgraph $C \subseteq G$, for two distinct vertices u and v in $V(G)$, let $h(u,v)$ and $h_C(u,v)$ denote the hop number of the shortest vertex path between this vertex pair through G and C, respectively.

Lemma 3 (see [19]). *Let G be a connected graph and C a dominating set of G. Then, for a constant $\beta \geq 5$ and any pair of distinct vertices u and v, $h_C(u, v) - 1 \leq \beta \cdot (h(u, v) - 1)$ if and only if for any pair of distinct vertices u and v with $h(u, v) = 2, h_C(u, v) - 1 \leq \beta$.*

Clearly, for any two adjacent vertices u and v in UDG, there is $d(u, v) \leq h(u, v)$ for $h(u, v) = 1$. Furthermore, by Lemma 3, if for any pair of vertices u and v with $h(u, v) = 2$, $h_C(u, v) \leq \beta + 1$. Then, for any pair of distinct vertices u and v, we have $h_C(u, v) \leq \beta \cdot h(u, v)$, where $\beta \geq 5$ [19].

Although in [20], a better performance of β was proposed, but it did not give any sufficient and necessary condition. And it needed a centralized computation through the sequence of a shortest vertex path between two corresponding distinct vertices in the network. Here, we improve the limitation of $h(u, v) = 2$ to obtain a simpler sufficient and necessary condition which could be implemented just with the help of 1-hop neighbors for each node.

Lemma 4. *Let G be a connected graph. For a constant $\beta \geq 5$ and any pair of distinct vertices u and v, $h_C(u, v) \leq \beta \cdot h(u, v)$ if and only if for any pair of distinct vertices u and v with $h(u, v) = 1, h_C(u, v) \leq \beta$.*

Proof. It is trivial to show the "only if" part. Next, we show the "if" part. Consider a pair of distinct vertices u and v. Let the shortest vertex path from u to v in G be $u_0u_1u_2 \cdots u_k$, where $u_0 = u$ and $u_k = v$. By the condition, we have $h_C(u_i, u_{i+1}) \leq \beta$ for $0 \leq i \leq k - 1$. Then, it implies that u and v are connected by a path in C with at most $\beta \cdot k$ hops. Hence, we obtain $h_C(u, v) \leq \beta \cdot h(u, v)$. □

By Lemma 4, we can distributedly construct the backbone with guaranteed routing overhead which is a foundation of our path planning infrastructure. Compared with the algorithm in [19] which contained two BFSes to connect any pair of vertices u and v in the DS with $h(u, v) \leq 4$, we construct a DS in the first step. Then we connect u and v in DS with hop distance $h(u, v) = 2$ and $h(u, v) = 3$ in the second and third step, respectively. From appearance, our algorithm is similar to that in [20]. However, the procedures of algorithm are much different, which have optimized rules of choosing connectors in each step. The detailed algorithm is shown in Algorithm 1.

Procedure 1. Coloring2(G, C)

Input: A connected graph G and a black node set C.
Output: A node subset V_{2C} with coloring grey.

(1) Each white node x with $k_x \geq 2$ sends packet (id_x, CL_x, B_x) to its neighbors, where id_x, CL_x and B_x are the id, color and black neighbor set of x, respectively.

(2) After black node u has received packets (id_x, CL_x, B_x)s from its white neighbors, u saves the black nodes in B_x of each packet into its 2-hop black neighbor set $Nb_1(u)$. And u constructs a 2 dimension table which saves its white neighbors' id, color, the black neighbors with $id > id_u$ and the corresponding number of each white neighbor in each column.

(3) For each black node u, we assume that there are at most t white neighbors. Note that $t \leq \delta$ and δ is the maximum node degree. Then u colors the white node x_i grey which has the maximum value in the third column of current 2-dimension table, deletes all the common black neighbors between white nodes x_i and

Table 1: The white neighbor list of black node u.

The white neighbors' id	The white neighbors' color	The number of black neighbors with $id > id_u$ of each white neighbor	The black neighbors with $id > id_u$ of each white neighbor
x_1	white	$2 \to 1$	$\{n_1, n_2\} \to \{n_2\}$
x_2	white $\to$ grey	3	$\{n_1, n_3, n_4\}$
—	—	—	—
x_t	white	$2 \to 1$	$\{n_3, n_5\} \to \{n_5\}$

x_j ($j \neq i$ and $1 \leq j \leq t$) and updates the numbers in the third column for remaining white nodes in the table.

(4) For each black node u, repeat step (3) until all numbers in the third column of the table are zero.

As shown in Table 1, we assume that x_2 is the first node which would be colored grey for black node u. Then, u deletes the common nodes n_1 and n_3 in the fourth column and updates the number of black neighbors with $id > id_u$ of x_1 and x_t, respectively. The updated details are presented behind symbol " $\to$ ".

Procedure 2. Coloring3(G, C, V_{2C})

Input: A connected graph G, a black node set C and a grey node set V_{2C}.

Output: A node subset V_{3C} with coloring red.

(1) Each grey node y and white node x send packet (id_y, CL_y, B_y) and (id_x, CL_x, B_x) to its neighbors respectively.

(2) When a grey node y_j which is a neighbor of y has received (id_y, CL_y, B_y), y_j sends packet $CN = (id_{y_j}, CL_{y_j}, id_y, CL_y, B_y)$. When a grey node y_j which is a neighbor of white node x has received (id_x, CL_x, B_x), y_j sends $CN = (id_{y_j}, CL_{y_j}, id_x, CL_x, B_x)$. When a white node x has received (id_y, CL_y, B_y) from its grey neighbor y, x sends $CN = (id_x, CL_x, id_y, CL_y, B_y)$.

(3) For each black node u in B_y, let $Nb_2(u)$ denote the 3-hop black neighbor set of u. Initially, $Nb_2(u) = Nb_1(u)$. When u has received $(id_{y_j}, CL_{y_j}, id_y, CL_y, B_y)$ or $(id_{y_j}, CL_{y_j}, id_x, CL_x, B_x)$ from its grey neighbor y_j, then $Nb_2(u) = Nb_2(u) \bigcup B_y$ or $Nb_2(u) \bigcup B_x$, respectively.

(4) For each black node u in B_x, when u has received $(id_x, CL_x, id_y, CL_y, B_y)$ from its white neighbor x, u saves the corresponding paths from u to B_y into $P(u, B_y)$. For each black node w in $B_y \setminus Nb_2(u)$, u chooses a path $uxyw$ to connect with w. Then, color x red and let $Nb_2(u) = Nb_2(u) \bigcup B_y$.

(5) Each white node x sends (id_x, CL_x, B_x) again. When a white node x_i has received (id_x, CL_x, B_x) from its neighbor x, x_i sends $(id_{x_i}, CL_{x_i}, id_x, CL_x, B_x)$. For each black node u in B_{x_i}, when u has received $(id_{x_i}, CL_{x_i}, id_x, CL_x, B_x)$, it saves the corresponding paths from u to B_x into $P(u, B_x)$ which saves all vertices in the paths. For each black node w in $B_x \setminus Nb_2(u)$ with $id_w > id_u$, a path ux_ixw is chosen to connect u with w. Then, color x_i and x red and let $Nb_2(u) = Nb_2(u) \bigcup B_x$.

Lemma 5. *The message complexity of Algorithm 1 is $O(n^2)$ and the time complexity is $O(n\delta^2)$.*

Proof. By Procedures 1, and 2 and step (3) of Algorithm 1, each node needs to send constant messages to construct V_{2C} and V_{3C}, respectively. The message complexity of step (1) in Algorithm 1 is $O(n^2)$ [17]. Hence, the message complexity of Algorithm 1 is $O(n^2) + O(n) + O(n) = O(n^2)$. In step (3) of Algorithm 1, note that x has at most 5 black neighbors [21]. Therefore, x needs $O(\delta)$ time to compute its black neighbors' information. And node u needs time $O(\delta^2)$ to compute $Nb_1(u)$ in step (2) of Procedure 1. In the step (3) of Procedure 1, the number of rows of a 2-dimension table for node u is at most δ and the value in the fourth column of each row is no more than 5. Thus, each node u needs time $O(\delta^2)$ to choose white nodes such that u connects with $Nb_1(u)$ at the end of step (4). Therefore, the time complexity of Procedure 1 is $n \cdot O(\delta^2 + \delta^2) = O(n\delta^2)$. In steps (3, 4, and 5) of Procedure 2, node u needs time $O(\delta^2)$ for "union" operation to compute $Nb_2(u)$. Hence, the time complexity of Procedure 2 is $n \cdot O(\delta^2 + \delta^2 + \delta^2) = O(n\delta^2)$. In summary, the time complexity of Algorithm 1 is $n \cdot O(\delta) + O(n\delta^2 + n\delta^2) = O(n\delta^2)$. □

After Algorithm 1, we have accomplished a preliminary backbone. Then for each sensor node, it saves the angle information of its neighbors by measuring the direction of wireless signals [22].

Definition 6. Given two vertices u and v, let $a(u,v)$ denote the angle of v relative to u.

For each vertex u in G, let $N(u)$ denote the neighbor set of u within 1-hop. Then, let $A(u) = \{a(u, v) \mid v \in N(u)\}$ refer to the relative angle set of u. Furthermore, by measuring the strength of wireless signals [23], we can obtain the Euclidean length between u and v, which denotes as $d(u,v)$. Hence, we have the following property.

Lemma 7. *Given the destination D and two adjacent vertices u and v, if there exist $d(v, D)$ and $a(v, D)$ of v, then $d(u, D)$ and $a(u, D)$ of u can be computed.*

Proof. First, as shown in Figure 1, let u and v choose the same direction as the reference direction. It is trivial to show that $\angle uvD = \pi - a(v, D) + a(u, v)$ or $\pi + a(v, D) - a(u, v)$.

Then, based on the law of cosine, we have $\cos \angle uvD = -\cos(a(v, D) - a(u, v))$.

Then,

$$d(u, D) = \sqrt{d(v, D)^2 + d(u, v)^2 - 2d(v, D) \cdot d(u, v) \cdot \cos \angle uvD}, \quad (1)$$

Input: A connected graph G.
Output: A node subset V_C.
(1) Adopt the algorithm in [17] to compute a dominating set C in graph G simultaneously.
(2) Color each node in C black and the others white.
(3) Each white node v computes the number of its black neighbors which is referred to k_v.
(4) Call procedure Coloring2(G, C) to color a part of white nodes with $k_v \geq 2$ grey.
(5) Call procedure Coloring3(G, C, V_{2C}) to color a part of remaining white nodes red.
(6) Let $V_C = C \bigcup V_{2C} \bigcup V_{3C}$.

Algorithm 1: Constructing the preliminary planning infrastructure (IRC).

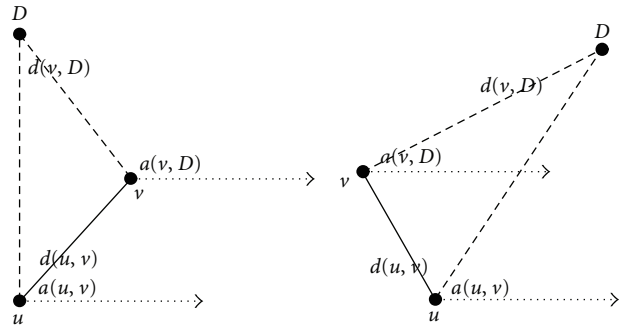


Figure 1: Compute $d(u, D)$ and $a(u, D)$.

$$a(u,D) = \begin{cases} a(u,v) - \angle vuD, & a(u,v) > a(v,D) \\ a(u,v) + \angle vuD, & a(u,v) \leq a(v,D). \end{cases} \quad (2)$$

Therefore, we can obtain $d(u, D)$ and $a(u, D)$ for u. □

By Lemma 7, for each vertex u in graph G, u computes the angle set $A(u)$ and unites it into the preliminary infrastructure which has been built by Algorithm 1. Eventually, we have accomplished a path planning infrastructure. In the following, we propose a path planning algorithm with constant distance cost based on the infrastructure.

4. A Path Planning Algorithm

In [13], Lee et al. proposed the overhead of navigation algorithm. Let S denote the initial position and D be the destination. $M(S,D)$ denotes the length of moving path and $d(S,D)$ denotes the Euclidean length from S to D. Hence, we introduce a general definition.

Definition 8. Given a constant $\lambda > 0$, for any two positions S and D, if a path planning algorithm makes $M(S, D) \leq \lambda \cdot d(S, D)$, then the algorithm guarantees a constant distance cost.

Before proposing our path planning algorithm, the destination data needs to be sent to sensor nodes by the infrastructure.

4.1. Send Destination Data. After the whole area has been monitored by a WSN, some sensor nodes would detect critical events when they have happened in the environment. Supposing that sensor node v has detected the event, then v will confirm the event point D by special measuring modules and transmit the packet (id_v, $d(v,D)$, $a(v,D)$) based on the planning infrastructure. Later, when a sensor node u which is a neighbor of v has received the packet, u could compute $d(u,D)$ and $a(u,D)$ by Lemma 7. Therefore, the whole sensor nodes can gain destination information by communicating with its neighbors.

4.2. A Path Planning Algorithm in WSNs. After sensor nodes in the network have obtained information of destination D, the mobile node which denotes as M with enough energy will move to D automatically using the information stored in sensor nodes. Here, we assume that the communication and sensing range of M are the same with those of sensor node which are R_C and r_s, respectively. Then, we could release M in any position of the environment. For the simplicity of discussion, let M have the same location of a sensor node u in the network. That is, M seems to be u and can obtain $a(M,D)$ which is a duplicate of $a(u,D)$. Therefore, without using localization, M can track its neighbors in the network to arrive at D. In the following, we describe the tracking process in detail.

Note that $N(M) = \{v \mid d(M, v) \leq R_C\}$ denotes the neighbors of M and $A(M) = \{a(M, v) \mid v \in N(M)\}$ refers to the relative angle set. Define $\theta = \angle vMD$ as the include angle of $a(M, v)$ and $a(M, D)$ for each v in $N(M)$. Then, let M choose a neighbor u to make $\theta = \angle uMD$ minimum as its temporary target within range R_C. It is trivial to show that if θ approximates zero, then $a(M, u)$ is the same as $a(M,D)$ which is the optimal direction of movement. In order to restrict the deviation of $a(M, D)$ and $a(M, u)$ by an upper bound, Algorithm 2 claims that the temporary optimal target u should be chosen in the sector $a(M, D) \pm \alpha/2$ ($\alpha < 2\pi/3$) within range R_C. For a randomly deployed WSN with a high density of sensor nodes, we prove that there is at least one sensor node in the chosen sector with high probability which will guarantee a constant distance cost.

For an extreme situation where there is no sensor node in $a(M, D) \pm \alpha/2$ ($\alpha < 2\pi/3$) within range R_C, Algorithm 2 designs a substituted moving path by computing virtual positions in the environment. Note that the network has been modeled as a UDG. If M finds that there is no node for current selection, then M computes a virtual sensor node u' on the direction $a(M,D)$ with $d(M,u') = 1$.

//*Next*(M) saves the temporary targets for M to track.
Input: A mobile node M, an initial position S and a destination D.
Output: A path consists of sensor nodes from S to D.
(1) Place mobile node M at the position S.
(2) $Next(M) = NULL$
(3) **while** M has not arrived at destination D
(4) **do** M updates $a(M,D)$ and chooses an optimal temporary target in $a(M,D) \pm \alpha/2 (\alpha < 2\pi/3)$ within range R_C from $N(M) \setminus Next(M)$
(5) **if** there is no condidate **then**
(6) M computes a virtual sensor node u'
(7) Call Algorithm 1 to update the planning infrastructure
(8) Call Procedure 3 to find a substituted path to bypass u'
(9) **end if**
(10) Add w into $Next(M)$ and let M move to w
(11) **end while**

Algorithm 2: A path planning algorithm for mobile node (MSNA).

By Lemma 4, update Algorithm 1 to make the new path planning infrastructure regard virtual u' as a dominate by setting all the sensor nodes which monitor the position of u' be dominators in the network as dense as possible. Compared with M, u' is much closer to D. For simplicity, we assume there exists at least one candidate sensor node w in the sector $a(M,D) \pm \alpha/2$ $(\alpha < 2\pi/3)$ within range R_C for u'. Then, M can find a feasible solution from M to w in the new infrastructure using shortest vertex path algorithm [18]. Obviously, M cannot communicate with w for $d(M,w) > 1$. Eventually, the algorithm also satisfies a constant distance cost which will be proven in the following.

The detailed algorithm is shown in Algorithm 2.

Procedure 3. Finding a substituted path
Input: $Next(M)$, M and u'.
Output: A temporary target w and a feasible vertex path from M to w.

(1) Compute $a(u', D)$ and choose an optimal temporary target in $a(u', D) \pm \alpha/2$ $(\alpha < 2\pi/3)$ within range R_C from $N(u') \setminus Next(M)$.
(2) Find a shortest vertex path $P(M,w)$ in the path planning infrastructure by the algorithm in [18].

To evaluate the performance of Algorithm 2, we assume that sensor nodes have been randomly deployed in a unit square. Then, we give the probability of nodes in each grid of a partition of this square using Chernoff's bound.

Lemma 9. *Given a randomly deployed node set V and a partition of unit square $[0,1]^2$ into grids with side length l, where $\sqrt{\log n/(c \cdot n)} \le l < 1$, then there exist constant c and $\delta \in (0,1)$, such that each grid contains at least $\delta \cdot \log n/c$ nodes with high probability, where $n = |V|$.*

Proof. Partition $[0,1]^2$ into $cn/\log n$ grids of equal size where $c < 1$. Given a fixed small grid Q_j, where $1 \le j \le cn/\log n$, if node i falls into grid Q_j, then $X_i = 1$, otherwise $X_i = 0$. Here X_i is a random variable. According to the observation, all random variables X_is, where $1 \le i \le n$, are independent and the probability $P(X_i = 1) = \log n/cn$. Let $= \sum_{i=1}^{n} X_i$, then

$$E(X) = E\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} E(X_i) = \frac{n \cdot \log n}{cn} = \frac{\log n}{c}. \quad (3)$$

Applying the Chernoff's bound, we have

$$P\left(X \le \delta \cdot \frac{\log n}{c}\right) \le e^{-((1-\delta)^2/2c)\log n}. \quad (4)$$

So, for all grids in unit square, we denote the number of nodes in $Q_{j'}$ $(Q_{j'} \in \{Q_j\})$ as X', and the probability of $X' < \delta \cdot \log n/c$ is $P(\delta)$, then we get

$$\begin{aligned} P(\delta) &\le \frac{cn}{\log n} \cdot e^{-((1-\delta)^2/2c)\log n} \\ &= e^{-((1-\delta)^2/2c)\log n + \ln(cn/\log n)} \\ &< e^{-((1-\delta)^2/2c)\log n + \log(cn/\log n)}. \end{aligned} \quad (5)$$

Using $c = 0.2$ and $\delta = 0.1$, we obtain

$$\begin{aligned} P(0.1) &\le \frac{cn}{\log n} \cdot e^{-((1-\delta)^2/2c)\log n} \\ &< e^{-((1-\delta)^2/2c)\log n + \log(cn/\log n)} \\ &< e^{-2\log n + \log n + \log c - \log\log n} < \frac{1}{n}. \end{aligned} \quad (6)$$

□

Then, based on Lemma 9, we introduce the probability of sensor nodes existing in the sector $a(M,D) \pm \alpha/2$ $(\alpha < 2\pi/3)$ within range R_C.

Theorem 10. *Given a random node set V and the communicating range R_C of sensor node, if $n \cdot R_C^2 > 8 \cdot \log n$, then there exist constant c and $\delta \in (0,1)$, such that each sector with angle α of the mobile node has $\delta \cdot \log n/c$ neighbors with high probability, where $n = |V|$.*

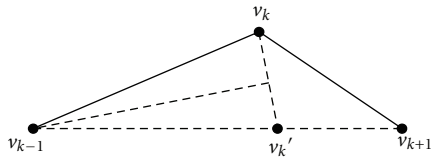


Figure 2: A vertex path from v_{k-1} to v_{k+1}.

Proof. By Lemma 9, when the mobile node M is in a grid Q_j, adjust the communication range such that $R_C = 2\sqrt{2} \cdot l > 2\sqrt{2} \cdot \sqrt{\log n/n}$. Then, we get that the mobile node that can communicate with at least 8 neighbors in its adjacent grids. While the mobile node separates the communicating disk into sectors with angle α, the area of each sector is $S_C = 4\alpha l^2$. Note that if $0.25 < \alpha < 2\pi/3$, then $S_C > l^2$. Denote the probability of each sector contains at least $\delta \cdot \log n/c$ nodes as $P(\delta)$, and then we have

$$P(\delta) \geq 1 - e^{-((1-\delta)^2/2c)\log n + \log(cn/\log n)}. \quad (7)$$

By setting $c = 0.2$, $\delta = 0.1$, and denoting the probability which has at least $\log n/2$ neighbors in a sector as P_r, we obtain $P_r \geq 1 - 1/n$. □

In order to analyze the distance cost of path by Algorithm 2, we propose a lemma when there is no extreme case happening.

Lemma 11. *Let $v_0 v_1 \cdots v_{k+1}$ be the path of mobile node M in Algorithm 2 without any extreme case, where $v_0 = S$ is the initial position and $v_{k+1} = D$ is the destination. Then*

$$M(S,D) \leq \frac{1}{1 - 2\sin(\alpha/4)} \cdot d(S,D) \quad \text{for } \alpha < \frac{2\pi}{3}. \quad (8)$$

Proof. Note that $d(u,v)$ is the Euclidean length from u to v and abbreviates to uv. Based on the choosing rule in Algorithm 2, we know $v_{k-1}v_k < v_{k-1}v_{k+1}$. Set a dot v'_k on dotted line $v_{k-1}v_{k+1}$ satisfying $v_{k-1}v'_k = v_{k-1}v_k$, as shown in Figure 2. Then we have $v_k v_{k+1} < v_k v'_k + v'_k v_{k+1}$ by triangle inequality. Obviously, for any v_i $(1 \leq i \leq k)$, there is $\theta_i = \angle v_i MD \leq \alpha/2$, where $\alpha < 2\pi/3$. Then, $v_k v'_k = 2\sin(\angle v_k v_{k-1} v'_k/2) \cdot v_{k-1}v_k \leq 2\sin(\alpha/4) \cdot v_{k-1}v_k$. Because $v'_k v_{k+1} = v_{k-1}v_{k+1} - v_{k-1}v'_k$, we have $v_{k-1}v_k = v_{k-1}v'_k = v_{k-1}v_{k+1} - v'_k v_{k+1} \leq v_{k-1}v_{k+1} + v_k v'_k - v_k v_{k+1} \leq v_{k-1}v_{k+1} + 2\sin(\alpha/4) \cdot v_{k-1}v_k - v_k v_{k+1}$. Hence, $v_{k-1}v_{k+1} - v_k v_{k+1} \geq (1 - 2\sin(\alpha/4))v_{k-1}v_k$.

Furthermore,

$$\begin{aligned} M(S,D) &= \sum_{i=0}^{k} d(v_i, v_{i+1}) \leq \frac{1}{1 - 2\sin(\alpha/4)} \\ &\quad \cdot \sum_{i=0}^{k} (v_i v_{k+1} - v_{i+1} v_{k+1}) \\ &= \frac{1}{1 - 2\sin(\alpha/4)} v_0 v_{k+1}. \end{aligned} \quad (9)$$

□

Without loss of generality, we assume that there exists only one extreme case during planning a path in Algorithm 2.

Theorem 12. *Let $v_0 v_1 \cdots v_{k+1}$ be the path of mobile node M in Algorithm 2 with an extreme case, where $v_0 = S$ is the initial position and $v_{k+1} = D$ is the destination. Then*

$$M(S,D) \leq \frac{10}{1 - 2\sin(\alpha/4)} \cdot d(S,D) \quad \text{for } \alpha < 2\pi/3. \quad (10)$$

Proof. We assume that the extreme case happens at v_j. That is, v_{j+1} is chosen by call Procedure 3 with $d(v_j, v_{j+1}) > 1$. Let u denote the virtual node. Then, by the path planning infrastructure and Lemma 4, we have that the shortest vertex path $h'(v_j, v_{j+1})$ from v_j to v_{j+1} satisfing $h'(v_j, v_{j+1}) \leq 5(h(v_j, u) + h(u, v_{j+1}))$. Because for any two adjacent sensor nodes u_1 and u_2 in the network which has been modeled as a UDG, we obtain $d(u_1,u_2) \leq h(u_1,u_2)$. Hence, the length of moving path from v_j to v_{j+1} which is denoted as $d'(v_j, v_{j+1})$ satisfies $d'(v_j, v_{j+1}) \leq h'(v_j, v_{j+1})$. By the triangle inequality, there is $d(v_j, u) + d(u, v_{j+1}) > d(v_j, v_{j+1}) > 1$. Then, $h(v_j, u) + h(u, v_{j+1}) = 2 < 2d(v_j, v_{j+1})$. Furthermore, $d'(v_j, v_{j+1}) \leq 10d(v_j, v_{j+1})$. Therefore, by Lemma 11,

$$M(S,D) \leq \frac{10}{1 - 2\sin(\alpha/4)} \cdot d(S,D). \quad (11)$$

□

Note that each v_i is a realistic sensor node. The virtual node is used for updating the path planning infrastructure for an extreme situation under a very low probability. If there are several extreme cases, the proof of Theorem 12 could be extended easily with the same constant ratio.

5. Simulation Results

As mentioned previously, many studies have shown novel algorithms for the infrastructures. In this section, firstly we use VC++6.0 to conduct simulations to compare the performance of algorithm IRC with those of GOC and ICDS in [19, 20], respectively. The area of simulation is a virtual square S_1 of 100×100, and nodes are randomly distributed in S_1. The number of nodes denoted by N is increased by 10 from 10 to 100 and the maximum transmission range R_C is assigned 20, 25, 30, and 35. For distinct nodes u and v, if and only if the Euclidean distance $d(u,v) \leq R_C$, u and v could communicate with each other. For the same settings under different transmission ranges, we randomly create 100 connected graphs for each N and accordingly construct the infrastructure for each connected graph. And for each infrastructure, we compute its size and diameter.

Figure 3 shows the infrastructure sizes of algorithm IRC, GOC, and ICDS under the different R_Cs. In this figure, since more nodes are needed in a bigger network for guaranteed overhead, all the sizes of infrastructures produced by algorithm IRC, GOC, and ICDS increase when the number of nodes increases. For the network with a small amount of nodes, these infrastructure sizes are almost

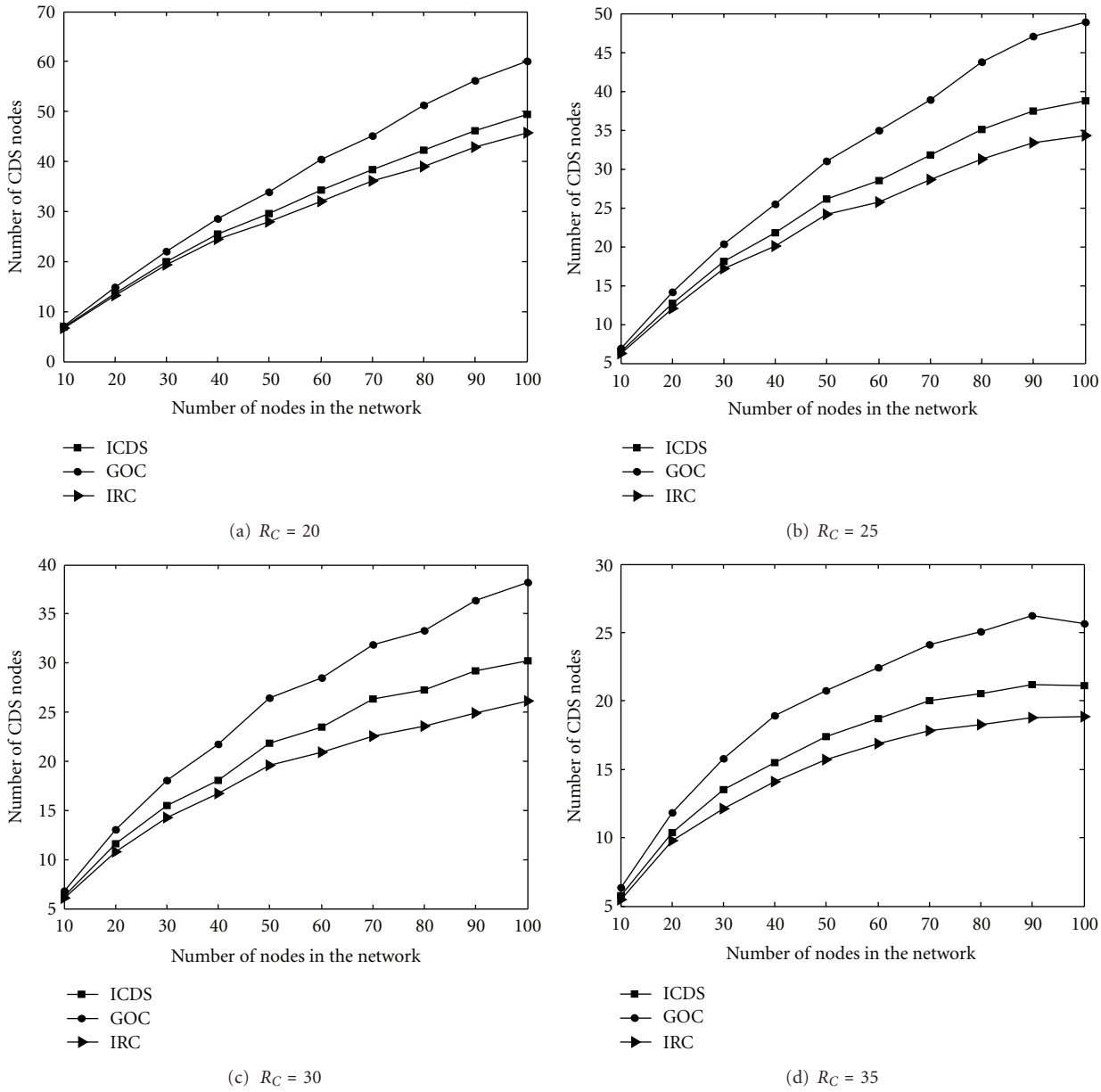


Figure 3: Infrastructure sizes.

equal. While nodes increase, algorithm IRC presents a better performance for different R_Cs.

Figure 4 shows the diameters of graph G and infrastructures produced by algorithm IRC and GOC. When R_C is small, the difference of diameters between two infrastructures and graph G is small. But as R_C increases, the difference goes larger what keeps pace with that of the infrastructure size in Figure 3. However, for different R_Cs, the difference of diameter between algorithm IRC and GOC is very small which implicits that there may exist several redundant nodes in the infrastructure which produced by GOC.

Then, based on the infrastructure which has been constructed, we use VC++6.0 and Matlab 7.0 to evaluate algorithm MSNA. To compare with the distance cost of algorithm ANHC in [13], we set the environment and network parameters to be the same with those in [13]. Table 2 shows the detailed parameters which will be used. We provide four different ways for sensor nodes deployment: (1)

Table 2: Simulation parameters.

Parameters	Value
The monitoring area S_2	$1100 \times 900\,m^2$
Communication range R_C	150 m
Sensing range r_s	15 m
The number of deployed sensor nodes	99, 114, 100, 150
Identifiers	$1 - N$

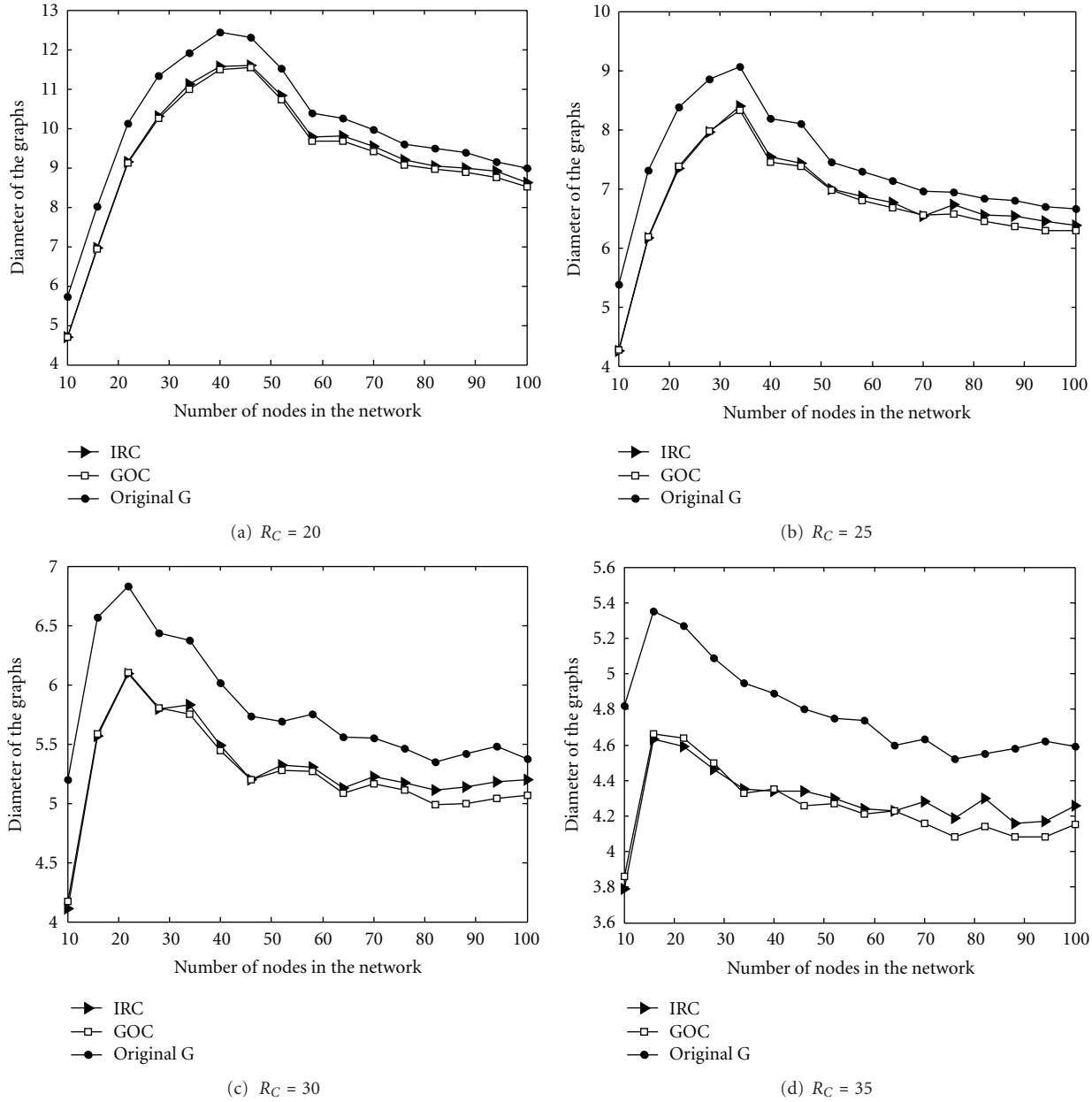


Figure 4: Diameters.

99 nodes are deployed in S_2 uniformly, the interval of each node is 100 m; (2) 114 nodes are deployed in S_2 with a hole in the center; (3) 100 nodes are randomly deployed in S_2; (4) 150 nodes are randomly deployed in S_2.

In Figure 5, four different planning paths are presented under corresponding deployed ways. By tracking the realistic sensor nodes in the network, all the paths can be defined as directed vertex paths in the graph which is modeled by a WSN. In each figure, we use the symbol "*" and "☆" to stand for the initial position and destination of a planning path, respectively. The dots which are encircled by "△" are represented as the sensor nodes tracked by mobile node. A dashed circle denotes the transmitting range of wireless signal. In Figure 5(a), at the beginning, the mobile node chooses the optimal sensor nodes from its neighbors as the temporary target. Without using a localization technique, the mobile node tracks the temporary targets which could be computed by algorithm MSNA in the uniform network. In Figure 5(b), for the given initial position, although there is a hole in the center of S_2, the mobile node has not encountered any extreme cases during selecting temporary target. So, the path consists of a node set alongside the border of hole. In Figure 5(c), an extreme case has happened in algorithm MSNA. The two square symbols "□" show the virtual nodes which were computed in algorithm MSNA with being closer to the destination in the path planning infrastructure. In Figure 5(d), the network is quite dense such that there is no extreme case for the mobile node. For the locations of sensor

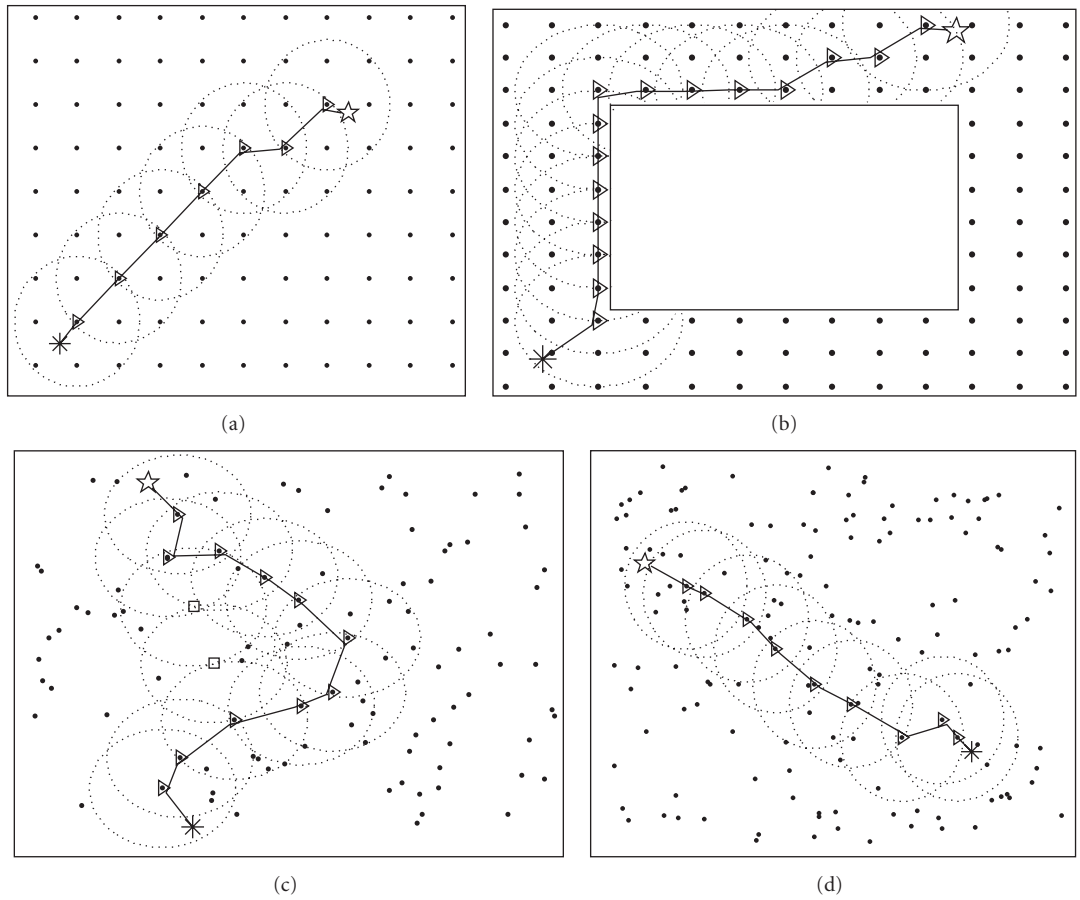


Figure 5: Trajectories of the mobile node by MSNA.

nodes in the path, we find that they almost distributed along the straight line from start to destination.

In [13], the algorithm ANHC using average hop-count of neighbors is the first one concerning the cost of a navigating path without localization. In the initial phase, each sensor node sets up its hop count value to the destination. Then, every sensor node computes the value *anhc* by communicating with its neighbors. It implicated that sensor nodes which are closer to the destination would have smaller *anhc* compared with the ones which are far away from the destination. The mobile node computes its *anhc* at its present location. By judging the variation of value *anhc*, the mobile node revises its direction. The disadvantage of this algorithm is that although the decreasing of *anhc* shows that the mobile node is moving to the destination, it cannot indicate the deviation between current direction and the optimal direction which may lead to a high cost. In algorithm MSNA, at current position, the mobile node chooses the optimal temporary target which makes the deviation between direction of movement and the optimal direction be minimum. But there also exists the disadvantage in MSNA because we cannot guarantee there must have candidates in the sector $a(M,D) \pm \alpha/2$ ($\alpha < 2\pi/3$) within R_C for mobile node M.

In the following, we randomly set the initial position S and the destination D with $200\,\text{m} \leq d(S,D) \leq 1000\,\text{m}$. Through employing a lot of randomly deployed networks, the average costs of paths have been presented in Figure 6 for the predefined shortest Euclidean distance.

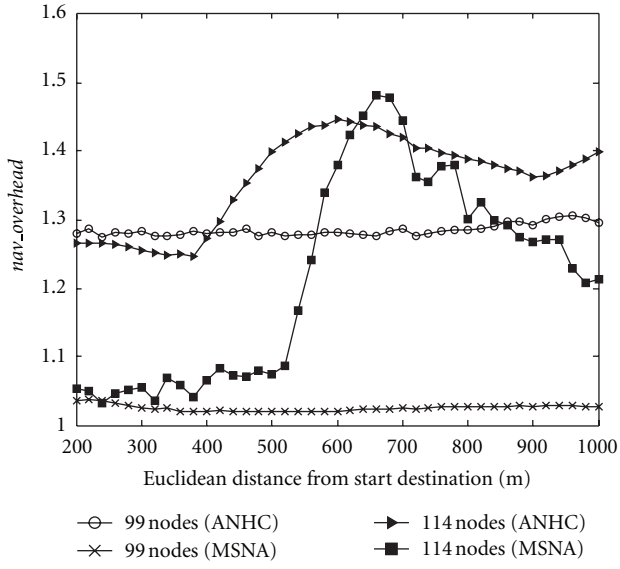


Figure 6: The path cost for a uniform deployed network.

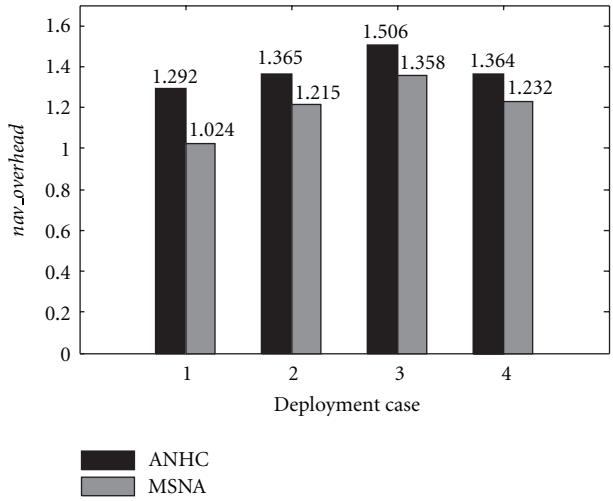


Figure 7: The average path cost for four different deployed ways.

It shows that $nav\_overhead_{\mathrm{ANHC}} > nav\_overhead_{\mathrm{MSNA}}$ for the first deployed way. And under the second way, $nav\_overhead_{\mathrm{ANHC}} > nav\_overhead_{\mathrm{MSNA}}$ is almost true for different predefined situations.

Figure 7 shows that the average costs of paths under four different deployed ways of the network. We randomly set the initial position S and the destination D in the environment. Through a lot of simulations, we find that the cost of algorithm MSNA is smaller than the algorithm ANHC for each deployed way.

6. Conclusion

In this paper, by characterizing the navigation problem as a path planning problem, we first present a distributed algorithm to construct a path planning infrastructure by uniting the neighbors' information of each sensor node into a CDS. Then, we propose a path planning algorithm to generate an effective path in the network even under an extreme case. We prove that the distributed algorithm has low time and message complexity and the path planning algorithm guarantees a constant distance cost. Simulation results show that the algorithms produce a smaller infrastructure size and a distance cost. Due to frequent node and link failure, which are inherent in WSNs, to construct a robust a path planning algorithm is our further work.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grants no. 61070169, 61170021 and 61201212, The Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, The Specialized Research Foundation for the Doctoral Program of Higher Education of China no. 20103201110018, The Application Foundation Research of Suzhou of China No. SYG201118, SYG201240, SYG201239 and sponsored by the Qing Lan Project.

References

[1] F. M. Al-Turjman, H. S. Hassanein, and M. A. Ibnkahla, "Connectivity optimization for wireless sensor networks applied to forest monitoring," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–6, June 2009.

[2] O. A. Postolache, J. M. Dias Pereira, and P. M. B. Silva Girão, "Smart sensors network for air quality monitoring applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 3253–3262, 2009.

[3] C. W. Chen and Y. Wang, "Chain-type wireless sensor network for monitoring long range infrastructures: architecture and protocols," *International Journal of Distributed Sensor Networks*, vol. 4, no. 4, pp. 287–314, 2008.

[4] K. Casey, A. Lim, and G. Dozier, "A sensor network architecture for Tsunami detection and response," *International Journal of Distributed Sensor Networks*, vol. 4, no. 1, pp. 28–43, 2008.

[5] J. Borenstein and H. R. Everett, *Navigating Mobile Robots: Sensors and Techniques*, John Wiley & Sons, New York, NY, USA, 1992.

[6] P. Sinha, R. Sivakumar, and V. Bharghavan, "Enhancing ad hoc routing with dynamic virtual infrastructures," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, pp. 1763–1772, April 2001.

[7] R. Fonseca, O. Gnawali, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 1–14, usa, November 2009.

[8] T. K. Moon and T. Y. Kuc, "An integrated intelligent control architecture for mobile robot navigation within sensor network environment," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, pp. 565–570, October 2004.

[9] A. J. Briggs, C. Detweiler, D. Scharstein, and A. Vandenberg-Rodes, "Expected shortest paths for landmark-based robot navigation," *International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 717–728, 2004.

[10] Q. Li, M. De Rosa, and D. Rus, "Distributed Algorithms for Guiding Navigation across a Sensor Network," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 313–325, September 2003.

[11] T. Jiang, Y. Yi, Q. Zhang, and K. Zhang, "Novel navigation algorithm for wireless sensor networks without information of locations," in *Proceedings of the Global Communications Conference (GLOBECOM '11)*, pp. 1–6, 2011.

[12] S. Li, A. Zhan, X. Wu, P. Yang, and G. Chen, "Efficient emergency rescue navigation with wireless sensor networks," *Journal of Information Science and Engineering*, vol. 27, no. 1, pp. 51–64, 2011.

[13] W. Y. Lee, K. Hur, and D. S. Eom, "Navigation of mobile node in wireless sensor networks without localization," in *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI '08)*, pp. 1–7, August 2008.

[14] M. Li, Y. Liu, J. Wang, and Z. Yang, "Sensor network navigation without locations," in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM '09)*, pp. 2419–2427, April 2009.

[15] Y. C. Tseng, M. S. Pan, and Y. Y. Tsai, "Wireless sensor networks for emergency navigation," *Computer*, vol. 39, no. 7, pp. 55–62, 2006.

[16] W. Wei and Y. Qi, "Information potential fields navigation in wireless Ad-Hoc sensor networks," *Sensors*, vol. 11, no. 5, pp. 4794–4807, 2011.

[17] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14–25, 2002.

[18] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, pp. 325–336, 1984.

[19] H. Du, Q. Ye, W. Wu et al., "Constant approximation for virtual backbone construction with Guaranteed Routing Cost in wireless sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 1737–1744, April 2011.

[20] Y. Wang and X. Y. Li, "Geometric spanners for wireless ad hoc networks," in *Proceedings of the 22nd IEEE International Conference on Distributed Systems (ICDCS '02)*, pp. 171–178, July 2002.

[21] P. J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," *Mobile Networks and Applications*, vol. 9, no. 2, pp. 141–149, 2004.

[22] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1734–1743, April 2003.

[23] L. Girod, V. Bychkovskiy, J. Elson, and D. Estrin, "Locating tiny sensors in time and space: a case study," in *Proceedings of the International Conference on Computer Design (ICCD '02) VLSI in Copmuters and Processors*, pp. 214–219, September 2002.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 803840, 6 pages
doi:10.1155/2012/803840

Research Article

Novel Node Localization Algorithm Based on Nonlinear Weighting Least Square for Wireless Sensor Networks

Fu Xiao,[1,2,3] Mingtan Wu,[1] Haiping Huang,[1,3,4] Ruchuan Wang,[1,3,4] and Sudan Wang[1,3]

[1] *School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2] *Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, China*
[3] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China*
[4] *Key Lab of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education, Nanjing 210003, China*

Correspondence should be addressed to Fu Xiao, xiaof@njupt.edu.cn

Received 2 August 2012; Accepted 11 October 2012

Academic Editor: Shan Lin

Positioning of the location information of wireless sensor network nodes is one of the key issues in wireless sensor network applications. The traditional node positioning method based on least-square algorithm heavily depends on the ranging accuracy, and therefore cannot guarantee high precision. This paper presents a new method for wireless sensor network node positioning based on nonlinear weighting least-square algorithm. Regarding ranging equation error-weighted sum as a whole, this method starts with the initial iteration point of stepwise refinement to explore the optimal solution and further reduces the positioning computational complexity by the simplification of the Taylor equation. Experimental results demonstrate that promising results have been achieved by using this method.

1. Introduction

Wireless sensor networks are widely used in military and industrial fields due to its strong ability to acquire information, high-level autonomy, broad coverage, long life cycle, and aniattenuation under harsh environment, and so forth. Sensor nodes, as the basic units of wireless sensor networks, which generally use limited battery-powered energy, are difficult to achieve the later energy resources supply once layout. Sensor nodes can obtain a variety of information from the surrounding environment, such as temperature, humidity, gas concentration, pulse, oxygen, and so on. The positioning of the location information of the wireless sensor network nodes is one of the key issues of wireless sensor network applications. If there is no access to the node location information, monitoring information obtained by the sensor would become meaningless in many scenarios. Target tracking and attacking is asked to provide location information [1]. In addition, node location information is very beneficial to network coverage quality and routing efficiency [2].

Wireless sensor networks positioning methods can be classified in various manners. For example, depending on whether to measure distances or not, they can be categories into nonranging-based approach [3] and ranging based approach; depending on where to perform algorithm, they can be categories into distributed approach [4] and centralized approach [5]. Non-ranging approach is to measure the connections between nodes to obtain the network connectivity and then compute node location, such as the MDS-MAP [6], APTI, DV-HOP, and so on. On the other hand, the ranging method first obtains information, including distance or angle between the nodes, by ranging techniques such as Received Signal Strength Indicator (RSSI) [7], Time-Difference-of-Arrival (TDOA) [8], and Angle-of-Arrival (AOA). The next step is to predict location information of unknown nodes by the positioning algorithm, such as the weighted cancroids algorithm [9] and least-square algorithm [10]. RSSI-based ranging technique [11] is a very popular ranging based positioning method. Most existing node hardware has the RSSI function, which does

not require additional hardware and software support. RSSI-based positioning are widely used in practical applications because of its low power consumption for node positioning, as well as small size requirements [11]. However, the accuracy of RSSI-based ranging method may be compromised due to the impact of external environment and other factors.

The traditional node positioning method based on least-square algorithm heavily depends on the ranging accuracy, and therefore cannot guarantee an accurate positioning. The Gaussian filtering and Taylor formula are introduced into this paper which based on the nonlinear least-square method in paper [10]. This paper presents a new method for wireless sensor network node positioning based on nonlinear weighting least-square algorithm. The main contributions of this paper including: (1) introducing a channel propagation model for measuring the distance between unknown nodes and anchor nodes, and (2) enhancing measurement accuracy by the Gaussian filtering method. In this manner, this paper has improved wireless sensor network localization algorithm based on the weighted nonlinear least square to achieve high-precision positioning. Compared with the traditional linear equations consisting of the ranging equations of weighted least-square sum, this algorithm achieves high-precision node localization. Simulation results demonstrate the effectiveness of the method.

2. Wireless Channel Models and Node Distance Measurement

2.1. Channel Model of Wireless Sensor Networks. The RSSI-based distance measurement has received widespread attention for its no requirement of complex hardware support, the outstanding advantages of the node energy consumption and low size overhead. In this paper, we adopt the positioning-based ranging algorithm, by which the distance can be measured efficiently. The unknown node first receives the RSSI value from the anchor node and further relies on the network channel model to estimate the distance (parameter d) from the unknown node to anchor node. Since the transmission of node radio waves can be influenced by many factors, such as multipath effects, scattering, the signal that an unknown node receives from an anchor node may have a significant fluctuation. Researchers have developed multiple channel models for small indoor environment and outdoor free space environment [12]. Lognormal channel model is widely used in common RSSI techniques for its simplicity and good match of the relationship between signal attenuation and distance. To be specific, the channel model formula is as follows:

$$p_r\,[\mathrm{dBm}] = p_0\,[\mathrm{dBm}] - 10\eta \log d + X_\sigma, \tag{1}$$

where p_r denote the signal power received by an unknown node, p_0 is the received signal power by an unknown node with a 1 m distance to an anchor node. d is the real distance between an unknown nodes and an anchor nodes. η is the channel fading index. Its value depends on the propagation environment, typically ranging from 2 to 3. X_σ is the Gaussian noise with mean value is 0 and variance σ^2, which is assumed to simulate the random component of the channel [10].

We can conclude from (1) that the RSSI value received from the same anchor node contains Gaussian distributed noises. Therefore, the corresponding RSSI value RSSI should consequently follow a Gaussian distribution. Before calculating distance by RSSI, we take into consideration Gaussian filtering for the received RSSI, for the purpose of filtering out certain values which have great deviations from the actual value.

2.2. Distance Measurement Based on Gaussian Filtering. Assuming that the unknown node received k RSSI value in total from an anchor node, and unknown node receives the ith RSSI value, which is expressed as RSSI_i, we know from (1) that the RSSI value received by unknown node is subject to $N(\alpha, \beta^2)$. α is the mean value of the right part of (1), which is equivalent to $p_0[\mathrm{dBm}] - 10\eta \log d$. Since it is constant, we can infer that $\beta = \sigma$.

With a received number of RSSI, we can calculate α and β by the maximum likelihood estimation method as follows:

$$\begin{aligned} \alpha &= \frac{1}{k}\sum_{i=1}^{k}\mathrm{RSSI}_i, \\ \beta^2 = \sigma^2 &= \frac{1}{k}\sum_{i=1}^{k}(\mathrm{RSSI}_i - \alpha)^2. \end{aligned} \tag{2}$$

Therefore, the RSSI value received is subject to the probability density function

$$f(x) = \frac{1}{\sqrt{2\pi\beta}} e^{-(x-\alpha)^2/2\sigma^2}. \tag{3}$$

In order to filter out some RSSI values with significant deviations, we first select a high probability range to filter out the values exceeding the selected probability range

$$\begin{aligned} &p(\alpha - t \le x \le \alpha + t) = 0.6 \\ &\int_{-\infty}^{\alpha+t} \frac{1}{\sqrt{2\pi\beta}} e^{-(x-\alpha)^2/2\sigma^2} dx = \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\left(\frac{t}{\sqrt{2}\beta}\right) \\ &\qquad = 1 - \frac{1-0.6}{2} = 0.8, \\ &\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-k^2} dk. \end{aligned} \tag{4}$$

We can derive that

$$t = 0.6\sqrt{2}\sigma. \tag{5}$$

In order to reduce the RSSI signal interference, we take value from the range $\mathrm{RSSI} \in [\alpha - t, \alpha + t]$, then calculate the mean value $\overline{p_r}$. Since RSSI value in (1) and the X_σ in the distance equation are both subject to $N(0, \sigma)$, we can derive that X_σ is subject to the probability density function

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}. \tag{6}$$

Following (1), we have

$$\widetilde{d} = e^{\frac{p_0 + x - \overline{p_r}}{10\eta}}. \tag{7}$$

Since $x \sim N(0, \sigma)$, the final value of d can be expressed as follows:

$$d = E\left(\widetilde{d}\right) = \int_{-\infty}^{+\infty} e^{(p_0 + x - \overline{p_r})/10\eta} g(x)dx = \left(\overline{p_r}\right)^{-1/\eta} e^{-\beta/2\eta^2\gamma}. \tag{8}$$

We know that $\gamma = 10/\ln(10)$, so d is the final distance between an unknown node and an anchor node.

3. Linearized Least-Square Method and Its Limitations

With the distance information that an unknown node obtains from N anchor nodes using above method, we can list N equations. Since there exist errors in distance measurements, if we calculate the N equations, the equation will be no solution. In order to highlight the contribution of the proposed nonlinear least-square approach in this paper, we use "linear least-square" to denote the traditional approach. By transforming nonlinear equations into linear equations, we figure out the estimated least-square solution [13–15].

The ith anchor node coordinate is $(x_i, y_i), i = 1, 2, 3, \ldots, N$. Parameter i is the number of anchor nodes. Parameter d_i, the distance calculated by the RSSI received by unknown node, may deviate from the real distance value. Assuming that the unknown node coordinate is (x, y), the N equations can be expressed as follows:

$$\begin{aligned} &\sqrt{(x_1 - x)^2 + (y_1 - y)^2} = d_1 \\ &\sqrt{(x_2 - x)^2 + (y_2 - y)^2} = d_2 \\ &\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ &\sqrt{(x_n - x)^2 + (y_n - y)^2} = d_n. \end{aligned} \tag{9}$$

Square both sides of (9),

$$\begin{aligned} &(x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ &(x_2 - x)^2 + (y_2 - y)^2 = d_2^2 \\ &\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ &(x_n - x)^2 + (y_n - y)^2 = d_n^2. \end{aligned} \tag{10}$$

Linear least-square solution is as follows. Eliminating the nonlinear part by minus the last equation with each else equation in (10), then we can obtain an overdetermined linear equations (12). Generally, the number of equations is larger than the number of variables, the exact solution cannot be obtained, and we can obtain estimated solution through the least square. The linear equation is as follows:

$$AX = B,$$

$$A = \begin{bmatrix} 2(X_1 - X_N) & 2(Y_1 - Y_N) \\ \vdots & \vdots \\ 2(X_{N-1} - X_N) & 2(Y_{N-1} - Y_N) \end{bmatrix}. \tag{11}$$

Number in A is $a_{ij}, i, j = 1, 2, \ldots, N-1$,

$$B = \begin{bmatrix} x_1^2 - x_N^2 + y_1^2 - y_N^2 + d_N^2 - d_1^2 \\ \vdots \\ x_{N-1}^2 - x_N^2 + y_{N-1}^2 - y_N^2 + d_N^2 - d_{N-1}^2 \end{bmatrix}. \tag{12}$$

The number in B is $b_i, i = 1, 2, 3, \ldots, N-1$.

We can figure out $\overline{X} = (A^T A)^{-1} A^T B$, the estimated value of the linear least square. In fact, it is the minimal solution of each equation bias

$$\min F(x) = \sum_{i=1}^{N-1} \left(b_i - \sum_{j=1}^{j=2} a_{ij} x_j \right)^2. \tag{13}$$

Since (12) is obtained by subtracting the last equation from all other equations in (10), we can assume that

$$d_i^2 - (x_i - x)^2 - (y_i - y)^2 = \ell_i. \tag{14}$$

As a matter of fact, the linear least square is the solution of $\min \sum_{i=1}^{N-1} (\ell_i - \ell_N)^2$. Therefore, the solution depends not only on the accuracy of the solution of the last equation in (10), but also on the accuracy of ℓ_N. If the last equation contains large deviation, the solution of (10) will have large deviation as well.

4. Node Positioning Based on Weighted Nonlinear Least Square

Due to the accuracy loss caused by linearization with the method of the linear least-square, this paper improves node positioning accuracy of the weighted nonlinear least-square method on basis of the literature [10]. By directly calculating the sum of squared errors to eliminate the dependency of the solution on the accuracy of any equation, we can obtain the optimal solution from the overall information, as well as reduce the computational complexity. In this paper, we commence the nonlinear part of the equation using the Taylor formula to reduce the computation intensity in the node. Simulation results show that this method does better in positioning than the linear least-square method.

Assuming the parameter in (9),

$$\xi_i = d_i - \sqrt{(x_i - x)^2 + (y_i - y)^2}, \quad i = 1, 2, \ldots, N \tag{15}$$

ξ_i is the deviation between the distance d_i of unknown node and the distance of the positing node. Empower

the value of k_i when calculating deviation. Since the greater the distance the greater the error, the corresponding weights should also be smaller, so that the impact of the ranging errors on positioning accuracy can be reduced. In this paper, we empower every ξ_i with the value of k_i

$$k_i = \frac{1}{\left(d_i \cdot \sum_{j=1}^{j=N} 1/d_j\right)}. \tag{16}$$

So the total deviation is as follows:

$$\zeta = \sum_{i=1}^{N} k_i \xi_i^2 = \sum_{i=1}^{N} k_i \left(d_i - \sqrt{(x_1 - x)^2 + (y_1 - y)^2}\right)^2. \tag{17}$$

Precise node positioning requires the solution of the corresponding value x, y when ζ obtains the minimum value.

The general solution of $\min \zeta$ can be obtained by a nonlinear optimization method that iterates to find the optimal value in the feasible region [8, 16]. This calculation is too much, and computing has the potential to fall into local optimal solution, so direct nonlinear optimization method is not appropriate for positioning node [17].

In this paper, we simplify ζ by commencing the nonlinear part of the equation using the Taylor formula to reduce the amount of computation, so we have a quadratic function $g(x, y)$ in terms of x, y. Therefore, the method is independent on the other equations and reduces the computation burden with minor loss of accuracy during commencing Taylor formula, quite applies to computing the resource-constrained sensor network nodes

$$\begin{aligned}\zeta &= \sum_{i=1}^{N} k_i \xi_i^2 = \sum_{i=1}^{N} k_i \left(d_i - \sqrt{(x_1 - x)^2 + (y_1 - y)^2}\right)^2 \\ &= -2x \sum_{i=1}^{N} k_i x_i - 2y \sum_{i=1}^{N} k_i y_i + x^2 \sum_{i=1}^{N} k_i + y^2 \sum_{i=1}^{N} k_i \\ &\quad + 2 \sum_{i=1}^{N} k_i d_i \sqrt{(x_i - x)^2 + (y_i - y)^2}.\end{aligned} \tag{18}$$

The first order Taylor series expansion $\sqrt{(x_i - x)^2 + (y_i - y)^2}$ to figure out an similar equation to simplify the method.

Assuming that

$$f_i(x, y) = \sqrt{(x_i - x)^2 + (y_i - y)^2}. \tag{19}$$

So the first order Taylor series expansion $f_i(x, y)$ in (x_0, y_0) is as follows:

$$\begin{aligned}f_i(x, y) &= f_i(x_0 + h, y_0 + p) \\ &= \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2} \\ &\quad + \frac{(x_0 - x_i)}{\sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}} (x - x_0) \\ &\quad + \frac{(y_0 - y_i)}{\sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}} (y - y_0).\end{aligned} \tag{20}$$

Assuming that

$$\begin{aligned}\sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2} &= M_i, \\ \frac{(x_0 - x_i)}{\sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}} &= A_i, \\ \frac{(y_0 - y_i)}{\sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2}} &= B_i.\end{aligned} \tag{21}$$

So,

$$f_i(x, y) = M_i + A_i(x - x_0) - B_i(y - y_0). \tag{22}$$

M_i, A_i, B_i are constants. After the first order Taylor series expansion, ζ is the quadratic form of x, y, is easy to solve

$$\begin{aligned}\zeta = &-2x \sum_{i=1}^{N} k_i x_i - 2y \sum_{i=1}^{N} k_i y_i + x^2 \sum_{i=1}^{N} k_i + y^2 \sum_{i=1}^{N} k_i \\ &+ 2 \sum_{i=1}^{N} k_i d_i (M_i + A_i(x - x_0) - B_i(y - y_0)).\end{aligned} \tag{23}$$

Getting rid of the transformation in the form ζ can be expressed as follows:

$$\begin{aligned}\zeta = &2x \sum_{i=1}^{N} (k_i d_i A_i - k_i x_i) + 2y \sum_{i=1}^{N} (k_i d_i B_i - k_i y_i) \\ &+ x^2 \sum_{i=1}^{N} k_i + y^2 \sum_{i=1}^{N} k_i.\end{aligned} \tag{24}$$

In order to compute $\min \zeta$,

$$\begin{aligned}\frac{\partial \zeta}{\partial x} &= 0, \\ \frac{\partial \zeta}{\partial y} &= 0.\end{aligned} \tag{25}$$

The solution is as follows:

$$\begin{aligned}x &= \frac{\sum_{i=1}^{N} (k_i x_i - k_i d_i A_i)}{\sum_{i=1}^{N} k_i}, \\ y &= \frac{\sum_{i=1}^{N} (k_i y_i - k_i d_i B_i)}{\sum_{i=1}^{N} k_i}.\end{aligned} \tag{26}$$

The point Taylor expansion (x_0, y_0) has a great influence on the effect of positioning. The method proposed in this paper aims at exploring a point close to the given initial point, which complies with the above formula to achieve a relatively small positioning error. Therefore, we generally choose the center of mass of the unknown node received from the anchor point to perform Taylor series expansion, so the computational complexity is relatively simple.

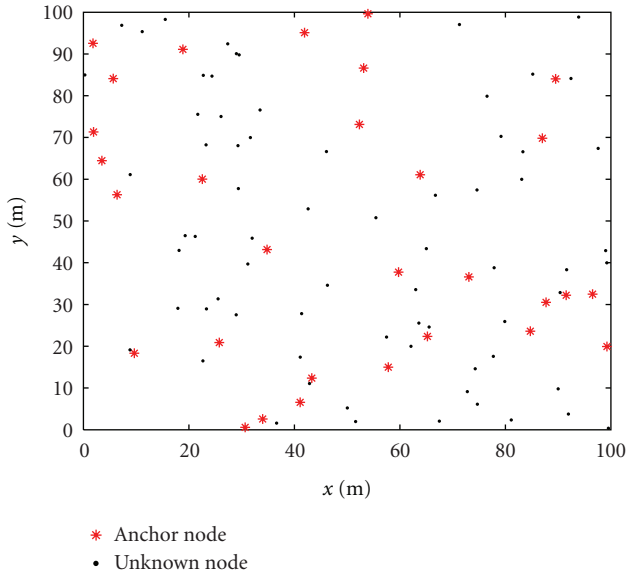


Figure 1: Node distribution diagram.

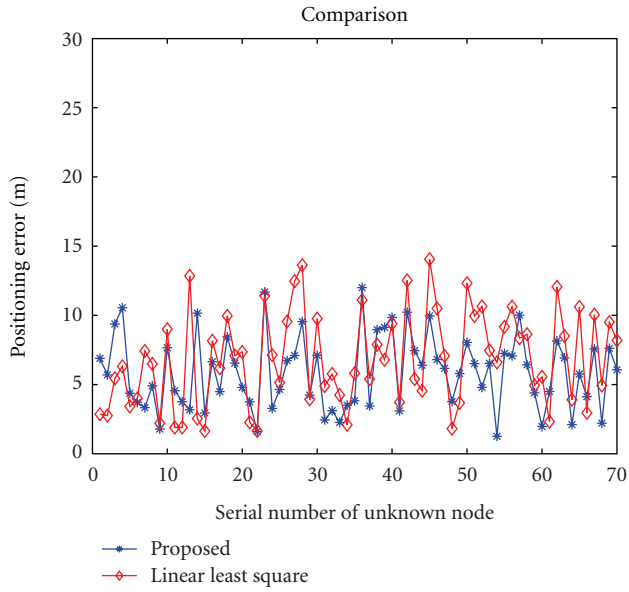


Figure 2: Positioning errors of the two positioning algorithms.

5. Simulation and Analysis

In this paper, we carried out many simulation experiments about positioning method. The experiments were performed by Matlab programming with a large number of unknown nodes and anchor nodes in multiple area.

Experiment 1. Arrange randomly 100 nodes in the regional area 100 m × 100 m, among them there are 70 unknown nodes and 30 anchor nodes, respectively. This paper presents the weighted nonlinear least-square method and compare with the conventional linear least-square method. We have simulated the two methods by Matlab programming for 100 times and obtained the positioning error of each node by both algorithms.

Figure 1 is the node distribution diagram of one simulation. The average error caused by the weighted nonlinear least-square method is 5.8710, while the error by linear least-square method is 6.8610. The weighted nonlinear approach is advantageous to the linear approach. From Figure 2, we can see that some point positioning error mutations in the positioning of the linear least-square method for the reason we mentioned before. The positioning accuracy depends on the ranging error of the last equation in (10): if the ranging error in the equation is large, positioning effect will accordingly be poor. The proposed algorithm does not depend on other ranging errors, and the positioning error of the proposed algorithm is relatively stable. Experiments verified the effectiveness of the proposed algorithm.

Experiment 2. Arrange randomly 100 nodes in the regional area 100 m × 100 m, change the ratio of anchor nodes to unknown nodes, and select five different ratios for 100 simulation runs to compare the two algorithms in

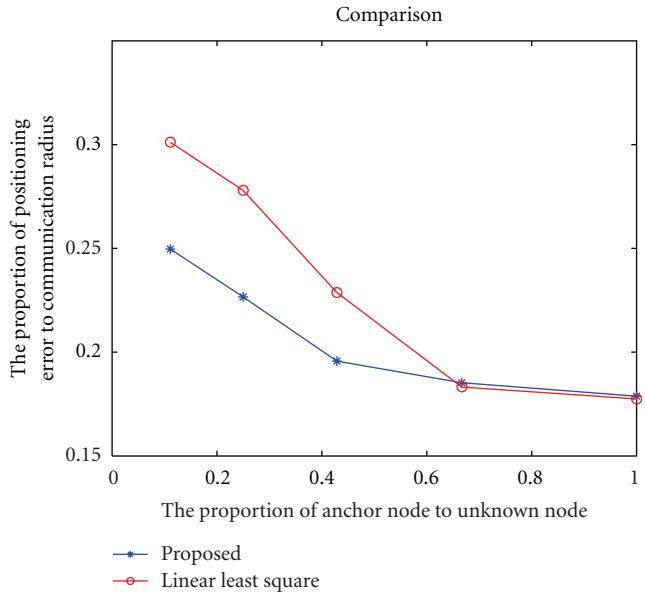


Figure 3: The relationship of two positioning algorithm effects.

positioning errors trend as the ratio of anchor nodes to unknown nodes changes.

It can be observed from Figure 3 that the performance of the proposed algorithm is better when the ratio of anchor nodes to unknown nodes is small, because the coverage of the anchor nodes in the region is relatively sparse in this situation. Relatively, the error also increases as the distance between unknown nodes and anchor nodes increases. Compared with the linear least-squares location method, the proposed algorithm is less dependent on ranging. Figure 3 shows that both algorithms indicate a decreasing trend of positioning error when the ratio of unknown nodes to

anchor nodes in the network goes up. With an increasing number of the anchor nodes, the positioning error of the two algorithms has been reduced accordingly. The distance between the two segments gradually decreases as the ratio of anchor nodes to unknown nodes increases. This is because, with the increase of the ratio, the accuracy loss of linear least-square approach caused by ranging error decreases, and the number of anchor nodes that unknown nodes can receive from will increase which compensates for the loss of accuracy to a certain extent.

6. Conclusion

In this paper, we have proposed a node positioning method based on nonlinear weighting least square to address the problem of positioning accuracy loss in the traditional least-square linear equation. In addition, this paper proposed a Gaussian filter to improve the ranging accuracy. On this basis, we have also proposed wireless sensor network localization algorithms based on the weighted nonlinear least square to achieve high-accuracy positioning. In contrast to the traditional linear equations consisting of the ranging equation of weighted least-square sum, the algorithm achieves high-precision node localization. Experimental results demonstrate the effectiveness of the method.

Acknowledgments

This paper is sponsored by the National Natural Science Foundation of China (61003236, 61170065), the Natural Science Foundation of Jiangsu (BK2011755), Scientific Technological Support Project of Jiangsu (BE2012183, BE2012755). The Project is sponsored by Jiangsu Provincial Research Scheme of Natural Science for Higher Education Institutions (11KJB520016), Scientific Research and Industry Promotion Project for Higher Education Institutions (JHB2012-7), Doctoral Fund of Ministry of Education of China (20103223120007), and Priority Academic Program Development of Jiangsu Higher Education Institutions (information and communication).

References

[1] C. XiaoMei and B. Yu, "Sensor node localization system security," *Journal of Software*, vol. 19, no. 4, pp. 869–877, 2008.

[2] F. B. Wang, L. Shi, and F. Y. Ren, "Self-localization systems and algorithms for wireless sensor networks," *Journal of Software*, vol. 16, no. 5, pp. 857–868, 2005.

[3] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[4] D. F. Larios, J. Barbancho, F. J. Molina, and C. Leon, "Localization based on an intelligent distributed fuzzy system applied to a WSN," *Ad Hoc Networks*, vol. 10, no. 3, pp. 604–622, 2012.

[5] J. A. Jiang, C. L. Chuang, T. S. Lin et al., "Collaborative localization in wireless sensor networks via pattern recognition in radio irregularity using omnidirectional antennas," *Sensors*, vol. 10, no. 1, pp. 400–427, 2010.

[6] Y. Shang and W. Ruml, "Improved MDS-based localization," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2640–2651, Hong Kong, China, March 2004.

[7] Z. L. Zeng and J. M. Gao, "Corrected range weighted centroid localization algorithm based on RSSI for WSN," in *Proceedings of the International Conference on Informatics, Cybernetics, and Computer Engineer*, pp. 453–460, November 2011.

[8] X. Y. Sun, J. D. Li, H. Pengyu, and P. Jiyong, "Total least-squares solution of active target localization using TDOA and FDOA measurements in WSN," in *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications Workshops (AINA '08)*, pp. 995–999, March 2008.

[9] W. Chen and W. F. Li, "RSSI-based wireless sensor networks weighted centroid localization algorithm," *Journal of Wuhan University of Technology*, vol. 30, no. 2, pp. 16–22, 2006.

[10] P. Tarrio, A. M. Bernardos, and J. R. Casar, "Weighting least-square techniques for improved received signal strength based localization," *Sensors*, vol. 11, no. 9, pp. 8569–8592, 2011.

[11] S. Gezici, "A survey on wireless position estimation," *Wireless Personal Communications*, vol. 44, no. 3, pp. 263–282, 2008.

[12] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall PTR, 2nd edition, 2001.

[13] T. K. Sarkar, Z. Ji, K. Kim, A. Medouri, and M. Salazar-Palma, "A survey of various propagation models for mobile communication," *IEEE Antennas and Propagation Magazine*, vol. 45, no. 3, pp. 51–82, 2003.

[14] P. Tarrío, A. M. Bernardos, J. A. Besada, and J. R. Casar, "A new positioning technique for RSS-based localization based on a weighted least squares estimator," in *Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS '08)*, pp. 633–637, October 2008.

[15] J. A. Costa, N. Patwari, and A. O. Hero, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 1, pp. 39–64, 2006.

[16] B. C. Liu, K. H. Lin, and J. C. Wu, "Analysis of hyperbolic and circular positioning algorithms using stationary signal-strength-difference measurements in wireless communications," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 2, pp. 499–509, 2006.

[17] X. Li, "Collaborative localization with received-signal strength in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3807–3817, 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 397961, 10 pages
doi:10.1155/2012/397961

Research Article

A Multiple-Dimensional Tree Routing Protocol for Multisink Wireless Sensor Networks Based on Ant Colony Optimization

Hui Zhou, Dongliang Qing, Xiaomei Zhang, Honglin Yuan, and Chen Xu

School of Electronics and Information, Nantong University, Jiangsu, Nantong 226019, China

Correspondence should be addressed to Chen Xu, xuchen@ntu.edu.cn

Received 13 January 2012; Revised 1 April 2012; Accepted 15 April 2012

Academic Editor: Shukui Zhang

Routing protocol is an important topic in the wireless sensor networks. For MultiSink wireless sensor networks, the routing protocol designs and implementations are more difficult due to the structure complexity. The paper deals with the problem of a multiple-dimensional tree routing protocol for multisink wireless sensor networks based on ant colony optimization. The proposed protocol is as follows: (1) listening mechanism is used to establish and maintain multidimensional tree routing topology; (2) taking into consideration hops, packet losses, retransmission, and delay account, a distributed ant colony algorithm is proposed. When nodes select routes in the data transmission, the algorithm is utilized to realize the real-time optimization by coordination between nodes. The simulation results show that the proposed protocol can realize the QoS optimization for multisink wireless sensor networks, and its performance is better than the routing protocol of minimum hop numbers.

1. Introduction

Multisink wireless sensor architecture networks have received more and more attention due to their advantages such as improving network throughput, balancing energy consumption, and prolonging network lifetime. Moreover, the reliability and robustness of networks are improved because multisink nodes increase the transmission routines of the sensor node information [1, 2].

Due to multiple sink nodes in multisink wireless sensor networks (multisink WSNs), the network topology is complex, which brings many difficulties to design and implement the network protocols. Currently, research on multisink WSNs is still insufficient, especially for the cooperation and quality of service (QoS) of multisink WSNs.

In this paper, the problem of a multiple dimensional tree routing protocol for multisink WSNs will be investigated based on listening and ant colony optimization (ACO), where multiple dimensional tree routing is defined in Section 3. Listening mechanism is first used to establish and maintain multidimensional tree routing topology in the proposed protocol. Then, a distributed ant colony algorithm is presented with the consideration of hops, packet losses, retransmission and delay.

The rest of this paper is organized as follows. (i) Section 2 states the related works about the multisink WSNs routing researches and the applications of the ant colony optimization in WSNs. (ii) Section 3 describes the multisink WSNs model. (iii) Section 4 introduces the routing establishment of the listening-based routes for the multisink WSNs and the routing selection based on the distributed ant colony optimization. (iv) Simulation experiments are performed and analyzed in Section 5. (v) Some concluding remarks are found in Section 6.

2. Related Works

2.1. Multisink Wireless Sensor Networks. Recently, research results of multisink WSNs routing have been reported in the literature [3–9]. In [3], Dubois-Ferrière et al. limited the sink node transmissions to deliver the query messages to the minimum number of the data collection nodes, using the Voronoi scoping algorithm. In [4], Ciciriello et al. proposed a scheme based on a periodic adaptation of the message routes, which could efficiently route data from multiple sources to multiple sinks. Min [5] proposed priority-based multisink routing protocol, which considered both the level

of node energy and the routing energy, so that the energy consumption was balanced efficiently and the lifetime of the network was prolonged. In [6], Kawano and Miyazaki proposed a minimum multihop routing protocol (MMHR), aiming to minimize the communication hops between each sensor node and a sink node in wireless sensor networks with multiple sink nodes. In [7], the optimal multisink positioning and energy-efficient routing protocol showed that the method to choose the route can be attributed to the linear programming model in order to realize the best deployment of multiple sink nodes and optimize the throughput of the whole network. In [8], Kalantari and Mark took the partial differential equations of Maxwell to resolve the optimization problem of the multisink networks and proposed the partial differential equations protocol. In the opportunistic routing protocol proposed in [9], each node measures the received signal strength indication from sink nodes in order to calculate mobility gradient, information of both the best neighbor node and the best sink nodes. In [10], an efficient multiple sink transmission power control scheme is analyzed for a sink-centric cluster routing protocol in multiple sink wireless sensor networks. It is worth pointing out that some problems such as node coordination, balance of the communication load, and robustness of routing protocol have not been sufficiently investigated in the above works.

2.2. ACO-Based Routing Protocol for WSNs. ACO is a swarm intelligent algorithm which analogs the ant foraging and exchanges pheromones to optimize complex problems [11–13]. Because of the inherent parallelism of ACO, it is appropriate to apply ACO to optimize wireless sensor networks [14]. ACO for single-sink WSNs has been investigated in the last decade [15–18]. In [15], Zhang et al. proposed three new ant-routing algorithms to improve the performance of WSNs. In [16], the energy efficient routing algorithm based on ACO was designed to extend network lifetime by reducing communication overhead in path discovering. It was achieved by energy efficient paths, which were established by using fixed size ant agents and introducing energy and number of hops in pheromone update mechanism. Cai et al. [17] proposed ACO-based QoS routing, which was a reactive protocol that tries to cope with strict delay requirements, limited energy, and computational resources available at sensor nodes. Ant-based service-aware routing algorithm proposed in [18] was a QoS-aware routing protocol for multimedia sensor networks.

ACO for multiple sink WSNs has received attention recently [19, 20]. Kiri et al. [19] described a cluster-based data gathering scheme aimed to achieve reliability and scalability in WSNs. Since WSNs architecture with a single sink is not robust to energy depletion, the authors in [19] proposed a multisink WSNs in which the nodes can use an alternate sink in case of failure of the network. In [20], Paone et al. proposed a routing protocol for multisink WSNs with interesting properties: self organization, fault tolerance, and environmental adaptation, which was inspired by the well known behavior (in artificial life studies) of "slime mold." However, some problems such as QoS and node coordination still need to be fully studied. This motivates the research of this paper.

3. Multisink Wireless Sensor Network Model

In this section, a multisink wireless sensor network model is provided under the following assumptions.

Assumption 1. Sink nodes and sensor nodes are deployed randomly and they cannot move.

Assumption 2. All the sink nodes have the same architecture, and so do the sensor nodes.

Assumption 3. Wireless channels are symmetrical, and the process of receiving and transmitting orientates all directions.

Assumption 4. Each node has its own ID address.

Definition 5 (one-dimensional tree routing). The wireless sensor network tree routing is said to be one-dimensional tree routing if in the WSNs with one sink node and M sensor nodes; routing topology is tree-type structure where the sink node is its root and M sensor nodes are elements.

Definition 6 (N-dimensional tree routing). The wireless sensor network tree routing is said to be N-dimensional tree routing if in the WSNs with N sink nodes and M sensor nodes; routing topology is tree-type structure which is composed of N one-dimensional tree routings T_j ($j = 0, 1, 2 \cdots N-1$) where sensor node n_i ($i = 0, 1, 2 \cdots M-1$) belongs to T_j ($j = 0, 1, 2 \cdots N-1$).

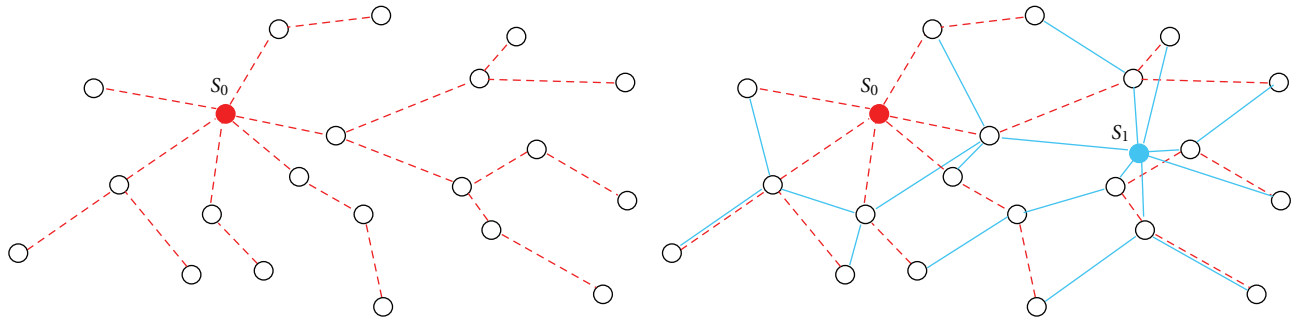
The topology structure of the one-dimensional tree routing and the two-dimensional tree routing is shown in Figures 1(a) and 1(b) separately.

4. ACOMSR Protocol

In this section, a multiple dimensional tree routing protocol for multisink wireless sensor networks based on ant colony optimization (ACOMSR) will be proposed.

4.1. ACOMSR Description. The ACOMSR is mainly made up of two parts: (i) the establishment and maintenance of the multiple dimensional tree routing topology by means of listening, (ii) the route selection and the pheromone update based on the distributed ant colony optimization.

The protocol establishes multidimensional tree topology routing, and the dimensional number is the same of the sink number. Each sensor node establishes N-dimensional routing tables, and every one-dimensional routing table takes sink ID, father node's ID, link quality, load, hop numbers, and other information. Sensor nodes use ant colony optimization algorithm to select routes according to the information of the routing table before they send their data packet.



(a) One-dimensional tree routing (b) Two-dimensional tree routing

Figure 1: The wireless sensor network tree routing.

Destination	Hop	Source	Style	Sink	Load	Time	Other

Figure 2: Frame format of packet.

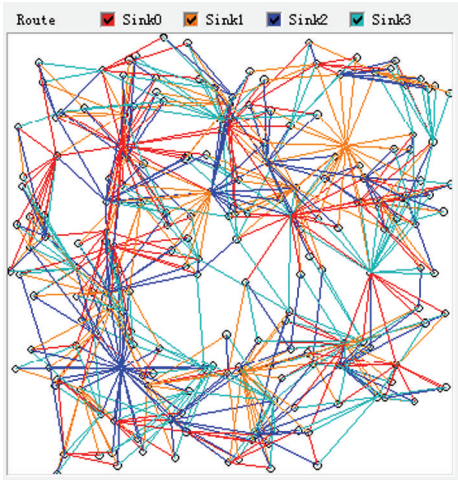


Figure 3: The topology structure of 4-sink network.

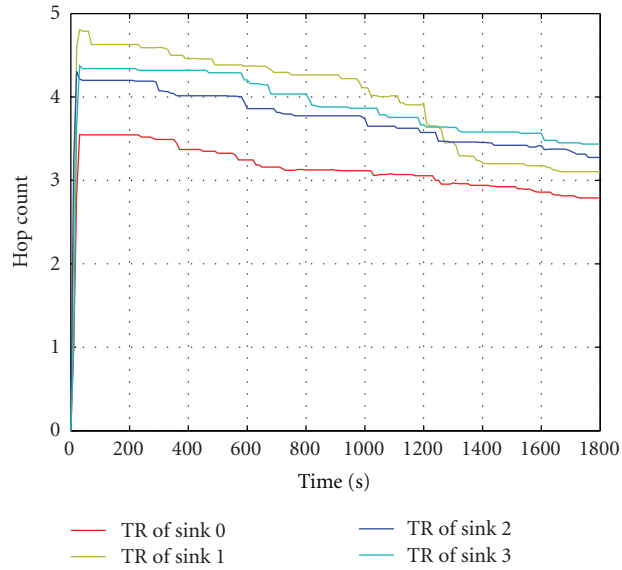


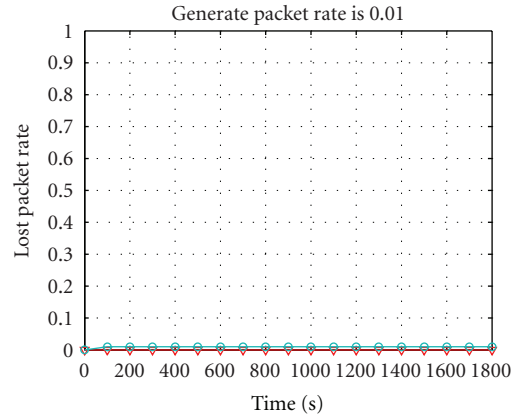
Figure 4: Average hop counts vary for 4-dimensional routing.

Table 1: Notation description.

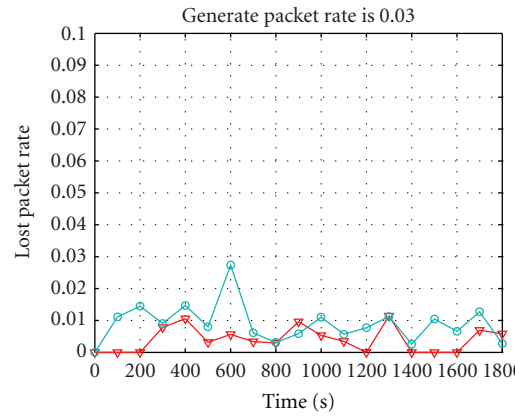
Notation	Description
h_{ik}	Hop number from sensor node i to sink node k.
e_{ik}	Number of packets lost when sensor node i sends data packets to sink node k.
r_{ik}	Times of retransmission when sensor node i sends the data packet to sink node k.
l_{ik}	Total usage of the upstream node buffer in the link from sensor node i to sink node k.
b_{ik}	Average value of the upstream node buffer usage in the link from sensor node i to sink node k.

The corresponding variables are defined in Table 1, and the implementation of the proposed protocol is shown in Algorithm 1.

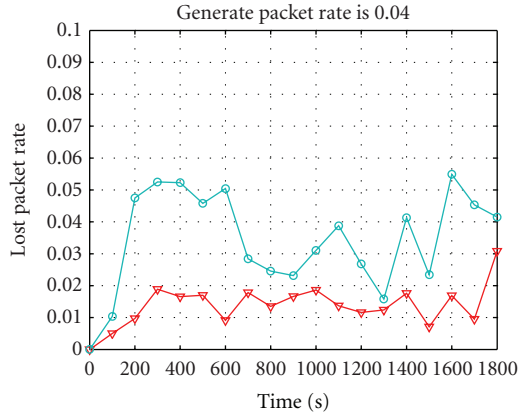
4.2. Establishment and Maintenance of ACOMSR. Listening-based minimum hop routing protocol adopts bottom-to-up approach to establish routes, forming a tree topology structure whose root node is the sink node. During the process of the routing establishment and maintenance, node i broadcasts the routing request packets (RREQ). It is assumed that node j has received the RREQ. If $h_{ik} \leq h_{jk} - 2$, where h_{ik} is hops of node i to sinks k, h_{jk} is hops of node j to sink k, node j will send the routing reply packets (RREP) with broadcast. Then all nodes which are the neighbor nodes of node i could receive RREP. These nodes will establish or update their own routing if they have more hops. Most nodes in the network establish or maintain routes only by listening



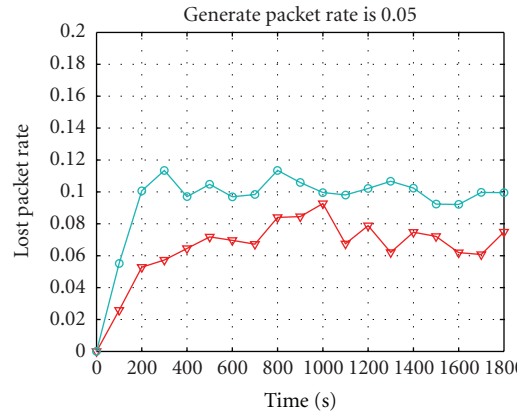
(a) Lost packet rate for the case PGR = 0.01



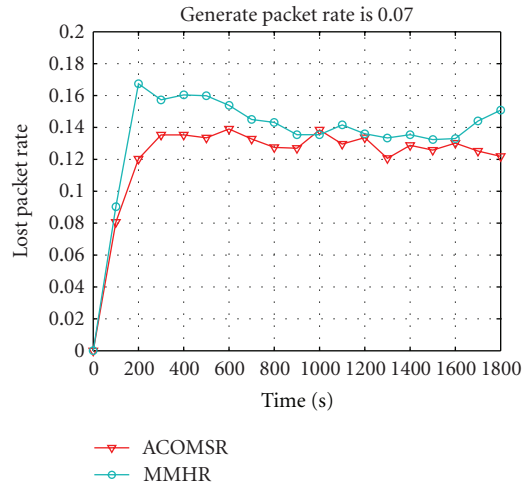
(b) Lost packet rate for the case PGR = 0.03



(c) Lost packet rate for the case PGR = 0.04



(d) Lost packet rate for the case PGR = 0.05



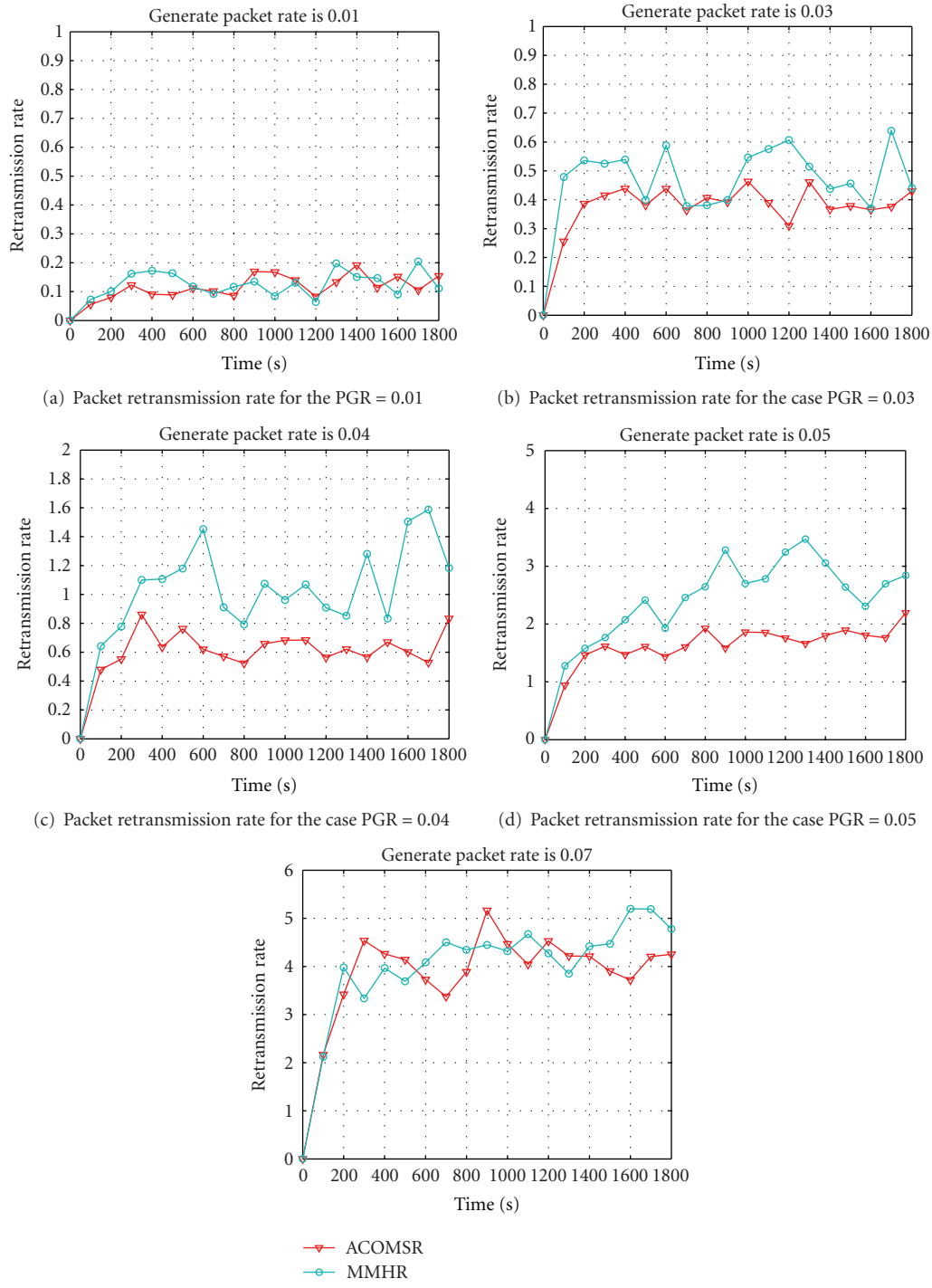
(e) Lost packet rate for the case PGR = 0.07

Figure 5: Lost packet rates at different PGR.

RREP, thus the protocol has the advantage of lower overhead and faster routing. The packet frame is described by Figure 2.

According to the listening mechanism, the process of routing establishment or maintenance can be shown in Algorithm 2.

Remark 7. (i) Since there are N sinks in multiple sink WSNs, N-dimensional tree routing should be established. (ii) In order to avoid collision, nodes use CSMA at the MAC layer when sensor nodes send out packet. (iii) Transmission power control is used to improve the link quality [21]. (iv) Regular



(a) Packet retransmission rate for the PGR = 0.01

(b) Packet retransmission rate for the case PGR = 0.03

(c) Packet retransmission rate for the case PGR = 0.04

(d) Packet retransmission rate for the case PGR = 0.05

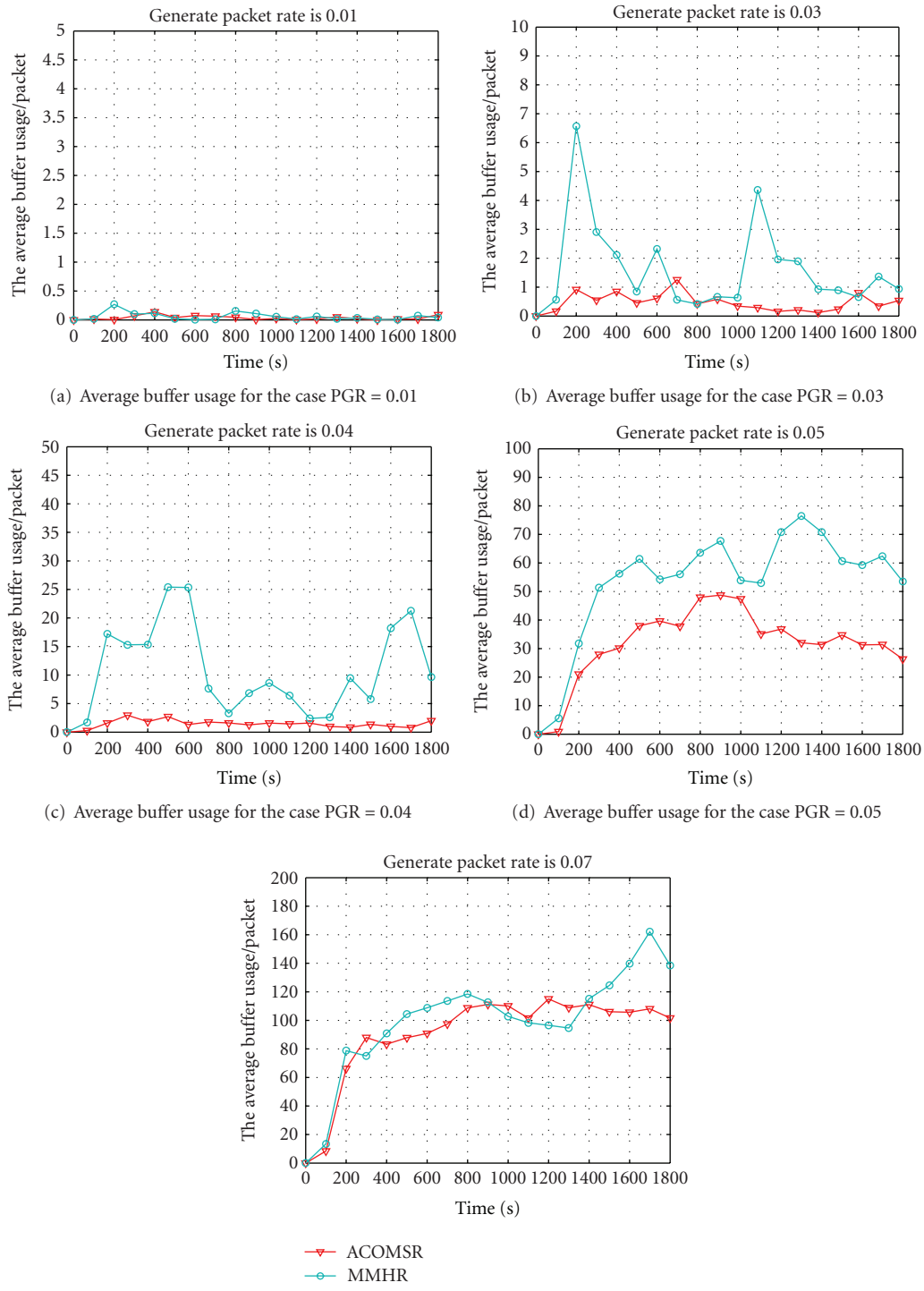
(e) Packet retransmission rate for the case PGR = 0.07

Figure 6: Packet retransmission rates at different PGR.

maintenance is also used in the routing protocol, and the maintenance period is set after the update of the routing messages.

4.3. The Routing Selection and the Pheromone Update Based on the Distributed Ant Colony Optimization. The basic idea of the ant colony optimization in multisink WSNs routing algorithm is to build mappings between routing protocol and ant colony optimization.

Path selection of multisink WSNs is treated as ant foraging, and then a distributed ant colony algorithm is designed. When sensor nodes send packets, the algorithm is used to choose a suitable route. The mapping between



(a) Average buffer usage for the case PGR = 0.01

(b) Average buffer usage for the case PGR = 0.03

(c) Average buffer usage for the case PGR = 0.04

(d) Average buffer usage for the case PGR = 0.05

(e) Average buffer usage for the case PGR = 0.07

Figure 7: Average Buffer Usage at different PGR.

the multisink WSNs routing protocol and the ant colony search space is given in Table 2.

The variable descriptions of the distributed ant colony algorithm are shown in Table 3.

Sensor nodes of multisink WSNs save the quality of links from them to all sink nodes, namely, pheromones when ants look for food sources. Sensor nodes choose routes according to the link quality, that is, ants select foraging path and food

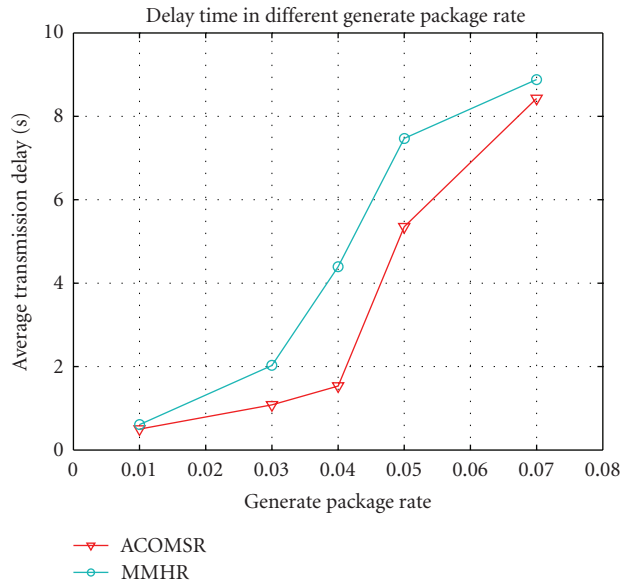


Figure 8: Average transmission delays at the different PGR.

```
Procedure main_algorithm(){
//check the route from the node to sink k.
   If (node. hop==∞) or (the lifetime is over){
      //broadcast the RREQ packet and wait for RREP
Packet
       Initialize the RREQ packet
       routing_algorithm(packet);
   }
   // After the routing is built up
   Initialize the pheromone.
   // When sending data
   path_algorithm();
   if (the data is transmitted successfully) { //receive
the ACK packet
      k = packet.sink;    //the ACK packet
      update_pheromone_algorithm (l_ik, h_ik, r_ik, 0, k);
   }
   else
      update_pheromone_algorithm (0, 0, r_ik, e_ik, k);
   Wait for the next data transmission;
}
```

Algorithm 1: Proposed main algorithm.

```
Procedure routing_algorithm(packet){
// If the node receives RREP packet
   if (receive the RREP){
       k = packet.sink;
       if(k ==Sink_num){
          if(node.hop== ∞) //the node.hop in the
Sink k
                update the route to the Sink k and set
lifetime for the route
             else
                if(node. hop > packet.hop+1)
                  update the rout to the Sink k and set
lifetime for the routing
       }
         else{
            To listening() mode and update the rout to
the sink k
             wait for a short time and then broadcast
RREQ packet again
       }
   else
      wait for a short time and then broadcast RREQ
again
   }
   // If the node receives RREQ packet
   if(receive the RREQ){
      k= packet.sink;
       if(node.hop!= ∞){ // the node.hop in the Sink k
          if(the packet.hop > the node.hop + 1)
             Initialize the RREP packet
       // broadcast the RREP
       }
   }
}
//the listening() mode
if(the neighbor node receives RREP){
   k= packet.sink;
       if(the node.hop > the packet.hop + 1){ // the
node.hop to the Sink k
          update the route to the Sink k and set the
lifetime for the route
   }
}
```

Algorithm 2: Routing establishment and maintenance.

Table 2: Mapping relation between multisink WSNs routing and the ant colony optimization.

Multisink WSNs routing protocol	Ant colony optimization
Sink node	Food source
Route	Foraging path
Data packet	Ant
Link quality	Pheromone concentration
Routing hops	Visibility
Routing selection	Selection of the foraging path

sources based on the pheromone concentration. When ants choose food sources at time t, the transition probability is

$$p_{ik}(t) = \frac{\tau_{ik}(t)\eta_{ik}(t)}{\sum_{k=0}^{K} \tau_{ik}(t)\eta_{ik}(t)}. \quad (1)$$

The pheromone can be presented by

$$\tau_{ik}(t+1) = \rho_{ik}(t) \cdot \tau_{ik}(0) + \Delta\tau_{ik}(t), \quad (2)$$

where $\tau_{ik}(0)$ is the initial pheromone.

Table 3: Notation description of ACO.

Notation	Description
$p_{ik}(t)$	The transition probability at time t when packets of sensor node i select the link between sensor node i and sink node k.
$\tau_{ik}(t)$	Pheromone of the route between sensor node i and sink node k at time t.
$\eta_{ik}(t)$	Visibility from sensor node i to sink node k at time t.
$\Delta\tau_{ik}(t)$	Pheromone update value from sensor node i to sink node k at time t.
$\rho_{ik}(t)$	Routing pheromone volatile coefficient from sensor node i to sink node k at time t.

Define the visibility as the reciprocal of the hop number from sensor node i to sink node k, which is given by

$$\eta_{ik}(t) = \frac{1}{h_{ik}(t)}. \tag{3}$$

The initial pheromone from sensor node i to sink node k is given by

$$\tau_{ik}(t) = \gamma^{h_{ik}(t)}, \tag{4}$$

where γ represents the initial pheromone when $h_{ik}(t) = 1$.

Define the pheromone update value associated with the link quality as follows:

$$\Delta\tau_{ik}(t) = \alpha \cdot \frac{1}{1 + (e_{ik}(t))^a + (r_{ik}(t))^b}, \tag{5}$$

where α is the weight of $\Delta\tau_{ik}(t)$, a and b represent the influence size on $\Delta\tau_{ik}(t)$ for packet loss and retransmission, respectively.

In order to avoid the pheromone unlimited accumulation which caused the imbalance evaluation of the link quality, we define the pheromone volatile coefficient as follows

$$\rho_{ik}(t) = \rho_0 \cdot \beta^{(b_{ik}(t)-\mathrm{th})/(m-\mathrm{th})}, \quad (0 < \rho_{ik}(t) < 1), \tag{6}$$

where ρ_0 is the initial value of the pheromone volatile coefficient, β is the basic number of the volatile coefficient, m is the maximum value of the buffer in a sensor node, th is a threshold of the average link load from sensor node i to sink node k, and $b_{ik}(t)$ denotes the average link load from sensor node i to sink node k, which is defined by

$$b_{ik}(t) = \frac{l_{ik}(t)}{h_{ik}(t)}. \tag{7}$$

The implementation procedure of the distributed ant colony algorithm is displayed in Algorithm 3.

5. Performance Evaluation

In this section, a simulation example is illustrated to show the effectiveness of the proposed protocol. Moreover, results compared with the minimum multihop multisink routing protocol (MMHR) are also provided.

Table 4: Network simulation parameters.

Parameters	Value
Sensor nodes	200
Sink nodes	4
Simulation area	7000 (m) × 7000 (m)
Channel model	Free-space model
Frequency band	433 MHZ
Power of DATA/ACK	0 dbm
Power of RREQ/RREP	−3 dbm
Data rate	20 Kbps
Channel bandwidth	0.2 MHZ
Simulation time	1800 s
Simulation step	0.0004 s
Packet size	48 Bytes
Length of ACK packet	8 Bytes
Size of node buffer	960 Bytes

Table 5: Parameters of the distributed ACO.

Parameters	Value
γ	0.01
α	0.001
a	4
b	2
ρ_0	0.9
β	0.7
n	2

5.1. Simulation Environment. Our simulation environment consists of 200 sensor nodes and 4 sink nodes randomly deployed in a field of 7000 m ∗ 7000 m. Table 4 shows the simulation parameters of the network. The parameters of distributed ant colony algorithm are shown in Table 5 which are selected by means of a lot of simulation.

We can analyze the performance of this proposed protocol by changing generate packet rate. Here, generate packet rate means the amount of packet the perception nodes produced in unit working time.

5.2. Simulation Results and Analysis. According to routing establishment and maintenance process given in Algorithm 2, the ACOMSR routing topology structure at the moment of 60 s simulation time shown by Figure 3 is obtained, where red represents the routing to sink 0, orange the routing to sink 1, blue the routing to sink 2, and green the routing to sink 3.

As displayed in Figure 4, when the packet generate rate (PGR) is 0.04, the average hop numbers differ from each other with the change of time in the 4-demensional routing. We can see from Figure 4 that the average hop number decreases step by step with the increase of time. This is the routing continuous optimization result from the minimum routing maintenance mechanism.

```
Procedure path_algorithm (h_ik, τ_ik){
    calculate the transition probability p_ik(t) for each
path by the formula (1);
    random=Rnd();
  if(p_i(k−1) < random < p_ik)
    choose the path to the sink k
}
Procedure update pheromone algorithm (h_ik, l_ik, r_ik, e_ik, k)
{//the node receives the ACK successfully
if(h_ik!=node.hop){ // the node.hop in the Sink k
  τ_ik = γ^h_ik; node.hop = h_ik;
}
  b_ik = l_ik/h_ik;
if(b_ik > th)
  calculate the volatilization coefficient ρ_ik by the
formula (6);
Else
  ρ_ik = ρ_0;
  calculate the correction value Δτ_ik by the formula (5);
update the pheromone τ_ik by the formula (2);
}
```

Algorithm 3: Distributed ant colony algorithm.

For the proposed protocol and MMHR [6], lost packet rates, packet retransmission rates, and average transmission delays at different PGR are shown by Figures 5, 6, and 7, respectively.

From Figure 5 it can be seen that ACOMSR outperforms MMHR in term of the packet loss rate. When the packet generate rate is 0.01, the difference between ACOMSR and MMHR is not obvious. When the packet generate rate is 0.03, 0.04, 0.05, or 0.07, the proposed protocol shows a better performance in reducing packet loss rate.

From Figure 6 it can be seen that ACOMSR outperforms MMHR in term of the packet retransmission rate. But the differences of the packet retransmission rate between the two protocols are not obvious when the packet generate rate is 0.01 and 0.07.

From Figure 7 it can be seen that ACOMSR outperforms MMHR in the average buffer usage of all sensor nodes. When the packet generate rate is 0.01, the difference between ACOMSR and MMHR is basically the same. When the packet generate rate is 0.03, 0.04, 0.05 or 0.07, the proposed protocol shows a better performance in reducing average buffer usage.

From Figure 8, it is easy to see that the average transmission delays obtained by using ACOMSR are less than those obtained by using MMHR.

The simulation results show that the proposed protocol ACOMSR has better performances than MMHR and the QoS optimization of multisink WSNs is achieved. The main reasons are two folds: (i) for selecting routing, MMHR takes the hop as the only performance index of selecting routes, whereas both hop numbers and QoS (including load balance, packet loss, and retransmission) in our protocol are simultaneously considered. The cooperation on the link information feedback and the nodes is achieved because of the interaction of the node link status information in DATA-ACK form. (ii) The introduction of the distributed ant colony algorithm makes the proposed routing protocol intelligent, and thus, a reasonable balance of network load is possible. Especially when the network load is heavy, the proposed protocol shows much better performance. It is worth pointing out that the real-time nature of the network transmission has been improved although the packet transmission route of the proposed protocol (ACOMSR) is not the shortest.

However, because of ACO algorithm, the computational overhead of the proposed protocol is more than the MMHR. With the case of 4 Sinks, each time the computation overhead of ACOMSR in a node is much more than that of MMHR's about 21 times multiplications. But sensor node can be perfectly qualified for this overhead. Taking the microprocessor LPC2131 of Philips Corp as an example, the microprocessor comes with a hardware multiplier so that it only costs 21 instruction cycles, which is about 2 ms to complete these multiplications. Therefore, the proposed protocol is feasible in actual application.

6. Conclusion

A multiple dimensional tree routing protocol for multisink WSNs based on listening and ant colony optimization has been proposed in this paper. The advantages of the proposed protocol can be summarized as follows. (i) In the process of the routing establishment and maintenance, the waste of resources is avoided and the reliability of routing is improved by utilizing the listening mechanism and the power control, respectively. (ii) The fault tolerance and robustness of routing are increased because multidimensional tree routes from each sensor node to all sink nodes are set up. (iii) The QoS optimization of multisink WSNs is achieved by using the proposed ACOMSR. Simulation experiments have

been made to show that the performance of the proposed ACOMSR is better than the routing protocol of minimum hop numbers. As a future work, we are intended to study the cross-layer optimization and multiobjective optimization for the multisink wireless sensor networks based on the ant colony algorithm.

Acknowledgment

This work was supported by National Natural Science Foundation of China under Grants nos. 61174065, 61071086, and 60901041.

References

[1] M. Ayaz, I. Baig, A. Abdullah, and I. Faye, "A survey on routing techniques in underwater wireless sensor networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1908–1927, 2011.

[2] Y. Yang, M. I. Fonoage, and M. Cardei, "Improving network lifetime with mobile wireless sensor networks," *Computer Communications*, vol. 33, no. 4, pp. 409–419, 2010.

[3] H. Dubois-Ferrière, E. Deborah, and S. Thanos, "Efficient and practical query scoping in sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 564–566, Lauderdale, Fla, USA, October 2004.

[4] P. Ciciriello, M. LucaPicco, and G. Pietro, "Efficient routing from multiple sources to multiple sinks in wireless sensor networks," in *Proceedings of the 4th European Conference on Wireless Sensor Networks (EWSN '07)*, pp. 34–50, Delft, The Netherlands, January 2007.

[5] M. Min, "PBR: priority based routing in multi-sink sensor networks," in *Proceedings of the Conference on Wireless Sensor Networks*, pp. 25–28, Las Vegas, Nev, USA, 2007.

[6] R. Kawano and T. Miyazaki, "Distributed data aggregation in multi-sink sensor networks using a graph coloring algorithm," in *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications Workshops/Symposia (AINA '08)*, pp. 934–940, March 2008.

[7] H. Kim, Y. Seok, N. Choi, Y. Choi, and T. Kwon, "Optimal multi-sink positioning and energy-efficient routing in wireless sensor networks," in *Proceedings of the International Conference on Information Networking (ICOIN '05)*, pp. 264–274, Jeju Island, Korea, February 2005.

[8] M. Kalantari and S. Mark, "Design optimization of multi-sink sensor networks by analogy to electrostatic theory," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '06)*, pp. 431–438, Las Vegas, Nev, USA, April 2006.

[9] A. Lukosius, *Opportunistic Routing in Multi-Sink Mobile Ad Hoc Wireless Sensor Networks*, University of Bremen, Bremen, Germany, 2007.

[10] L. Cao, C. Xu, W. Shao et al., "Distributed power allocation for sink-centric clusters in multiple sink wireless sensor networks," *Sensors*, vol. 10, no. 3, pp. 2003–2026, 2010.

[11] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 26, no. 1, pp. 29–41, 1996.

[12] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, UK, 2004.

[13] N. Jiang, R. G. Zhou, and S. Q. Yang, "An improved ant colony broadcasting algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 1, pp. 45–45, 2009.

[14] M. Saleem, G. A. Di Caro, and M. Farooq, "Swarm intelligence based routing protocol for wireless sensor networks: survey and future directions," *Information Sciences*, vol. 181, no. 20, pp. 4597–4624, 2011.

[15] Y. Zhang, L. Kuhn, and M. Fromherz, "Improvements on ant routing for sensor networks," in *Proceedings of the 4th International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS '04)*, pp. 154–165, Brussels, Belgium, September 2004.

[16] T. Camilo, C. Carreto, J. S. Silva, and F. Boavida, "An energy-efficient ant-based routing algorithm for Wireless Sensor Networks," in *Proceedings of the 5th International Workshop—Ant Colony Optimization and Swarm Intelligence (ANTS '06)*, pp. 49–59, Brussels, Belgium, September 2006.

[17] W. Cai, X. Jin, Y. Zhang, K. Chen, and R. Wang, "ACO based QoS routing algorithm for wireless sensor networks," in *Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing (UIC '06)*, pp. 419–428, Wuhan, China, September 2006.

[18] Y. Sun, H. D. Ma, and L. Liu, "Ant-colony optimization based service aware routing algorithm for multimedia sensor networks," *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 35, no. 4, pp. 705–711, 2007.

[19] Y. Kiri, M. Sugano, and M. Murata, "Self-organized data-gathering scheme for multi-sink sensor networks inspired by swarm intelligence," in *Proceedings of the 1st International Conference on Self-Adaptive and Self-Organizing Systems (SASO '07)*, pp. 161–170, Cambridge, Mass, USA, July 2007.

[20] M. Paone, L. Paladina, M. Scarpa, and A. Puliafito, "A multi-sink swarm-based routing protocol for wireless sensor networks," in *Proceedings of the IEEE Symposium on Computers and Communications 2009 (ISCC '09)*, pp. 28–33, Sousse, Tunisia, July 2009.

[21] S. L. Zhu, C. Xu, Q. Sun, and X. Huang, "A novel transmission power selection mechanism for wireless sensor networks," in *Proceedings of the International Conference on Electronics, Communications and Control (ICECC '11)*, pp. 661–665, Ningbo, China, September 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 790459, 11 pages
doi:10.1155/2012/790459

Research Article

Towards Aid by Generate and Solve Methodology: Application in the Problem of Coverage and Connectivity in Wireless Sensor Networks

Placido Rogerio Pinheiro, Andre Luis Vasconcelos Coelho, Alexei Barbosa Aguiar, and Alvaro de Menezes Sobreira Neto

Graduate Program in Applied Informatics, University of Fortaleza-UNIFOR, Avenue Washington Soares 1321, 60811-905 Fortaleza, CE, Brazil

Correspondence should be addressed to Placido Rogerio Pinheiro, placidrp@uol.com.br

Received 23 July 2012; Accepted 10 September 2012

Academic Editor: Shan Lin

The integrative collaboration of genetic algorithms and integer linear programming as specified by the Generate and Solve methodology tries to merge their strong points and has offered significant results when applied to wireless sensor networks domains. The Generate and Solve (GS) methodology is a hybrid approach that combines a metaheuristics component with an exact solver. GS has been recently introduced into the literature in order to solve the problem of dynamic coverage and connectivity in wireless sensor networks, showing promising results. The GS framework includes a metaheuristics engine (e.g., a genetic algorithm) that works as a generator of reduced instances of the original optimization problem, which are, in turn, formulated as mathematical programming problems and solved by an integer programming solver.

1. Introduction

High power consumption efficiency in wireless sensor networks is always desirable. The integrative collaboration of genetic algorithms and integer linear programming as specified by this methodology tries to merge their strong points and has offered significant results when applied to wireless sensor networks domains. However, its original implementation showed some deficiencies which limits its performance, being one of them the density explosion. Besides, it was conceived exclusively for this problem domain, the methodology could serve as a start point for using with other domains, since the genetic algorithm, implemented in this approach, is flexible and can be applied to many other problems. The correct use of a hybrid methodology to deal with a class of problem can show good results. The methodology mainly consists in reducing a problem, typically a high-complexity problem, that the time to solve it is infeasibly big, into subproblems. These subproblems maintain the original problem's basic characteristics and the time spent to find its solution is very small, allowing finding a feasible original problem's solution through the solution of many subproblems. So, the methodology consists in generating the subproblems, then solving them, and evaluating continuously, until a stop condition is satisfied. Genetic algorithm was chosen to be the subproblems generator. A genetic algorithm characteristic is the evolution, so it can evaluate and conduct the evolution of the subproblems to come closer to the original problem's optimal solution. A dynamic coverage and connectivity for wireless sensor network problem was adopted in this work [1–3]. In Section 2 is given an outline about wireless sensor network, with all the relevant features for this work. In Section 3 is shown the mathematical model implemented in this work. More details of the hybrid methodology are given in Section 4. Section 5 gives an improvement for the density explosion problem that could derail the whole approach. Section 6 presents the computational results obtained using the methodology described. Finally, in Section 7 is presented some conclusion obtained from this work.

2. The Wireless Sensor Network

A Wireless Sensor Network (WSN) typically consists of a large number of small, low-power, and limited-bandwidth computational devices, named sensor nodes. These nodes can frequently interact with each other, in a wireless manner, in order to relay the sensed data towards one or more processing machines (a.k.a. sinks) residing outside the network [4]. For such a purpose, special devices, called gateways, are also employed in order to interface the WSN with a wired, transport network. To avoid bottleneck and reliability problems, it is pertinent to make one or more of these gateways available in the same network setting, a strategy that can also reduce the length of the traffic routes across the network and consequently lower the overall current consumption. A typical sensor node is composed of four modules, namely, the processing module, the battery, the transceiver module, and the sensor module as described in [5]. Besides the packet building processing, a dynamic routing algorithm runs over the sensor nodes, in order to discover and configure in runtime, the "best" network topology in terms of transmission number and current waste. Due to the limited resources available to the microprocessor, most devices make use of a small operating system that supplies basic features to the application program. To supply the power necessary to the whole unit, there is a battery, whose lifetime duration depends on several aspects, among which its storage capacity and the levels of electrical current employed in the device. The transceiver module, conversely, is a device that transmits and receives data using radio-frequency propagation as media and typically involves two circuits, namely, the transmitter and the receiver. Due to the use of public-frequency bands, other devices in the neighborhood can cause interference during sensor communication [6]. Likewise, the operation/interaction among other sensor nodes of the same network can cause this sort of interference. So, the lower is the number of active sensors in the network, the more reliable tends to be the radio-frequency communication among these sensors. The last component, the sensor module, is responsible to gauge the phenomena of interest; the ability of concurrently collecting data pertaining to different phenomena is a property already available in some models of sensor nodes.

For each application scenario, the network designer has to consider the rate of variation for each sensed phenomenon in order to choose the best sampling rate of each sensor device. Such decision is very important to be pursued with precision as it surely has a great impact on the amount of data to be sensed and delivered, and, consequently, on the levels of current consumed prematurely by the sensor nodes [7]. This is the temporal aspect to be considered in the network design.

Another aspect to be considered is the spatial one. On the other hand, [8] define coverage as a measure of the ability to detect objects within a sensor field. The lower the variation of the physical variable being measured across the area, the shorter has to be the radius of coverage for each sensor while measuring the phenomenon. This will have an influence on the number of active sensors to be employed to cover all demand points related to the given phenomenon. The fact is the more sensors are active in a given moment, the bigger is the overall current consumed across the net. WSNs are usually deployed in hostile environments, with many restrictions of access. In such cases, the network would be very unreliable and unstable if the minimum number of sensor nodes was effectively used to cover the whole area of observation. If some sensor node fails to operate, its area of coverage would be out of monitoring, preventing the correlation of data coming from this area with others coming from other areas.

Another worst-case scenario occurs when we have sensor nodes as network bottlenecks, being responsible for routing all data coming from the sensor nodes in the neighborhood. In this case, a failure in such nodes could jeopardize the whole network deployment. To avoid these problems and make a robust design of the WSN, extra sensor nodes are usually employed in order to introduce some sort of redundancy. By this means, the routing topology needs to be dynamic and adaptive: when a sensor node that is routing data from other nodes fails, the routing algorithm discovers all its neighbor nodes and then the network reconfigures its own topology dynamically. One problem with this approach is that it entails unnecessary current consumption. This is because the coverage areas of the redundant sensor nodes overlap too much, giving birth to redundant data. And these redundant data bring about extra current consumption in retransmission nodes. The radio-frequency interference is also stronger, which can cause unnecessary retransmissions of data, increasing the levels of current expenditure. On the other hand, [9] present many integer linear programming models to minimize current consumption but do not consider the dynamic time scheduling.

3. Model

The solution proposed by [10, 11] is to create different schedules, each one associated with a given time interval, that activate only the minimum set of sensor nodes necessary to satisfy the coverage and connectivity constraints. The employment of different schedules prevents the premature starvation from some of the nodes, bringing about a more homogeneous level of consumption of battery across the whole network. This is because the alternation of active nodes among the schedules is often an outcome of the model, as it optimizes the current consumption of the whole network taking into account all time intervals and coverage and connectivity constraints.

In order to properly model the WSN setting, some previous remarks are necessary.

(1) A demand point is a geographical point in the region of monitoring where one or more phenomena are sensed. The distribution of such points across the area of monitoring can be regular, like a grid, but can also be random in nature. The density of such points varies according to the spatial variation of the phenomenon under observation. At least one sensor must be active in a given moment to sense each demand point. Such constraint is implemented in the model.

(2) Usually, the sensors are associated with coverage areas that cannot be estimated with accuracy. To

simplify the modeling, we assume plain areas without obstacles. Moreover, we assume a circular coverage area with a radius determined by the spatial variation of the sensed phenomenon. Within this area, it is assumed that all demand points can be sensed. The radio-frequency propagation in real WSNs is also irregular in nature. In the same way, we can assume a circular communication area. The radius of this circle is the maximum distance at which two sensor nodes can interact.

(3) A route is a path from one sensor node to a sink possibly passing through one or more other sensor nodes by retransmission. Gateways are regarded as special sensor nodes whose role is only to interface with the sinks. Each phenomenon sensed in a node has its data associated with a route leading to a given sink, which is independent of the routes followed by the data related to other phenomena sensed in the same sensor node.

(4) The electric charge consumption is actually the electric current drawn by a circuit in a given time period.

These are the decision variables:

$x_{ij}^t \in \mathbf{R}^+$: if sensor $i \in S$ covers demand point $j \in D$ in period $t \in T$;

$w_i^t \in \mathbf{R}^+$: if sensor i was activated in period t for at least one phenomenon;

$z_{lij}^t \in \{0,1\}$: If arc ij belongs to the route from sensor $l \in S$ to a sink in period $t \in T$;

$y_i^t \in \{0,1\}$: if sensor i is activated in period t;

$h_j^t \in \{0,1\}$: if demand point j is not covered by any sensor in period t;

$e_i \in \mathbf{R}^+$: energy consumed by sensor i considering all time periods.

The objective function (1) minimizes the total electrical charge consumption through all time periods. The second term penalizes the existence of some uncovered demand points, but the solution continues feasible. It penalizes unnecessary activation for the phenomenon too:

$$\min \sum_{i \in S} e_i + \sum_{t \in T} \sum_{j \in D} EH h_j^t. \tag{1}$$

These are the constraints adopted:

$$\sum_{i \in S} \sum_{j \in D} AD_{ij} x_{ij}^t + h_j^t \geq 1, \quad \forall j \in D, \ \forall t \in T. \tag{2}$$

Constraint (2) enforces the activation of at least one sensor node i to cover the demand point j in period t. Otherwise, the penalty variable h is set to one. This last condition will occur only in those cases when no sensor node can cover the demand point.

$$x_{ij}^t \leq y_i^t, \quad \forall i \in S, \ \forall j \in S, \ \forall t \in T. \tag{3}$$

The y (which means that a sensor node is actively sensing in period t) is turned on, if it is associated sensor node is indeed allocated to cover any demand point. Constraint (3) indicates to the model that maintenance energy is being consumed, so the model could attribute the corresponding value for this consumption indicated in another constraint described later on

$$\sum_{i \in (S - \{j\})} A_{ij} z_{lij}^t - \sum_{k \in (S \cup M - \{j\})} A_{jk} z_{ljk}^t = 0, \quad \forall j \in S, \ \forall l \in S, \ \forall t \in T. \tag{4}$$

According to the flow conservation principle applied to the connectivity issue, if there is an incoming route to a sensor node, there should be an outgoing route from this same sensor node. So, constraint (4) enforces this statement, setting an outgoing route from sensor node j to sensor node k if there is already an incoming route from sensor node i to sensor node j

$$\sum_{k \in (S \cup M - \{l\})} A_{ij} z_{ljk}^t = y_l^t, \quad \forall l \in S, \ \forall t \in T. \tag{5}$$

If there is an active sensor node, a route must be created. That is what constraint (5) above enforces.

$$\sum_{i \in S} \sum_{j \in M} A_{ij} z_{lij}^t = y_l^t, \quad \forall t \in T, \ \forall l \in S. \tag{6}$$

Constraint (6) is necessary to create a route that reaches a sink if a sensor is active.

$$A_{ij} z_{lij}^t \leq y_j^t, \quad \forall j \in S, \ \forall l \in (S - \{j\}) \quad \forall i \in (S - \{j\}), \ \forall t \in T. \tag{7}$$

All data sensed must reach a sink node, but many sensors node have no direct connectivity to a sink node. So, other sensor nodes might be activated just to turn viably the route to the sink. In constraint (7), if there is an outgoing route passing through sensor node i, then this sensor node has to be active

$$A_{ij} z_{lij}^t \leq y_i^t, \quad \forall j \in S, \forall l \in (S - \{j\}) \quad \forall i \in (S - \{j\}), \forall t \in T. \tag{8}$$

In the same way, with constraint (8) if there is an incoming route passing through sensor i, then this sensor has to be active

$$\sum_{t \in T} EM_i y_i^t + EA_i w_i^t + \sum_{l \in (S - \{i\})} \sum_{k \in S} ER_i z_{lki}^t + \sum_{l \in S} \sum_{j \in (S \cup M)} ET_{ij} z_{lij}^t \leq e_i, \quad \forall i \in S. \tag{9}$$

The total electrical charge consumed by a sensor node is the sum of the parcels given constraint (9). The maintenance energy is attributed when the sensor is active for any reason. The activation energy is summed only when there was an

effective activation through time intervals. The w variable values are given in other restrictions. The reception and transmission energy are given when there are incoming and outgoing routes, respectively, passing from a sensor node. The sum of these terms has to be equal or less than the battery's energy

$$0 \leq e_i \leq EB_i, \quad \forall i \in S. \tag{10}$$

Constraint (10) enforces that each sensor node should consume at most the capacity limit of its battery

$$w_i^0 - y_i^0 \geq 0, \quad \forall i \in S, \tag{11}$$

If a sensor is active in the first time interval, it means that it consumed energy to activate. The w variable indicates this activation. Then, the variable's value is set to 1. On the other hand, if the sensor is kept off in the first time interval, the value is set to 0. Constraint (11) ensures that

$$w_i^t - y_i^t + y_i^{t-1} \geq 0, \quad \forall i \in S,\ \forall t \in T,\ t > 0. \tag{12}$$

In Constraint (12), the sensor's past and current activation states are compared. If the sensor node was active from period $t-1$ to period t, then w is set to 1, 0 otherwise.

4. The Base Hybrid Methodology

Although distinct, both the exact and metaheuristics approaches have pros and cons when dealing with hard combinatorial optimization problems. But their hybridization, when properly done, may allow the merging of their strong points in a complementary manner. For instance, it is well known that the direct application of exact methods is only possible for limited-sized instances. However, the size and complexity of the optimization problems faced nowadays has increased a lot, demanding for the development of new methods and solutions that can find acceptable results within a reasonable amount of time.

In [12–14], WSN problems were explored regarding the heterogeneity of the phenomena. This model however suffers a shortage of variables due to the increase of complexity as many matrices had gained one more dimension.

In this regard, it has become ever more evident that a skilled combination of concepts stemming from different metaheuristics can be a very promising strategy that one should resort to when having to deal with complicated optimization tasks. The hybridization of metaheuristics with other operations research techniques has been shown great appeal as well, as they typically represent complementary perspectives over the problem solving process as a whole. In general, the label of "hybrid metaheuristics" has referred to combinations of components coming from different metaheuristics and from more conventional exact methods into a unique optimization framework by [15–17].

In this context, a hybrid methodology has been recently introduced in the literature by [14, 18–23], trying to push forward the boundaries that limit the application of an exact method through the decomposition of the original problem into two conceptual levels. According to the framework underlying this approximate methodology (see Figure 1), the exact method (encapsulated in the solver of reduced instances (SRIs) component) works no more with the original problem, but with reduced instances (i.e., subproblems) of it that still preserve its conceptual structure. By this means, an optimal solution to a given subproblem will also be a feasible solution to the original problem. On the other hand, the metaheuristics component of the framework works on a complementary optimization problem, that is, the design of reduced instances of the original problem formulated as mathematical programming (namely, integer linear programming (ILP) models). It is referred to as the generator of reduced instances (GRIs), whose goal is to determine the subset of points of the reducible structure that could derive the best subproblem instance, that is, the subproblem which, when submitted to the SRI, would bring about the feasible solution with the highest possible objective function value. In this scenario, the objective function values of the solutions that could be realized by the solver are used as figure of merit (fitness) of their associated subproblems, thus guiding the metaheuristics search process. The interaction between GRI and SRI is iterative and repeats until a given stopping condition is satisfied.

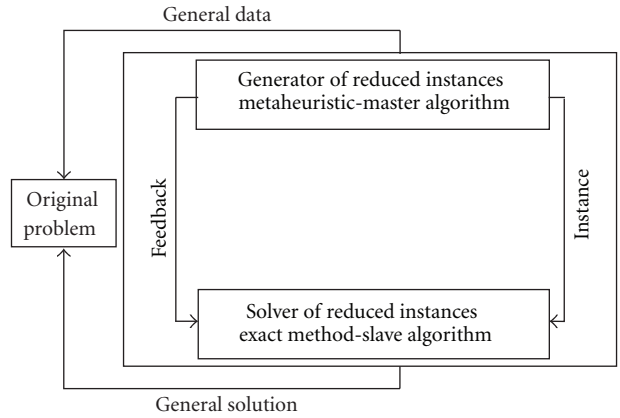


Figure 1: The hybrid framework under investigation.

So far, the metaheuristics chosen to implement the generator of reduced instances has been a genetic algorithm (GA) as explained by [24]. This option is due mainly to the good levels of flexibility and adaptability exhibited by the class of evolutionary algorithms when dealing with a wide range of optimization problems as presented by [25]. The genetic representation of the individuals (chromosomes) follows a binary encoding that indicates which decision variables belonging to the reducible structure will be kept in the new subproblem to be generated. That is, those genes having "1" as alleles define the subset of variables that generates the reduced instance. Conversely, the exact method is assumed to be any state-of-the-art algorithm used to solve mixed integer-linear problems, such as Branch-and-bound or Branch-and-cut described in [26]. Usually, the solver libraries available incorporate sets of strategies, heuristics, and problem reduction techniques that complement the main exact method and enhance its performance.

According to the classification proposed in [27], the methodology falls into the category of integrative combinations. The quality of the solutions to the instances generated by the metaheuristic is determined when the subproblems are solved by the exact method, and the best solution obtained throughout the whole metaheuristic process is deemed to be the final solution to the original problem.

Although showing remarkable levels of performance for some case problems studied in the realm of cutting and packing problems in [14, 18–21], the original version of the aforementioned hybrid methodology has drawbacks, some of which are circumvented with the adoption of the mechanisms discussed here. Other impacting factor that must be noticed is that the original version addressed only the cutting and packing problem class. One consequence of this particularity is that it requires some changes in order to be adapted to new optimization problem classes, described as follows in Section 5.

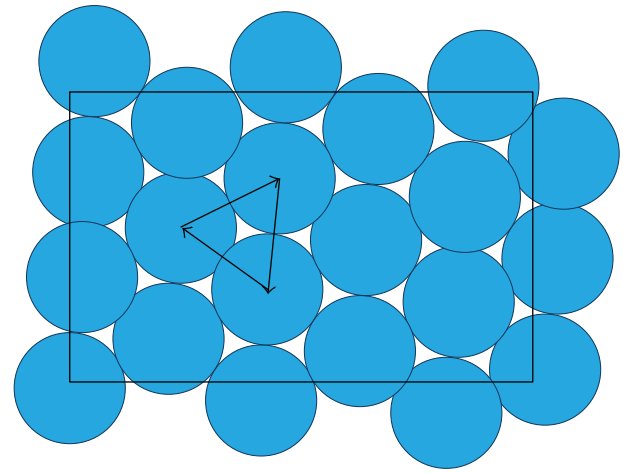


Figure 2: Maximizing coverage.

5. Improvements for the Dynamic Coverage and Connectivity in Wireless Sensor Network Problem

Adopting the base hybrid methodology to be suitable for a totally different class of problem is a challenge. Even the direction of optimization is opposite and requires changing since genetic algorithms natively maximize, while this problem is a minimization one. Although this issue is easy to solve, it shows how distinct problem classes can be even right in the beginning. A drawback that has limited the effectiveness of the base hybrid methodology as presented in Section 3 relates to its propensity for bringing about an uncontrolled density explosion over the individuals (i.e., reduced instances of the original problem) produced by the GRI. We define "density of an individual" as the ratio between the number of genes having "1" as alleles (referred to as activated) and its total length. The fact is that an increase in density tends to generate subproblems more closer to the original problem, thus possibly yielding better solutions. This situation can be better pictured as if having some sort of an "attractor" pushing the overall population density up as the GRI (GA) evolves. Although expected, this phenomenon may have an undesirable side effect if it occurs prematurely. This is because, usually, high densities imply higher complexity to be dealt with by the SRI, which indirectly affects the search process conducted by the GRI as the time spent in each generation tends to become progressively higher. This may cause a drastic limitation to the number of search iterations performed by the SRI, hindering both the effectiveness and efficiency of the whole optimization.

Other undesirable characteristic of the original version of this methodology is that its binary chromosome encoding can be prohibitively long, depending on the chosen reducible matrix. Long chromosomes can lead to problems.

5.1. Spread Sensor Heuristic in Population Initialization. The first initial population generation strategy is a randomized one. Due to the evolutionary nature of genetic algorithm (GA), the start point should not necessarily be a good one. And this was the case.

However, there is a trend that sensor nodes that are selected to participate in a subproblem are too close to each other by comparing their coverage radii. Overlapping of coverage is desirable until a certain point because it provides diversity of choices and network reliability. But keeping sensor nodes too close can lead to a concentration of coverage in the most central part of the observation area and possibly leaving some peripheral areas uncovered. The configuration that maximizes the coverage area ratio (covered area divided by total area) with a minimum number of sensors is that where the distances between two neighbor sensor nodes are equal to two times the coverage radius (Figure 2).

The key parameter to provide good balance between reliability and the number of sensor nodes is the density. This density can be determined by the network designer based upon the requirements of the specific application. But due to irregular activation and deactivation of sensor nodes, this density tends to be not so homogeneous along the observation area. As a side effect, this model tries to distribute this density more homogeneously because this is the configuration where the uncovered points penalize less and the number of active sensors spending energy is reduced, minimizing the objective function.

The idea of the spread sensors heuristic is to provide a good starting point to the GA in order to reduce the time spent in the initial phase of the evolution process. It is based on the selection of sensor nodes spread along the observation area which are encoded in the chromosomes of the initial population. This heuristic criterion of sensor node selection causes the density to be more regular.

There is a risk of decreasing the genetic diversity and even of causing some genetic drift when a deterministic algorithm is used to create individuals within an initial population.

To avoid this possible drawback, some sort of stochastic behavior was maintained. The algorithm is described as follows.

Table 1: Part of a chromosome with binary encoding.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	⋯
0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	⋯

Table 2: Part of a chromosome with the new compact encoding.

3	5	11	16	2	5	10	13	4	8	9	15	1	7	⋯

The population is divided into two classes of individuals: the random and the heuristic.

A parameter controls the probability of each individual being generated by the heuristic approach. The others are created by the standard random procedure. The heuristic starts picking a random sensor node for the first gene.

Each other sensor node is selected as having the largest of the smallest distance from other previously selected sensor nodes. This procedure is repeated for each group of genes that represents the set of sensor nodes of each time interval.

5.2. Compact Chromosome Encoding. According to [24], the right representation of the individuals is one of the most difficult parts of designing a good evolutionary algorithm.

The binary chromosome encoding was used in the original version of the hybrid methodology. Each gene represents the inclusion of the equivalent element of the reducible structure that will be considered in the generation of the new subproblem. It is well suited for the cutting and packing problem class for which the methodology was designed. This type of chromosome encoding however is not appropriate for other problem domains like the one treated in this work. It would generate too large chromosomes (i.e., 10 times intervals $\times$ 36 sensors). Table 1 shows a possible chromosome with the binary encoding. Each color represents a set of 16 genes associated to its respective sensors of the each time interval.

The proposed new encoding (Table 2) [18] represents the integer indexes of the sensors that must be taken in the subproblem generation. So there is no need of representing all sensors. Only a small amount of sensors has to be considered and the length of this chromosome can be down to 17% of the binary encoding one. In the original version of the hybrid methodology, the density rise was a problem as described in Section 4. The first resource created to avoid this undesirable effect was the density control operator that effectively accomplished its goal and is expected to be published soon. Here, the issue is solved with a much more controlled expedient: constant density.

This new compact chromosome encoding has a side effect of turning the density constant, since the ratio of sensors considered in the subproblem and the total number of sensors is always fixed. Now the solver can work in its best range of operation, balancing efficiency and effectiveness.

6. Computational Results

The dynamic coverage and connectivity in wireless sensor networks problem is a very different problem class than the cutting and packing class used in the original version. This means a good opportunity to motivate changes in the Generate and Solve methodology towards flexibility. Moreover, the publication of some papers related to this subject [28] brought a good understanding about how to give new contributions to this area and present better solutions.

Experiments were made for the dynamic coverage and connectivity in wireless sensor networks problem using the mentioned hybrid methodology.

It follows most premises of Section 2. The grid sensor placement was used for simplicity sake because the random scenario did not present significant variation of the problem complexity which is the main concern of these experiments. The machine used on this test was an Intel Core 2 Quad 64 bits with 8 GB of RAM machine with openSUSE Linux 11.0 64 bits. As integer linear programming (ILP) solver, the IlogCplex 10.1 dynamic library [29] was used attached to the Java program implementation of the methodology. The pure integer linear Programming approach is a particular case of the methodology and is obtained by a proper parameter set in a XML script.

Table 3 presents the comparison of hybrid methodology (HM) and integer linear programming (ILP) approaches. In these experiments, the demand points are disposed in a grid. Due to the stochastically nature of the HM, it is presented the average and standard deviation of results found in a batch of 10 problem instances. The notation used here is the average value followed by the $\pm$ sign and the standard deviation value.

In [30, 31], are conceived their computational experiments to produce WSNs with the minimum energy consumption as possible, while maintaining the coverage and connectivity constraints. The gain obtained in their paper is calculated by the comparison of the minimum set of sensors that have to be active and the waste of energy caused by the activation of all sensors with high coverage overlapping in a high-density configuration [1].

This idea seems coherent at first glance since energy waste reduction is often desirable. However, this point of view distorts the actual need of an WSN that stands in a place where it is infeasible to change the batteries: extend its lifetime as far as possible.

The objective function is composed of two parts. The summation of electrical charge consumption in all sensor nodes and the penalties. The penalties are an artifact that allow uncovered demand points giving flexibility to the model, but at the same time avoid the unnecessary use of this

Table 3: Simulation results for demand point in grid com ILP.

	Hybrid methodology	Hybrid methodology	Integer linear programming
Time intervals	6	10	10
Demand points	400	400	400
Sensor nodes	36	36	16
Sinks	1	1	1
Time (minutes)	151.78 ± 14.61	189.80 ± 61.01	298.55
Time for first final solution	79.84 ± 54.65	104.14 ± 75.06	298.55
Uncovered demand points (%)	0.93 ± 0.39	2.35 ± 0.54	0.33
Real objective	22,223.27 ± 2,614.63	33,374.04 ± 2,389.76	26,665.91

Table 4: Simulation results for demand point in aleatory positions.

	Hybrid methodology	Hybrid methodology
Time intervals	6	10
Demand points	400	400
Sensor nodes	36	36
Sinks	1	1
Time (minutes)	146.77 ± 32.45	366.28 ± 13.54
Time for first final solution	91.55 ± 47.14	196.05 ± 78.30
Uncovered demand points (%)	1.56 ± 0.61	2.10 ± 0.53
Real objective	20,472.51 ± 4,135.35	32,522.51 ± 2,628.33

resource. Thus, the real objective is calculated by subtracting the artificial coverage penalties of the objective function or just calculating the first part (summation) of the objective expression.

Table 4 shows the results of similar experiments. However, this time the demand points are spread randomly through the sensed area.

The real purpose of this model is to extend the WSN lifetime as far as possible, preserving the WSN cost. So, lower electrical charge consumption is not necessarily an important issue if it does not reflect in more time slots. The number of time slots multiplied by the duration of each time slot represents this WSN lifetime.

Given this explanation it is reasonable to say that both solutions found by hybrid methodology (HM) and integer linear programming are equivalent in effectiveness. However, the HM approach can handle an amount of sensors 325% times larger, extending the working range of this application.

The only drawback here is the uncovered demand point rate, which is worse than ILP value. Despite this small imperfection of 2.35%, many real applications tolerate some lack of coverage by the nature of the observed phenomenon and other aspects. However, these uncovered demand points are often situated at the periphery of the observed area. The coverage radius does not reflect necessarily a sharp threshold of sensing.

Figure 3 shows the evolution of the best individual fitness in plain line and population fitness average in line with points as well.

Figures 4, 5, 6, 7, 8, 9, 10, 11, 12, and 13 are graphical representations of 10 time slots of a solution example. It shows the active sensor nodes, their coverage radii, the covered and uncovered demand points, and the routes from sensor nodes to the sink in the center.

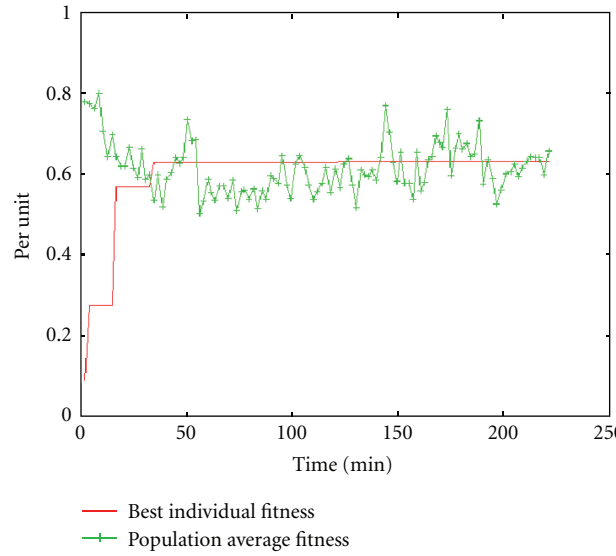


Figure 3: Best individual and population average evolution.

7. Conclusion and Future Works

This hybrid methodology is not only suitable for solving complex instances in the domain of cutting and packing problems, but also can be adapted to tackle other problem classes like WSN as shown here. The key point in this adaptation is finding the best or at least a good reducible structure. This

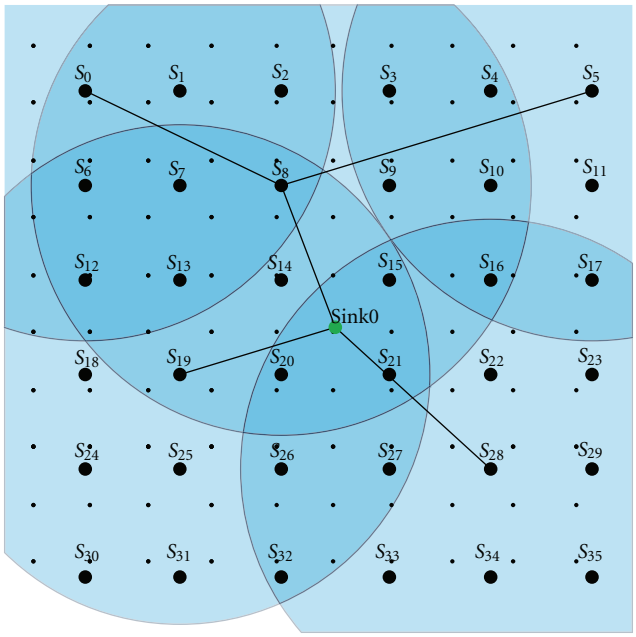


Figure 4

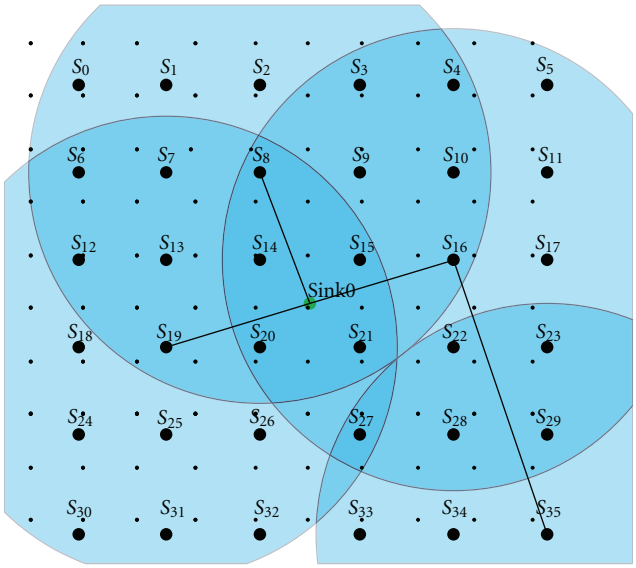


Figure 5

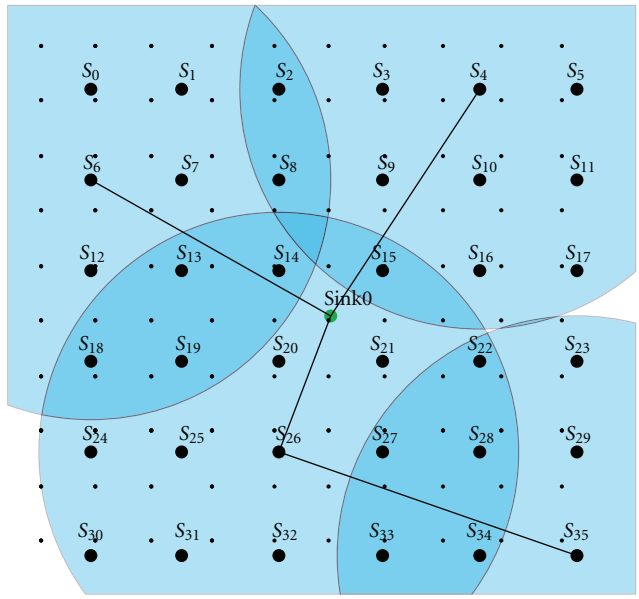


Figure 6

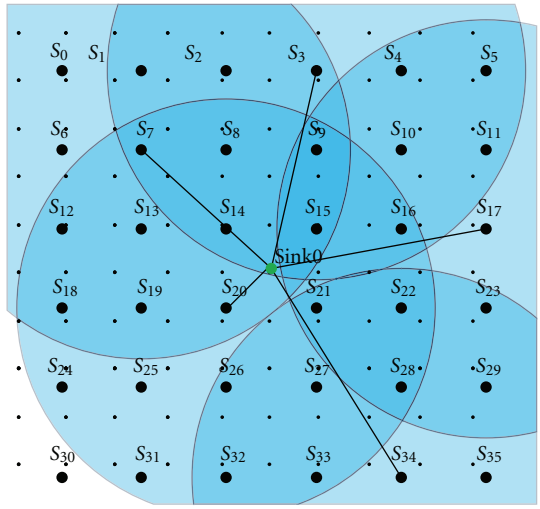


Figure 7

analysis is very linked to the chromosome encoding choice as it represents a tradeoff between subproblem complexity range width and chromosome size. A good reducible structure allows a wide range of subproblem complexity from very light and fast subproblems to the actual real problem. On the other hand, the reducible matrix size affects the chromosome size and a large chromosome size reduces the GA effectiveness.

In this problem, a good reducible structure was found, but it is much larger than the ones found in the cutting and packing problem instance. That is the reason why a new chromosome encoding was developed. This new encoding makes the matrix choice viable. The use of integer linear programming approach is limited to a certain level of complexity that sometimes is not enough for a real size network (Table 3).

The results found are far better than reference literature and leave opportunities of future enhancements as new supplementary algorithms and heuristics aggregated to this methodology. In [32] is implemented a hybrid methodology using genetic algorithm to deal with dynamic coverage and connectivity heterogeneous wireless sensor network. The difference between the homogeneous, addressed in this paper, and the heterogeneous, is the last one that has the capacity to deal with different phenomena independently, giving different sample data and sensed range for each phenomenon. This implies in a different way to treat the network's transmission data. Although it does not implement any density explosion control, good solutions are found in

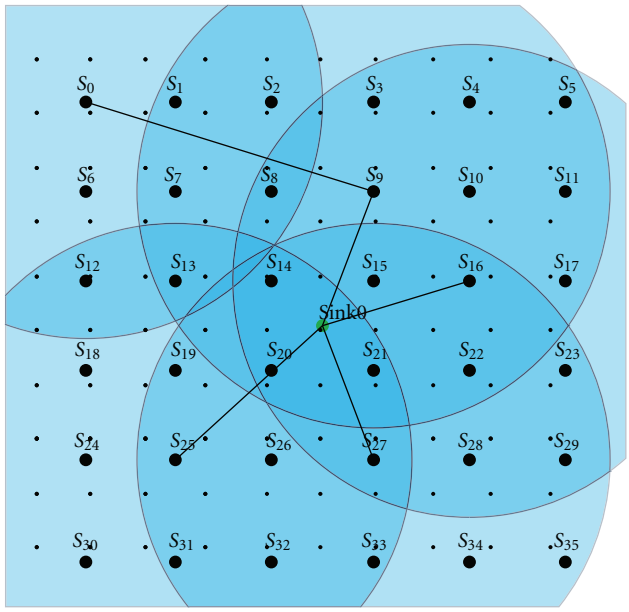


Figure 8

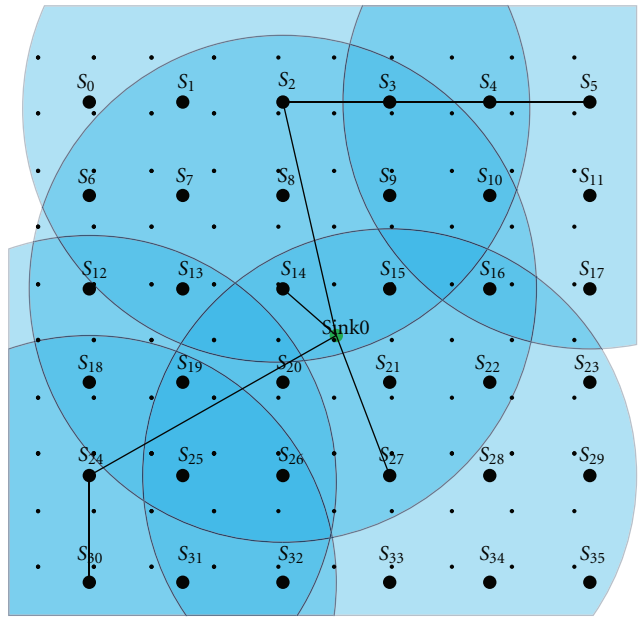


Figure 10

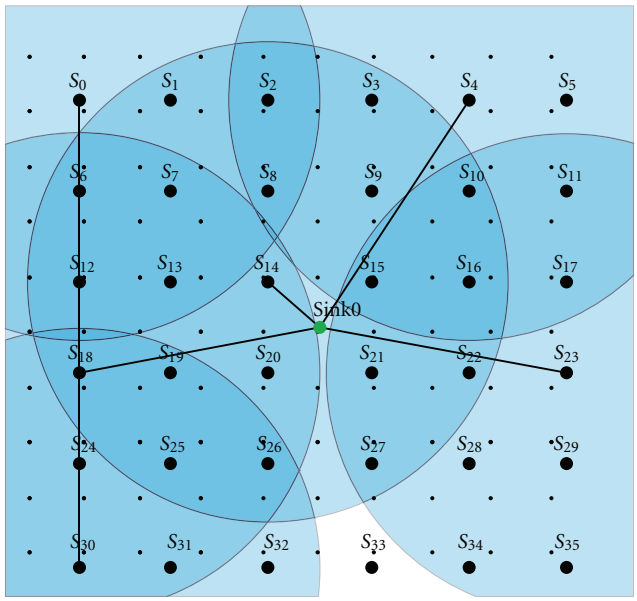


Figure 9

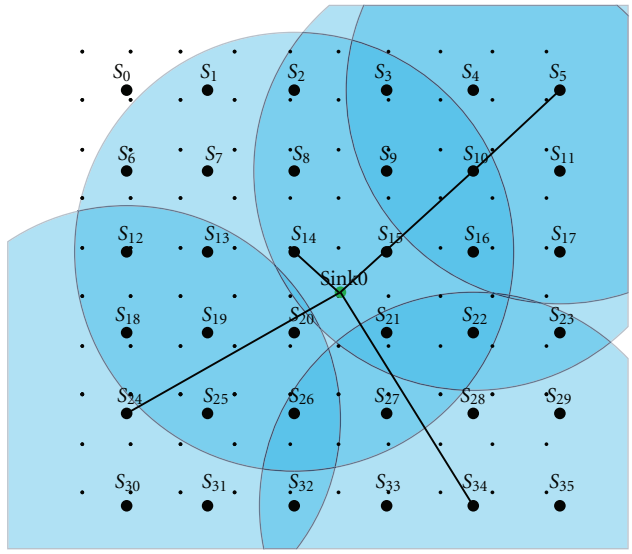


Figure 11

feasible execution time for wireless sensor network that could not be possible if only ILP was applied.

Another promising line of investigation involves the design and implementation of parallel/distributed versions of the framework, by means of which several GRIs (generator of reduced instances) and SRIs (solver of reduced instances) instances could run concurrently, each one configured to explore different aspects of the optimization problem at hand. The use of insular GA can bring more diversity and possibilities, resulting in effectiveness enhancement. Also, other metaheuristics such as particle swarm could be experimented replacing or working cooperatively with GA.

Abbreviations

S: Set of sensors
D: Set of demand points
M: Set of sinks
T: Set of n scheduling periods
AD_{ij}: Set of arcs ij, $i \in S$, $j \in D$ that link sensors to demand points
A_{ij}: Set of arcs ij, $i \in S$, $j \in S \cup M$ that interconnect sensors
EB_i: Accumulated battery charge for sensor $i \in S$
EA_i: Electrical charge dissipated while activating sensor $i \in S$

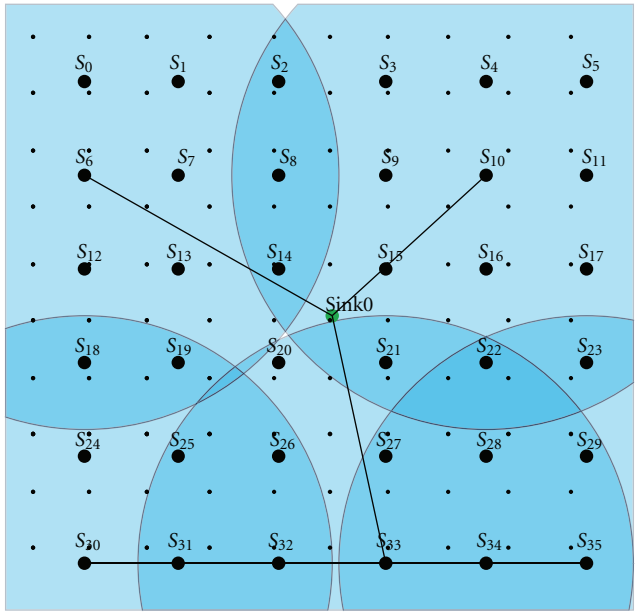


Figure 12

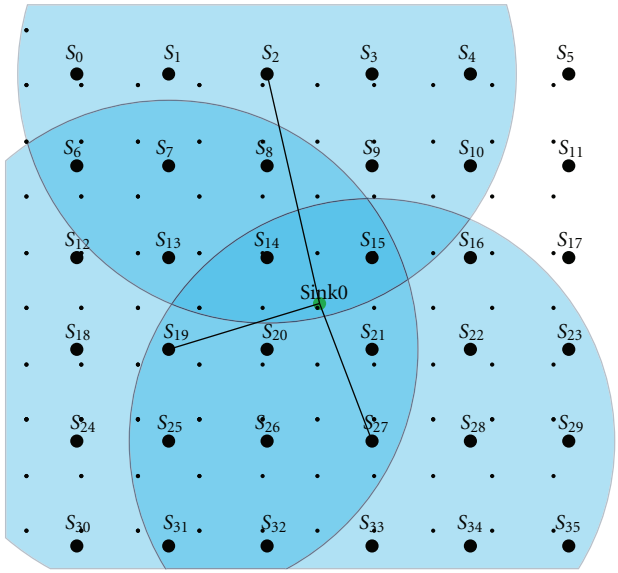


Figure 13

EM_i: Electrical charge dissipated while sensor $i \in S$ is activated (effectively sensing)

ET_{ij}: Energy dissipated when transmitting data from sensor $i \in S$ to sensor $j \in S$. Such valuescan be different for each arc ij if a sensor can have its transmitter power adjusted based on the distance to the destination sensor

ER_I: Energy expended in the reception of data for sensor $i \in S$

EH: Penalty applied when a demand point in any time interval is not covered by any sensor

x_{ij}^t: If sensor i covers demand point $j \in D$ in period $t \in T$

z_{lij}^t: If arc ij belongs to the route from sensor $l \in S$ to a sink in period $t \in T$

w_l^t: If sensor i was activated in period t for at least one phenomenon

y_i^t: If sensor i is activated in period t

h_j^t: If demand point j is not covered by any sensor in period t

e_i: Electrical charge consumed by sensor i considering all time periods.

Acknowledgments

The first and second authors are thankful to National Counsel of Technological and Scientific Development (CNPq) via Grants no. 305844/2011-3 and no. 312934/2009-2, the third author is thankful to Coordination for the Improvement of Higher Education Personnel (CAPES), and the fourth author is thankful to Foundation for Support of Scientific and Technological Development Ceara State (FUNCAP) for the support received for this project. The authors also acknowledge IBM for making the IBM ILOG CPLEX Optimization Studio available to the academic community.

References

[1] A. B. de Aguiar, A. de Meneses Sobreira Neto, R. P. P. Cunha, P. R. Pinheiro, and A. L. V. Coelho, "A hybrid methodology for coverage and connectivity in wireless sensor network dynamic planning," in *Proceedings of the 41st Simpósio Brasileiro de Pesquisa Operacional*, Bento Gonçalves, Brazil, 2010.

[2] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 28–39, November 2003.

[3] P. R. Pinheiro, A. L. V. Coelho, A. B. Aguiar, and A. M. S. Neto, "Applying the generate and solve methodology in the problem of dynamic coverage and connectivity in wireless sensor networks," in *Proceedings of the Information Science and Industrial Applications (ISI) (SERSC '12)*, pp. 252–257, Cebu, Philippines, 2012.

[4] H. Zhou, T. Liang, C. Xu, and J. Xie, "Multiobjective coverage control strategy for energy-efficient wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 720734, 10 pages, 2012.

[5] F. P. Quintão, F. G. Nakamura, and G. R. Mateus, "Evolutionary algorithm for the dynamic coverage problem applied to wireless sensor networks design," in *Proceedings of the IEEE Congress on Evolutionary Computation (IEEE CEC '05)*, vol. 2, pp. 1589–1596, September 2005.

[6] H. Ishebabi, P. Mahr, C. Bobda, M. Gebser, and T. Schaub, "Answer set versus integer linear programming for automatic synthesis of multiprocessor systems from real-time parallel programs," *International Journal of Reconfigurable Computing*, vol. 2009, Article ID 863630, 11 pages, 2009.

[7] K. Nguyen, T. Nguyen, and S. C. Cheung, "On reducing communication energy using cross-sensor coding technique," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 837128, 12 pages, 2011.

[8] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak, "Exposure in wireless sensor networks: theory and practical solutions," *Wireless Networks*, vol. 8, no. 5, pp. 443–454, 2002.

[9] S. Megerian and M. Potkonjak, "Lower power 0/1 coverage and scheduling techniques in sensor networks," Tech. Rep. 030001, University of California, Los Angeles, Calif, USA, 2003.

[10] F. V. C. Martins, F. G. Nakamura, F. P. Quintão, and G. R. Mateus, "Model and algorithms for the density, coverage and connectivity control problem in flat WSNs," in *Proceedings of the International Network Optimization Conference*, 2007.

[11] F. G. Nakamura and G. R. Mateus, "Planejamento dinâmico para controle de cobertura e conectividade em redes de sensores sem fio," in *Proceedings of the Workshop de Comunicação sem Fio e Computação Móvel*, vol. 1, pp. 182–191, 2004.

[12] A. B. de Aguiar, P. R. Pinheiro, and A. L. V. Coelho, "Optimizing energy consumption in heterogeneous wireless sensor networks: a novel integer programming model," in *Proceedings of the 4th International Conference on Operational Research for Development (ICORD '07)*, pp. 496–505, 2007.

[13] A. B. de Aguiar, P. R. Pinheiro, A. L. V. Coelho, A. S. Neto, and R. P. P. Cunha, "Scalability analysis of a novel integer programming model to deal with energy consumption in heterogeneous wireless sensor networks," *Communications in Computer and Information Science*, vol. 14, pp. 11–20, 2008.

[14] A. B. Aguiar, *Tackling the problem of dynamic coverage and connectivity in wireless sensor networks with an extended version of the generate and solve methodology generate and solve methodology [M.S. dissertation]*, Graduate Program in Applied Informatics, University of Fortaleza, 2009.

[15] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, 2003.

[16] I. Dumitrescu and T. Stützle, "Combinations of local search and exact algorithms," *Lecture Notes in Computer Science*, vol. 2611, pp. 211–223, 2003.

[17] E. G. Talbi, "A taxonomy of hybrid metaheuristics," *Journal of Heuristics*, vol. 8, no. 5, pp. 541–564, 2002.

[18] P. R. Pinheiro, A. L. V. Coelho, A. B. de Aguiar, and T. O. Bonates, "On the concept of density control and its application to a hybrid optimization framework: investigation into cutting problems," *Computers and Industrial Engineering*, vol. 61, pp. 463–472, 2011.

[19] N. V. Nepomuceno, P. R. Pinheiro, and A. L. V. Coelho, "Tackling the container loading problem: a hybrid approach based on integer linear programming and genetic algorithms," in *Proceedings of the EvoCOP*, vol. 4446 of *Lecture Notes in Computer Science*, pp. 154–165, Springer, Valência, Spain, 2007.

[20] N. Nepomuceno, P. Pinheiro, and A. L. V. Coelho, "A hybrid optimization framework for cutting and packing problems: case study on constrained 2D non-guillotine cutting," *Studies in Computational Intelligence*, vol. 153, pp. 87–99, 2008.

[21] L. J. P. Araújo and P. R. Pinheiro, "Combining heuristics backtracking and genetic algorithm to solve the container loading problerm with weight distribution," *Advances in Intelligent and Soft Computing*, vol. 73, pp. 95–102, 2010.

[22] L. J. P. de Araujo and P. R. Pinheiro, "Applying backtracking heuristics for constrained two-dimensional guillotine cutting problems," *Lecture Notes in Computer Science*, vol. 7030, pp. 113–120, 2011.

[23] A. P. Bhondekar, R. Vig, M. L. Singla, C. Ghanshyam, and P. Kapur, "Genetic algorithm based node placement methodology for wireless sensor networks," in *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS '09)*, vol. 1, pp. 18–20, Hong Kong, 2009.

[24] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.

[25] T. Back, D. B. Fogel, and Z. Michalewicz, *Handbook of Evolutionary Computation*, IOP, Bristol, UK, 1997.

[26] L. A. Wolsey, *Integer Programming*, John Wiley & Sons, 1998.

[27] J. Puchinger and G. R. Raidl, "Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification," in *Proceedings of the 1st International Work Conference on the Interplay between Natural and Artificial Computation (IWINAC '05)*, vol. 3562 of *Lecture Notes in Computer Science*, pp. 41–53, June 2005.

[28] A. B. de Aguiar, A. d. M. S. Neto, P. R. Pinheiro, and A. L. V. Coelho, "Applicability of a novel integer programming model for wireless sensor networks," *International Journal of Computer Science and Information Security*, vol. 20093, no. 1, pp. 7–13, 2009.

[29] ILOG. ILOG CPLEX 10.0 User's Manual. United States of America, January 2006.

[30] I. B. D. De Andrade, G. R. Mateus, and F. G. Nakamura, "A GRASP heuristic to density control: solving multi-period coverage and routing problems in wireless sensor networks," in *Proceedings of the 14th IEEE Symposium on Computers and Communications (ISCC '09)*, pp. 493–499, Sousse, Tunisia, July 2009.

[31] F. P. Quintão, F. G. Nakamura, and G. R. Mateus, "A hybrid approach to solve the coverage and connectivity problem in wireless sensor networks," in *Proceedings of the 4th European Workshop on Metaheuristics: Design and Evaluation of Advanced Hybrid Metaheuristics*, Nottingham, UK, 2004.

[32] A. M. S. Neto, "Um modelo em otimização de energia aplicado às redes de sensores sem fio heterogêneas," Graduation Conclusion Work, Graduate in Computer Science, University of Fortaleza, 2010.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 605494, 11 pages
doi:10.1155/2012/605494

Research Article

An Efficient Reliable Communication Scheme in Wireless Sensor Networks Using Linear Network Coding

Jin Wang,[1] Xiumin Wang,[2] Shukui Zhang,[1,3] Yanqin Zhu,[1] and Juncheng Jia[1]

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *School of Computer and Information, Hefei University of Technology, Hefei 230009, China*
[3] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

Correspondence should be addressed to Xiumin Wang, xiuminwang2@gmail.com

Received 1 August 2012; Accepted 6 September 2012

Academic Editor: Yong Sun

We address the modeling and design of *linear network coding* (LNC) for reliable communication against multiple failures in wireless sensor networks (WSNs). To fulfill the objective, we design a deterministic LNC scheme *RDLC* based on the average number of path failures simultaneously happening in the network other than the maximum number of path failures. The scheme can significantly improve the network throughput comparing with the traditional approaches. In our study, we also investigate the potential of *random* linear code *RRLC* for providing reliable communication in WSNs and prove the low bound of the probability that the RRLC can provide the reliable communication. Finally, extensive simulation experiments have been conducted, and the results demonstrate the effectiveness of the proposed LNC schemes.

1. Introduction

In recent years, wireless sensor network (WSN) has attracted significant attention for future generations of wireless applications in industry, agriculture, and military [1–5]. Despite its salient potentials, there are still many challenges to be addressed. In this paper, we will investigate how to provide reliable data transmissions in WSNs, which is very challenging because not only the wireless medium is vulnerable to severe channel fading and interference but also the sensor node always faces to energy exhaustion and physical damage.

Clearly, how to provide reliable communication in WSNs can be studied from multiple layers, including the physical layer and the MAC layer [6–9]. In this work, we will focus on the network layer and transport layer. Particularly, we will study how to provide reliable communication in WSNs against loss of data packets.

To provide reliable communication against node or link failures, there are two kinds of traditional approaches [10]: *the proactive recovery* and *the reactive recovery*, which aim at recovering the data transmission when node or link failures happen. The proactive recovery approaches can recover data transmission immediately when a link or node failure occurs, because it reserves the bandwidth (in backup paths) between the source node and the destination node in advance and the data flow is simultaneously transmitted on both primary paths and backup paths. On the other hand, the proactive recovery approaches do not provide any immediate recovery in advance. When the failure occurs on the routing paths of the data flow, the affected flow will be retransmitted to the destination by using available bandwidth in the network. Therefore, although the proactive recovery approaches can recover data transmission immediately when the link or node failures happen, network resources (e.g., bandwidth on the backup paths) are wasted when no failure happens. For the proactive recovery approaches, although network resources are used efficiently (no network resource is reserved in advance), the procedure of transferring the data traffic from the failed paths to the new routing paths incurs a considerable delay to data communications.

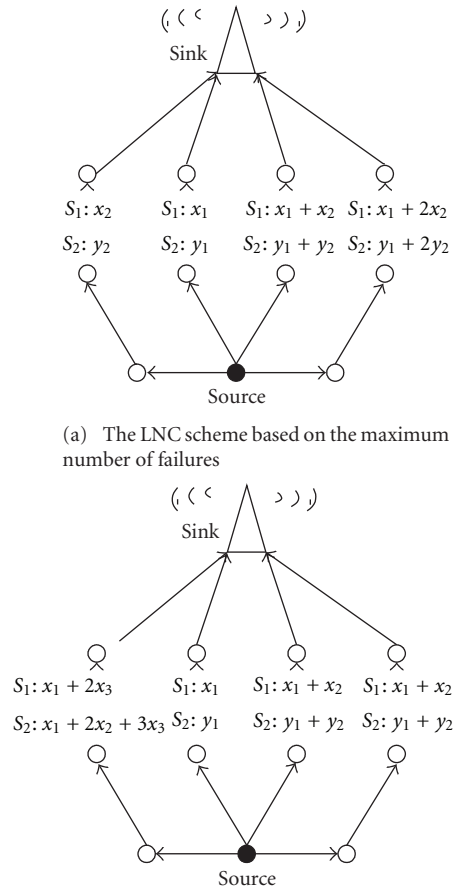
Following its success in maximizing network throughput [11–15], linear network coding (LNC) has recently been shown to be a promising approach that can achieve reliable communication with much better efficiency, because it can be used to provide protection in a proactive manner with the bandwidth cost in a reactive manner [16–20]. LNC has

been first introduced by Kamal [16, 19] to provide 100% protection against single link failures in optical network. Recently, the design of reliable communication based on LNC has attracted an increasing amount of attention in optical networks [16], wireless networks [18], and hybrid wireless-optical access networks [20], respectively. However, in these studies, the LNC scheme, which can generate large number of linear combinations of original data packets to provide reliable communication when link or node failures happen, is designed according to the maximum number of failures ($f_{\max}$, i.e., the worst case) happening simultaneously. In this case, with such LNC schemes, the destination can receive sufficient coded data packets and recover the original data packets, as long as the number of failures happening simultaneously is no more than $f_{\max}$. However, the number of failures happening simultaneously in WSNs varies at different times in practice. When the number of failures happening simultaneously in the network is less than $f_{\max}$, the network throughput decreases because the capacity of the WSN is wasted by the redundant coded data packets transmitted in the WSN.

Therefore, our key idea is to increase the number of original data packets transmitted and recovered in the transmission rounds with large number of failures happening simultaneously, by utilizing the redundant coded data packets transmitted in the transmission rounds with fewer failures happening simultaneously in the network to improve the network throughput.

Figure 1 gives an example to compare the network throughput that can be achieved by LNC scheme based on the maximum number of failures $f_{\max}$ and the LNC scheme based on the average number of failures $f_{\exp}$ (denoted in Section 2.1). Specifically, in the WSN, there are 4 edge-disjoint paths between the source S and the destination (sink) D. Suppose that $f_{\max} = 2$ and $f_{\exp} = 1$. Without loss of generality, we assume that 2 path failures happen in time slot 1 and no failure happens in time slot 2. Different LNC schemes are shown in Figures 1(a) and 1(b), in which the notation "$S_1 : \mathbf{x}_1 + \mathbf{x}_2$" on a link denotes the link is used to transmit a coded symbol $\mathbf{x}_1 + \mathbf{x}_2$ at time slot 1. Moreover, for LNC schemes shown in both Figures 1(a) and 1(b), the coded data packets transmitted on the links belonging to the same path are same in given time slot.

Since the $f_{\max} = 2$, in Figure 1(a), the LNC scheme designed based on $f_{\max}$ can ensure that the destination can decode and recover even if 2 path failures happen. However, no failure happens in time slot 2, which decreases the network throughput because the capacity of the WSN is wasted by the redundant coded data packets transmitted in time slot 2. On the other hand, in Figure 1(b), the LNC scheme was designed based on $f_{\exp}$, in which part of original data packets sent previously is still encoded into the following time slots. In this case, the network throughput can be improved by utilizing the redundant coded data packets transmitted in time slot 2. Specifically, with different LNC schemes, the sink can recover 4 original data packets in 2 time slots (i.e., the achieved network throughput is 2) in the case shown in Figure 1(a), while the sink can recover 6 original data packets in 2 time slots (i.e., the achieved network throughput is 3) in the case shown in Figure 1(b). Obviously, the LNC scheme based on the average number of path failures can significantly improve the network throughput.



(a) The LNC scheme based on the maximum number of failures

(b) The LNC scheme based on the average number of failures

Figure 1: Reliable communication using LNC: an example.

In this paper, we will design an efficient LNC scheme based on the average number of path failures, to provide reliable communication in WSNs. It must be noted that our study in this paper is different to our previous work in [21], where we have investigated the LNC design for providing $N + k$ protection in the wireless mesh networks. Specifically, as we can see from the rest of the paper, the main differences include the following.

Firstly, different LNC designs are given to provide reliable communication in this paper. The deterministic LNC scheme proposed in Section 3 achieves the network throughput $N = L - \lceil f_{\exp} \rceil$ (L denotes the number of edge-disjoint paths between the source and the destination and $f_{\exp}$ is the expected number of path failures per time slot), while the LNC scheme proposed in [21] does not ensure that the network throughput can achieve $N = L - \lceil f_{\exp} \rceil$. Therefore, the deterministic LNC scheme proposed in this paper can achieve higher network throughput than our previous work in [21]. Moreover, in this paper, we also investigate the potential of standard random LNC to thwart path failures in WSNs

and analyze the lower bound of the probability that a random LNC can provide reliable communication. On the other hand, the work in [21] only considered the deterministic LNC design.

Secondly, comparing with the work in [21], we not only give the theoretical design of deterministic LNC scheme and random LNC scheme but also conduct considerable simulations with four parameters to demonstrate the effectiveness of the proposed LNC schemes, from view of both network throughput and recovery delay. We also show the impact of the size of finite field on the performance of the proposed random LNC schemes, which is not investigated in [21].

The rest of the paper is organized as follows. Section 2 describes the system models and problem description. In Section 3, we design a deterministic LNC scheme to provide reliable communication in WSNs and give the theoretically analysis to show the achieved network throughput. The analysis of the usage of random LNC to realize reliable communication in WSNs is shown in Section 4. Simulation results are given in Section 5. Finally, we conclude the paper in Section 6.

2. Reliable Communication against Multiple Failures in WSNs Based on LNC

In this section, we will describe the problem studied in this paper. Specifically, we first introduce the network model. We then give the description of the reliable communication problem.

2.1. The Network Model. In this paper, we consider a multihop wireless sensor network as a directed acyclic graph $G = (V, E)$, where V is the set of sensor nodes and E is the set of edges. We assume that each edge in G has the same unit capacity. Note that the capacity of different edges can be different in practice. However, we can always convert an edge with a certain capacity C (a nature number) data units to C edges with unit capacity. Suppose that all data packets have the same unit size. There are one source S, one destination D and L edge-disjoint paths between them. We assume that it will cost one time slot that the source sends L coded packets through L edge-disjoint paths and gets the feedback from the destination.

Although the wireless medium is vulnerable and the sensor node always faces energy exhaustion and physical damage, LNC can be designed to tolerate the link or node failures and provide reliable communications in WSNs. Specifically, L coded data packets are generated at the source node by linearly combining original data packets and transmitted through L edge-disjoint paths in each time slot. Since there will be different number of edge-disjoint paths failing simultaneously in different transmission rounds, we denote p_i as the probability that i edge-disjoint paths are failed simultaneously, where $1 \le i \le L$. Therefore, $f_{\exp} = \sum_{i=1}^{L} i p_i$ is the expected number of path failures per time slot. Let $N = L - \lceil f_{\exp} \rceil$.

Suppose that the data stream arrive rate is N data packets per time slot; the data packets arrived are firstly buffered at the source S, waiting for encoding and transmission towards the destination D. We will design an LNC scheme to protect the data stream with arrive rate N.

2.2. The LNC Scheme. In practical LNC schemes, the source node first divides the whole file into fix-size original packets. Then, the coded packets can be generated by encoding the original packets together. Moreover, each coded packet in the network corresponds to an *encoding vector*, which consists of the coding coefficients that it is produced with respect to the set of original packets. When the destination node received sufficient coded packets, it can decode and recover the original packets according to their encoding vectors.

Due to the limited computational capability of the sensor nodes in WSNs, we assume that the intermediate sensor nodes simply store and forward encoding packets. Such assumption is practical because coding operations require extra computation capability which imposes the processing overheads and may slow down the switching speed.

2.3. Problem Description. To improve the network throughput, we design the LNC schemes based on the expected number of path failures per time slot ($f_{\exp}$). However, since the number of path failures may exceed $f_{\exp}$ in some time slots, which may cause that the destination cannot decode the received coded data packets, it is desirable to utilize the redundant coded data packets transmitted in the transmission rounds with fewer failures ($< f_{\exp}$) to improve the network throughput.

Therefore, in this paper, we aim to design an LNC scheme providing reliable communication in WSNs to achieve network throughput $N = L - \lceil f_{\exp} \rceil$, in which $f_{\exp}$ is the expected number of path failures per time slot. We will show in the rest of the paper that the original data packets can be recovered by the destination and the network throughput N can be achieved, if only the average number of failures is no more than $f_{\exp}$.

2.4. Notations. To facilitate further discussions, we summarize main notations to be used throughout the rest of the paper in Table 1. We also denote the $L \times iN$ dimensional matrix $\mathbf{\Lambda}_i$ as follows:

$$\mathbf{\Lambda}_i = \begin{pmatrix} 1 & \lambda_{(i-1)L+1} & \lambda_{(i-1)L+1}^2 & \cdots & \lambda_{(i-1)L+1}^{i*N-1} \\ 1 & \lambda_{(i-1)L+2} & \lambda_{(i-1)L+2}^2 & \cdots & \lambda_{(i-1)L+2}^{i*N-1} \\ & \cdots & & & \\ 1 & \lambda_{iL} & \lambda_{iL}^2 & \cdots & \lambda_{iL}^{i*N-1} \end{pmatrix}, \quad (1)$$

where $\lambda_{i_1} \neq \lambda_{i_2}$, for all $i_1 \neq i_2$.

3. Reliable Communication Using Deterministic Linear Network Coding

In this section, we provide a deterministic LNC scheme to provide reliable communication (*RDLC*) for multiple failures in WSNs.

We assume that a buffer exists in the source to buffer the newly arrived data packets and parts of original data packets

Table 1

Symbol	Definition
Bold font	Vectors, matrixes
Normal font	A normal number
$\mathbb{F}_q$	A finite field of size q, over which the LNC is defined
$\{\mathbf{B}\}_i^j$	A set of vector composed by i_{th} to j_{th} row vectors of matrix $\mathbf{B}$
$[\mathbf{B}]_{i,j}^{m,n}$	A matrix composed by the i_{th} to j_{th} row and m_{th} to n_{th} column of matrix B
$[\mathbf{B}]_i$	The i_{th} row vector of matrix B
$[\mathbf{B}]_{i,j}$	The matrix composed by the i_{th} to j_{th} row vectors of matrix B
$[\mathbf{V}]$	The matrix formed by vectors in the set $\mathbf{V}$ as its rows
Rank $(\mathbf{B})$	The rank of a matrix $\mathbf{B}$
$\mathbf{m}_{i,j}$	The j_{th}, $j \le N$ original data packet arrived at time slot i
$\mathbf{M}_i$	N original data packets $\mathbf{M}_i = \{\mathbf{m}_{i,1}, \ldots, \mathbf{m}_{i,N}\}$ arrived at time slot i
L	The number of edge-disjoint paths between the source and the destination
$\mathbf{0}_{m\times n}$	A $m \times n$ dimensional zero matrix
N	$N = L - \lceil f_{\exp} \rceil$, in which $f_{\exp}$ is the expected number of path failures per time slot
Span $(\cdot)$	Linear span of a set of row vectors of a matrix
Dim $(\cdot)$	Dimension of a linear space
$P(A)$	The probability that condition A is satisfied

arrived previously. We also assume that the destination has a buffer to buffer the coded data packets which have not been decoded. At the end of each time slot, the destination sends an acknowledgment to the source in order to report the number of coded data packets received in this time slot. Then, according to the number of coded data packets received by the destination in the previously time slots, the source first removes some original data packets arrived perviously in its buffer and then generates L new coded data packets by encoding the original data packets arrived previously and the newly arrived data packets together. After that, the source sends the L new coded data packets to the destination. At the end of time slot t, for all $t \ge 1$, if the destination has totally received no less tN coded data packet, it can decode and recover the tN original data packets, clear its buffer and send an acknowledgment to notify the source that all the received coded data packets are decoded.

We denote the N original packets arrived at time slot t as $\mathbf{M}_t = \{\mathbf{m}_{t,1}, \ldots, \mathbf{m}_{t,N}\}$. The details of *RDLC* scheme is shown in Algorithms 1 and 2.

Suppose that c_t denotes the number of coded data packets received by the destination at the end of time slot t; we have $c_t \le L$, for all $t > 0$. Let

$$T = \min_{t:t>0}\left\{ t \mid \sum_{i=1}^{t} c_i \ge tN \right\}. \tag{2}$$

Next, we will prove that the destination can decode and recover the TN original data packets once it receives no less than TN coded data packets.

Lemma 1. *When $T = 1$, the destination can decode and recover the N original data packets.*

Proof. When $T = 1$, the destination can receive $c_1 \ge N$ coded data packets. The global encoding vector of each coded data packets is one row vector in the matrix $\mathbf{\Lambda}_1$. According to the matrix $\mathbf{\Lambda}_1$, any N rows can construct a square $N \times N$ dimensional Vandermonde matrix. Since $\lambda_i \ne \lambda_j$, for all $i \ne j$, the determinant of the $N \times N$ dimensional Vandermonde matrix is not equal to 0. Hence, the destination receives N coded data packets with N linearly independent global encoding vectors. Therefore, the destination can decode and recover the N original data packets. □

Lemma 2. *When $T = 2$, the destination can decode and recover the $2N$ original data packets.*

Proof. When $T = 2$, the destination receives no less than $2N$ coded data packets. We suppose that it receives c_1 coded data packets in time slot 1. Without loss of generality, let these coded data packets be $[\mathbf{\Lambda}_1]_{1,c_1}^{1,N}[\mathbf{M}_1]$. Since the destination receives no less than $2N$ coded data packets we can select $2N - c_1$ coded data packets received by the destination in time slot 2. Without loss of generality, let these coded data packets be $[\mathbf{0}_{c_2\times c_1}\ [\mathbf{\Lambda}_2]_{1,c_2}^{1,2N-c_1}]\begin{bmatrix}[\mathbf{M}_1]\\ [\mathbf{M}_2]\end{bmatrix}$. Therefore, the matrix $\mathbf{Y}_2$ composed by global encoding vectors of the coded packets as its rows can be represented by

$$\mathbf{Y}_2 = \begin{bmatrix} [\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1} & [\mathbf{\Lambda}_1]_{1,c_1}^{c_1+1,N} & \mathbf{0} \\ \mathbf{0} & [\mathbf{\Lambda}_2]_{1,2N-c_1}^{1,N-c_1} & [\mathbf{\Lambda}_2]_{1,2N-c_1}^{N-c_1+1,2N-c_1} \end{bmatrix}. \tag{3}$$

Next, we will prove that $\mathrm{Rank}(\mathbf{Y}_2) = 2N$. If $c_1 = 0$, then $c_1 + c_2 = c_2 \ge 2N$. We have $\det(\mathbf{Y}_2) = \det([\mathbf{\Lambda}_2]_{1,2N}^{1,2N}) \ne 0$ because $[\mathbf{\Lambda}_2]_{1,2N}^{1,2N}$ is a $2N \times 2N$ dimensional Vandermonde matrix. Otherwise, if $c_1 > 0$, obviously, the block matrix $[\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1}$ in the left upper corner of the matrix $\mathbf{Y}_2$ is $c_1 \times c_1$ dimensional and has full rank. Therefore, each column vector in $[\mathbf{\Lambda}_1]_{1,c_1}^{c_1+1,N}$ can be written as a linear combination of the column vectors in $[\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1}$. Since the $(2N - c_1) \times c_1$ dimensional block matrix in the left lower corner of the matrix $\mathbf{Y}_2$ is zero matrix, the matrix $\mathbf{Y}_2$ can be transformed to be matrix $\mathbf{Y}_2'$ by column transformation:

$$\begin{aligned}\mathbf{Y}_2' &= \begin{bmatrix} [\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & [\mathbf{\Lambda}_2]_{1,2N-c_1}^{1,N-c_1} & [\mathbf{\Lambda}_2]_{1,2N-c_1}^{N-c_1+1,2N-c_1} \end{bmatrix} \\ &= \begin{bmatrix} [\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1} & \mathbf{0} \\ \mathbf{0} & [\mathbf{\Lambda}_2]_{1,2N-c_1}^{1,2N-c_1} \end{bmatrix},\end{aligned} \tag{4}$$

and $\det(\mathbf{Y}_2) = \det(\mathbf{Y}_2')$.

At time slot t:
N original data packets $\mathbf{M}_t = \{\mathbf{m}_{t,1}, \ldots, \mathbf{m}_{t,N}\}$ are arrived and stored into its buffer;
if $t = 1$ **then**
Generate L coded packets: $\mathbf{\Lambda}_1[\mathbf{M}_1]$ and send them to the destination through L edge-disjoint paths;
end if
if $t > 1$ **then**
Receives $ACK(c_{t-1})$ from the destination;
if $c_{t-1} = -1$ **then**
Clear the buffer, set $t = 1$ and get ready for a new transmission round;
else
Remove the c_{t-1} earliest arrived data packets in the buffer;
Generate L coded data packets:

$$\begin{bmatrix} \mathbf{m}'_{t,1} \\ \vdots \\ \mathbf{m}'_{t,L} \end{bmatrix} = [\mathbf{0}_{L\times\sum_{i=1}^{t-1} c_i} \quad [\mathbf{\Lambda}_t]_{1,L}^{1,tN-\sum_{i=1}^{t-1} c_i}] \begin{bmatrix} [\mathbf{M}_1] \\ \vdots \\ [\mathbf{M}_t] \end{bmatrix}$$

Send each coded packet $\mathbf{m}'_{t,i}$ with encoding vector
$\mathbf{v}_{t,i} = [\mathbf{0}_{L\times\sum_{i=1}^{t-1} c_i} \quad [\mathbf{\Lambda}_t]_{1,L}^{1,tN-\sum_{i=1}^{t-1} c_i}]_i$ through path i to the receiver, $\forall i \in \{1, \ldots, L\}$;
end if
end if

Algorithm 1: Encoding at the source.

At time slot t:
Received c_t coded data packet in time slot t;
If $\sum_{i=1}^{t} c_i \geq tN$ **then**
Extend each received encoding vector $\mathbf{v}_{i,j}$ to its corresponding global encoding vector $\mathbf{v}'_{i,j}$ with length tN by adding $(t-i)N$ zeros, i.e., $\mathbf{v}'_{i,j} = [\mathbf{v}_{i,j} \quad \mathbf{0}_{1\times(t-i)N}]$;
Suppose that the matrix $\mathbf{Y}_T$ is composed by the global encoding vectors of the received TN coded packets as its rows and the matrix $\mathbf{M}'$ is composed by corresponding coded packets as its rows;
Decode and recover the tN original data packets as follows:

$$\begin{bmatrix} [\mathbf{M}_1] \\ \vdots \\ [\mathbf{M}_t] \end{bmatrix} = \mathbf{Y}_T^{-1}\mathbf{M}';$$

Clear the its buffer and send a $ACK(-1)$ to the source;
else
Store the c_t coded data packet to its buffer;
Send a $ACK(c_t)$ to the source;
end if

Algorithm 2: Decoding at the destination.

We have

$$\begin{aligned} \det(\mathbf{Y}_2) &= \det\left(\begin{bmatrix} [\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1} & \mathbf{0} \\ \mathbf{0} & [\mathbf{\Lambda}_2]_{1,2N-c_1}^{1,2N-c_1} \end{bmatrix}\right) \\ &= \det\left([\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1}\right) \det\left([\mathbf{\Lambda}_2]_{1,2N-c_1}^{1,2N-c_1}\right). \end{aligned} \tag{5}$$

Since the block matrix $[\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1}$ is a $c_1 \times c_1$ dimensional Vandermonde matrix, $[\mathbf{\Lambda}_2]_{1,2N-c_1}^{1,2N-c_1}$ is a $(2N-c_1)\times(2N-c_1)$ dimensional Vandermonde matrix and $x_i \neq x_j$, for all $i \neq j$, we have $\det([\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1}) \neq 0$ and $\det([\mathbf{\Lambda}_2]_{1,2N-c_1}^{1,2N-c_1}) \neq 0$.

Therefore, we have $\det(\mathbf{Y}_2) \neq 0$, which indicates that $\text{Rank}(\mathbf{Y}_2) = 2N$, that is, when $i = 2$, the destination can decode and recover the original data packets if it receives no less than $2N$ coded data packets. □

Theorem 3. *The destination can decode and recover the TN original data packets as long as it totally receives no less than TN coded data packets at the end of T time slots.*

Proof. According to Lemmas 1 and 2, the theorem holds when $T = 1, 2$. When $T = t+1, t > 1$, it means that $\sum_{i=1}^{j} c_i < jN$, for all $j \leq t$ and $\sum_{i=1}^{t+1} c_i \geq (t+1)N$. When $\sum_{i=1}^{t+1} c_i = (t+1)N$, that is, $c_T = TN - \sum_{i=1}^{t} c_i$, without loss of generality, let these coded data packets be

$$\left[\mathbf{0}_{c_t\times\sum_{i=1}^{t-1} c_i} \quad [\mathbf{\Lambda}_t]_{1,c_t}^{1,tN-\sum_{i=1}^{t-1}}\right] \begin{bmatrix} [\mathbf{M}_1] \\ \vdots \\ [\mathbf{M}_t] \end{bmatrix}. \tag{6}$$

The matrix $\mathbf{Y}_T$ composed by the global encoding vectors of these TN coded packets as its rows can be represented by

$$\begin{bmatrix} [\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1} & * & \cdots & * & * \\ \mathbf{0} & [\mathbf{\Lambda}_2]_{1,c_2}^{1,c_2} & \cdots & * & * \\ & & \cdots & \ddots & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & [\mathbf{\Lambda}_T]_{1,c_T}^{1,(t+1)N-\sum_{i=1}^{t} c_i} \end{bmatrix}. \quad (7)$$

Similarly to the proof of Lemma 2, the block matrix $[\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1}$ in the left upper corner of the above matrix is $c_1 \times c_1$ dimensional and has full rank. Therefore, each column vector in $[\mathbf{Y}_T]_{1,c_1}^{c_1+1,TN}$ can be written as a linear combination of the column vectors in $[\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1}$. Since the $(TN - c_1) \times c_1$ dimensional block matrix in the left lower corner of the matrix $\mathbf{Y}_T$ is zero matrix, the elements in between 1 to c_1 row and $c_1 + 1$ to $TN - c_1$ will be transformed to zeros in matrix $\mathbf{Y}_T$ by column transformation. And then by using the full rank block matrix $[\mathbf{\Lambda}_2]_{1,c_2}^{1,c_2}$, the elements in between $c_1 + 1$ to $c_1 + c_2$ row and $c_1 + c_2 + 1$ to $TN - c_1$ will be transformed to zeros in matrix $\mathbf{Y}_T$ by column transformation.

By step to step column transformation, the matrix $\mathbf{Y}_T$ can be finally transformed to be $\mathbf{Y}'_T$:

$$\begin{bmatrix} [\mathbf{\Lambda}_1]_{1,c_1}^{1,c_1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & [\mathbf{\Lambda}_2]_{1,c_2}^{1,c_2} & \cdots & \mathbf{0} & \mathbf{0} \\ & & \cdots & \ddots & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & [\mathbf{\Lambda}_T]_{1,c_T}^{1,(t+1)N-\sum_{i=1}^{t} c_i} \end{bmatrix}. \quad (8)$$

Obviously, $\det(\mathbf{Y}'_T) = \prod_{i=1}^{T} \det([\mathbf{\Lambda}_i]_{1,c_i}^{1,c_i})$. Since each matrix $[\mathbf{\Lambda}_i]_{1,c_i}^{1,c_i}$ is a square Vandermonde matrix and $x_i \neq x_j$, for all $i \neq j$, we have $\det([\mathbf{\Lambda}_i]_{1,c_i}^{1,c_i}) \neq 0$, for all $i \in \{1, \ldots, T\}$. Therefore, $\det(\mathbf{Y}'_T) \neq 0$. Since the column transformation on a matrix does not change the rank of the matrix, the matrix $\mathbf{Y}_T$ has full rank, that is, the destination can decode and recover TN organical packets.

When $c_T > TN - \sum_{i=1}^{t} c_i$, the destination can randomly select $TN - \sum_{i=1}^{t} c_i$ coded data packets. Together with the $\sum_{i=1}^{t} c_i$ coded packets received from time slot 1 to time slot t, the destination have TN coded packets. Similarly to the above proof, we can have the destination that can decode and recover TN original packets. □

Corollary 4. *If the average number of path failures is no more than f_{exp}, then the network throughput $N = L - \lceil f_{\text{exp}} \rceil$ can be achieved.*

Proof. When the average number of path failures is no more than f_{exp}, we have the average number of coded packets received by the destination in each time slot is $L - f_{\text{exp}}$. Therefore, there exists a time slot T that the destination totally receives $T(L - f_{\text{exp}})$ coded data packets at the end of T time slots. Since $N = L - \lceil f_{\text{exp}} \rceil < L - f_{\text{exp}}$, the destination totally receives $T(L - f_{\text{exp}}) > TN$ coded data packets at the end of T time slots. Therefore, according to Theorem 3, we have that the destination can decode and recover the TN original data packets, that is, the network throughput $N = L - \lceil f_{\text{exp}} \rceil$ is achieved. □

4. Reliable Communication Using Random Linear Network Coding

In the previous section, we have discussed how to construct linear network code at the source node in a deterministic manner to provide the reliable communication. In practice, *random linear coding* has been widely used in the literature [15, 22], because of the simplicity of the coding scheme. With random linear coding, random linear combinations of the packets can be forwarded by a node, which the node received previously, to outgoing edges. It has been proved in pervious work that such a simple approach can obtain valid linear codes for multicast with probability $(1 - d/q)^{\eta}$, where η is the number of edges with associated randomized coefficients, q is the size of the finite field, and d is the number of destination nodes.

In this section, we investigate the behavior of the random linear coding, when it is applied to the reliable communication problem we discuss in this paper. The usage of such linear code is similar to the one we discussed in Section 3. The coding operations are only done at the source node and destination node. The major difference is that, instead of computing the coding matrix $\mathbf{Y}_i$ at the source node, the elements of $L \times iN$ dimensional coding matrix $\mathbf{B}_i$ for time slot i are randomly chosen from the finite field $\mathbb{F}_q$ according to the number of coded packets received by the destination in each time slot. We referred to such random linear coding scheme as reliable random linear coding *RRLC*. Since the destination does not need to send an acknowledgment to notify the source that the number of coded packets received in each time slot, the communication overhead of RRLC, is smaller than the *RDLC*. We also show that the RRLC can ensure that the destination can recover the iN original packets when it receives iN coded packets with high probability.

Specifically, the L coded data packets sent during time slot i can be represented by

$$\mathbf{B}_i \begin{bmatrix} [\mathbf{M}_1] \\ \vdots \\ [\mathbf{M}_i] \end{bmatrix}. \quad (9)$$

Suppose that the number of coded packets received within time slot i by the destination node d is c_i, without loss of generality, let these coded data packets be

$$[\mathbf{B}_i]_{1,c_i} \begin{bmatrix} [\mathbf{M}_1] \\ \vdots \\ [\mathbf{M}_i] \end{bmatrix}. \quad (10)$$

Suppose the $(\sum_{w=1}^{i} c_w) \times (iN)$ dimensional matrix $\overline{\mathbf{B}}_i$ is

$$\begin{bmatrix} [\mathbf{B}_1]_{1,c_1} & 0 & 0 & \cdots & 0 \\ [\mathbf{B}_2]_{1,c_2}^{1,N} & [\mathbf{B}_2]_{1,c_2}^{N+1,2N} & 0 & \cdots & 0 \\ & \cdots & \ddots & & \\ [\mathbf{B}_i]_{1,c_i}^{1,N} & [\mathbf{B}_i]_{1,c_i}^{N+1,2N} & [\mathbf{B}_i]_{1,c_i}^{2N+1,3N} & \cdots & [\mathbf{B}_i]_{1,c_i}^{(i-1)N+1,iN} \end{bmatrix}. \quad (11)$$

Therefore, the total coded packets received by the destination node D during i time slots can be represented by

$$\overline{\mathbf{B}}_i \begin{bmatrix} [\mathbf{M}_1] \\ \vdots \\ [\mathbf{M}_i] \end{bmatrix}. \tag{12}$$

From above theoretical analysis, the destination D can decode and recover the set of iN original packets $\bigcup_{t=1}^{i} \mathbf{M}_t$, if and only if $\text{Rank}(\overline{\mathbf{B}}_i) = iN$.

Next, we will give the lower bound of the probability that the destination D can decode and recover the original packets once it totally receives no less than iN coded packets after time slot i.

We set $\sum_{k=1}^{0} f(x) = 0$ and $\prod_{k=1}^{0} f(x) = 1$, for all function $f(x)$. We first prove a lemma.

Lemma 5. *If $\sum_{w=1}^{i'} c_w \le i'N$, for all $i' \le i$, the probability that the $(\sum_{w=1}^{i} c_w) \times (iN)$ dimensional matrix $\overline{\mathbf{B}}_i$ has full rank is*

$$\prod_{w=1}^{i} \prod_{j=\sum_{k=1}^{w-1} c_k+1}^{\sum_{k=1}^{w} c_k} \left(1 - \frac{1}{q^{wn-j+1}}\right). \tag{13}$$

Proof. Obviously, the matrix $\overline{\mathbf{B}}_i$ has full rank, if and only if

$$[\overline{\mathbf{B}}_i]_j \notin \text{Span}\left([\overline{\mathbf{B}}_i]_{1,j-1}\right), \quad \forall j \in \left\{1, \ldots, \sum_{w=1}^{i} c_w\right\}. \tag{14}$$

We first give the theoretical analysis on the first c_1 row vectors in the matrix $\overline{\mathbf{B}}_i$. Suppose that the first j, $(j \in \{1, \ldots, c_1 - 1\})$ rows of $\overline{\mathbf{B}}_i$ are selected and $[\overline{\mathbf{B}}_i]_{j'} \notin \text{Span}([\overline{\mathbf{B}}_i]_{1,j'-1})$, for all $j' \{1, \ldots, j\}$, that is, $\text{Rank}([\overline{\mathbf{B}}_i]_{1,j}) = j$, for the $j+1$th row vector of the matrix $\overline{\mathbf{B}}_i$, the total number different vectors can be selected from the finite filed is q^N, because $[\overline{\mathbf{B}}_i]_{j+1}$, forall $j \in \{1, \ldots, c_1 - 1\}$ that have the first N elements randomly selected in finite filed $\mathbb{F}_q$ and other elements are 0. Let σ_{j+1} be the number of vectors which can be selected as the $j+1$th row vector of the matrix $\overline{\mathbf{B}}_i$ such that $[\overline{\mathbf{B}}_i]_{j+1} \notin \text{Span}([\overline{\mathbf{B}}_i]_{1,j})$. The total number of vectors in $\text{Span}([\overline{\mathbf{B}}_i]_{1,j})$ is q^j. Therefore, if $[\overline{\mathbf{B}}_i]_{j+1} \notin \text{Span}([\overline{\mathbf{B}}_i]_{1,j})$, we have $\sigma_{j+1} = q^N - q^j$.

Similarly, we then give the probability that the matrix $\overline{\mathbf{B}}_i$ has full rank.

Suppose that the first j, $(j \in \{\sum_{k=1}^{w-1} c_k + 1, \ldots, \sum_{k=1}^{w} c_k - 1\})$ rows of $\overline{\mathbf{B}}_i$ are selected and $[\overline{\mathbf{B}}_i]_{j'} \notin \text{Span}([\overline{\mathbf{B}}_i]_{1,j'-1})$, for all $j' \in \{1, \ldots, j\}$, that is, $\text{Rank}([\overline{\mathbf{B}}_i]_{1,j}) = j$, for the $j+1$th row vector of the matrix $\overline{\mathbf{B}}_i$, the total number different vectors can be selected from the finite filed which is q^{wN}, because $[\overline{\mathbf{B}}_i]_{j+1}$ for all $j \in \{\sum_{k=1}^{w-1} c_k + 1, \ldots, \sum_{k=1}^{w} c_k - 1\}$ have the first wN elements randomly selected in finite filed $\mathbb{F}_q$ and other elements are 0. Let σ_{j+1} be the number of vectors which can be selected as the $j+1$th row vector of the matrix $\overline{\mathbf{B}}_i$ such that $[\overline{\mathbf{B}}_i]_{j+1} \notin \text{Span}([\overline{\mathbf{B}}_i]_{1,j})$. The total number of vectors in $\text{Span}([\overline{\mathbf{B}}_i]_{1,j})$ is q^j. Therefore, if $[\overline{\mathbf{B}}_i]_{j+1} \notin \text{Span}([\overline{\mathbf{B}}_i]_{1,j})$, we have $\sigma_{j+1} = q^{wN} - q^j$.

Let σ be the number of matrix $\overline{\mathbf{B}}_i$ that can be constructed in the finite field which satisfies the condition shown in (14). We have

$$\sigma = \prod_{j=1}^{\sum_{k=1}^{i} c_k} \sigma_j = \prod_{w=1}^{i} \prod_{j=\sum_{k=1}^{w-1} c_k+1}^{\sum_{k=1}^{w} c_k} \left(q^{wN} - q^{j-1}\right). \tag{15}$$

The total number of different matrix $\overline{\mathbf{B}}_i$ with dimension $(\sum_{k=1}^{i} c_k) \times (iN)$ is $q^{\sum_{k=1}^{i} kc_kN}$, because there are $\sum_{k=1}^{i} kc_kN$ elements randomly selected in finite filed $\mathbb{F}_q$ and other elements are 0 in matrix $\overline{\mathbf{B}}_i$.

Therefore, the probability that the matrix $\overline{\mathbf{B}}_i$ has full rank is

$$\begin{aligned} \frac{\sigma}{q^{\sum_{k=1}^{i} kc_kN}} &= \frac{\prod_{w=1}^{i} \prod_{j=\sum_{k=1}^{w-1} c_k+1}^{\sum_{k=1}^{w} c_k} \left(q^{wN} - q^{j-1}\right)}{q^{\sum_{k=1}^{i} kc_kN}} \\ &= \prod_{w=1}^{i} \prod_{j=\sum_{k=1}^{w-1} c_k+1}^{\sum_{k=1}^{w} c_k} \left(1 - \frac{1}{q^{wN-j+1}}\right). \end{aligned} \tag{16}$$

□

Theorem 6. *If the total number of coded packets received by destination d during the first i time slots is no less than iN, that is, $\sum_{w=1}^{i} c_w \ge iN$ and for all $i' < i$, $\sum_{w=1}^{i'} c_w < i'N$, the lower bound of the probability that can decode and recover the set of iN original packets $\bigcup_{t=1}^{i} \mathbf{M}_t$ is*

$$\prod_{w=1}^{i} \prod_{j=\sum_{k=1}^{w-1} c_k+1}^{\min(\sum_{k=1}^{w} c_k, iN)} \left(1 - \frac{1}{q^{wn-j+1}}\right). \tag{17}$$

Proof. The destination node D can decode and recover the set of iN original packets $\bigcup_{t=1}^{i} \mathbf{M}_t$, if and only if $\text{Rank}(\overline{\mathbf{B}}_i) = iN$. Since $\sum_{w=1}^{i} c_w \ge iN$ and for all $i' < i$, $\sum_{w=1}^{i'} c_w < i'N$, we have $c_i \ge iN - \sum_{w=1}^{i-1} c_w$. Obviously, if the first iN row vectors are linearly independent, $\text{Rank}([\overline{\mathbf{B}}_i]_{1,iN}) = iN$, we have $\text{Rank}(\overline{\mathbf{B}}_i) = iN$. Therefore, the probability that $\text{Rank}(\overline{\mathbf{B}}_i) = iN$ is lower bounded by the probability that $\text{Rank}([\overline{\mathbf{B}}_i]_{1,iN}) = iN$. According to matrix $[\overline{\mathbf{B}}_i]_{1,iN}$, the number of row vectors in it belonging to time slot i is $c_i' = iN - \sum_{w=1}^{i-1} c_w$. Therefore, from the Lemma 5, the probability that $\text{Rank}([\overline{\mathbf{B}}_i]_{1,iN}) = iN$ is

$$\begin{aligned} &\frac{\prod_{w=1}^{i-1} \prod_{j=\sum_{k=1}^{w-1} c_k+1}^{\sum_{k=1}^{w} c_k} \left(q^{wN} - q^{j-1}\right) \prod_{j=\sum_{k=1}^{i-1} c_k+1}^{iN} \left(q^{iN} - q^{j-1}\right)}{q^{\sum_{k=1}^{i-1} kc_kN + (iN - \sum_{k=1}^{i-1} c_k)iN}} \\ &= \prod_{w=1}^{i} \prod_{j=\sum_{k=1}^{w-1} c_k+1}^{\min(\sum_{k=1}^{w} c_k, iN)} \left(1 - \frac{1}{q^{wn-j+1}}\right). \end{aligned} \tag{18}$$

Since the probability that $\text{Rank}(\overline{\mathbf{B}}_i) = iN$ is lower bounded by the probability that $\text{Rank}([\overline{\mathbf{B}}_i]_{1,iN}) = iN$, the probability that $\text{Rank}(\overline{\mathbf{B}}_i) = iN$ is large than

$$\prod_{w=1}^{i} \prod_{j=\sum_{k=1}^{w-1} c_k+1}^{\min(\sum_{k=1}^{w} c_k, iN)} \left(1 - \frac{1}{q^{wn-j+1}}\right). \tag{19}$$

□

5. Performance Evaluation

In this section, we conduct simulations to compare the scheme that without considering reliable communication (denoted as *Unreliable Communication Scheme*) the linear coding schemes are designed for recovering from maximum number of edge failures and the proposed linear coding schemes.

In the transmission scheme without network protection, the source sent totally L original packets along the L paths between the source node and the destination node in each time slot, until the destination node receive all the L original packets. Since each path may fail in each time slot, the destination may wait for some time slots to receive all the L original packets.

In the linear coding schemes designed for recovering from maximum number of edge failures, the scheme codes $L - f_{\max}$ original packets in each time slot and makes sure that the destination can decode all of them even if the maximum number of edge failures happens in one time slot (i.e., the worst case). However, such a scheme will waste lots of throughput of the network, because the number of edge failures probably is less than the maximum number of edge failures. We denote such schemes as *LCW* scheme.

Therefore, in our coding scheme, we code the packets sent in the pervious time slots with the new packets together which can fully utilize the available paths in the network to achieve higher throughput. To achieve this goal, in the proposed linear coding scheme, the destination receives coded packets instead original packets in each time slot which may not be decoded immediately. Therefore, the destination may wait for some time slots to receive enough coded packets to decode and recover all the original packets sent by the source node in the past time slots. However, we will show in this section that we can decrease a bit of throughput to achieve a much low decoding delay.

The objectives of the simulation conducted in this work are as follows.

(i) To compare the *throughput* achieved when using different schemes under different parameter settings. The throughput is referred to as the total number of original packets recovered (or reviewed) in the past time slots divided to the total number of time slots.

(ii) To compare the *recovery delay* using different schemes under different parameter settings. The delay will cause by different schemes to receive (or decode and recover) all the original packets sent by the source node in the past time slots.

5.1. Simulation Setup. We have four parameters in our simulations.

(i) The number of paths between the source node and the destination node, L, which varies from 10 to 20.

(ii) The maximum number of paths between the source node and the destination node which can simultaneously fail, $f_{\max} = \lfloor \alpha L \rfloor$, in which α varies from 0.2 to 0.7.

(iii) The number of original packets coded (or sent) in each time slots, N, which varies from 2 to $\lfloor L - F_{\max}/2 \rfloor - 1$.

(iv) The size of finite field, $q = 2^r$, in which r varies from 1 to 7.

In a network G, we suppose that there are L edge-disjoint paths between the source node S and the destination node D. In each time slot, there may be f', $0 \le f' \le f_{\max}$ edges failure and the packets transmitted on these paths cannot be received by the destination node D. In the simulation, we randomly select f' in $\{0, \ldots, f_{\max}\}$ and then select f' paths in the L paths which will fail in the following one time slot.

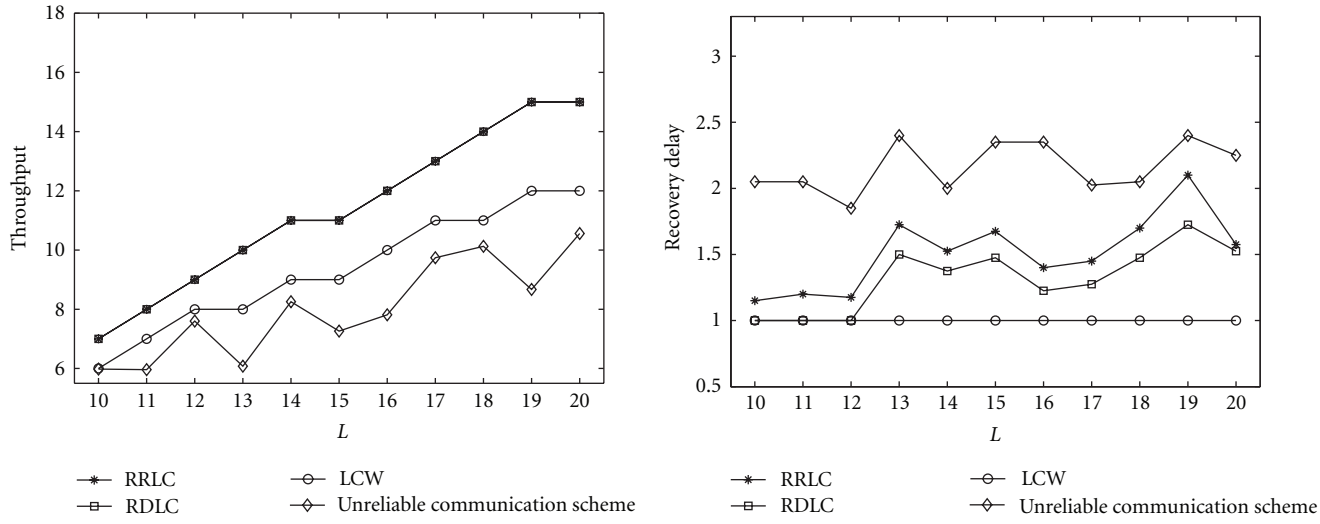
For each combination of parameters L, $f_{\max}$, N and q, we generate 100 instances. For each instance, we evaluate the performance of the data transmission using unreliable communication scheme, the linear coding schemes designed for recovering from maximum number of edge failures, and the proposed linear coding schemes.

5.2. Simulation Results. We compare the network throughput and the recovery delay of the proposed *RDLC* and *RRLC* schemes with *LLW* scheme and the unreliable communication scheme.

Firstly, in Figure 2, we set $q = 2^2$, $\alpha = 0.4$, $f_{\max} = \lfloor \alpha L \rfloor$, $N = \lceil L - f_{\max}/2 \rceil - 1$ and vary L in the range of $[10, 20]$.

In Figure 2(a), the throughput of all the four schemes increases with the increase of the number of edge-disjoint paths between the source node and the destination node. The reason is that the more the number of edge-disjoint paths exist between the source node and the destination node the more packets can be transmitted successfully to the destination. The throughput of the *LLW* scheme is higher than the unreliable communication scheme, because the *LLW* scheme exploits the LNC to ensure that all the original packets sent in one time slot can be recovered by the destination node which limit the number of original packets transmitted in each time slot. On the other hand, the unreliable communication scheme will transmit the same set of original packets many times to make sure the destination node can successfully receive all of them. The throughput of the proposed *RDLC* and *RRLC* schemes are much higher (more than 25% compared with *LLW* and more than 40% compared with the unreliable communication scheme when L grows sufficiently large), because the proposed schemes fully utilize the unfailed paths to transmit coded packets. Since the number of original packets transmitted in each time slot is the same in the proposed *RDLC* and *RRLC* schemes, the two schemes has the same throughput.

In Figure 2(b), the recovery delay of the unreliable communication scheme is higher than the other three schemes. The unreliable communication scheme will transmit the same set of original packets many times to make sure the destination node can successfully receive all of them. The recovery delay of the *LLW* scheme is always one time slot because a limited number of original packets are transmitted in each time slot to make sure the destination can decode and recover them in one time slot. The recovery delay of the proposed *RDLC* and *RRLC* schemes are between the unreliable



(a) Throughput

(b) Transmission delay

FIGURE 2: $q = 2^2$, $\alpha = 0.4$, $f_{\max} = \lfloor \alpha L \rfloor$, $N = \lceil L - f_{\max}/2 \rceil - 1$.

communication scheme and the *LLW* scheme. Therefore, we can observe that the proposed *RDLC* and *RRLC* schemes can achieve highest throughput and moderate recovery delay. The figure also shows that the recovery delay of the *RRLC* scheme is higher than the recovery delay of the *RDLC* scheme. The reason is that according to the theoretical analysis a destination may not be decode and recover the original packets when using the *RRLC* scheme, even if it totally received more than $i * N$ coded packets during the past i time slots. Therefore, when using the *RRLC* scheme, the destination will wait more time slots to achieve the same throughput as the *RDLC* scheme.

Secondly, in Figure 3, we set $L = 16$, $q = 2^2$, $f_{\max} = \lfloor \alpha L \rfloor$, $N = \lceil M - f_{\max}/2 \rceil - 1$ and vary α in the range of $[0.2, 0.9]$.

In Figure 3(a), the throughput of all the four schemes decreases with the increase of α, because the number of edge-disjoint paths between the source node and the destination node is fixed but the number of failure paths increases. We can observe that the throughput of *LLW* scheme will be lower than the unreliable communication scheme when α grows sufficiently large. The reason is that when α grows sufficiently large, the throughput of of *LLW* scheme $L - f_{\max}$ will decrease to zero, while the unreliable communication scheme can exploit unfailed paths to transmit packets. The throughput of the proposed *RDLC* and *RRLC* schemes are much higher (more than 60% compared with the unreliable communication scheme when α grows sufficiently large).

In Figure 3(b), the recovery delay of the proposed *RDLC*, *RRLC* schemes, and the unreliable communication scheme increase with the increase of α. Moreover, The recovery delay of the unreliable communication scheme is higher than the other three schemes. The recovery delay of the *LLW* scheme is always one time slot because a number of original packets transmitted in each time slot decreases when the α increase to make sure the destination can decode and recover them in one time slot.

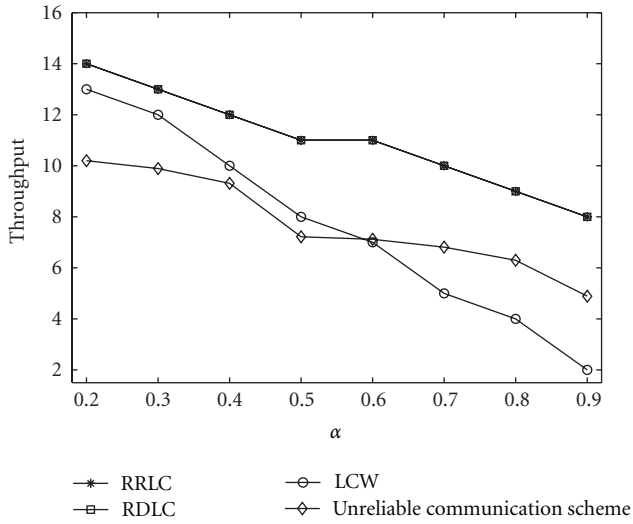
Thirdly, in Figure 4, we set $L = 16$, $q = 2^2$, $\alpha = 0.4$, $f_{\max} = \lfloor \alpha L \rfloor$, $N < \lceil M - f_{\max}/2 \rceil$.

In Figure 4(a), the throughput of the proposed *RDLC*, *RRLC* schemes increases with the increase of N, because the throughput of the proposed *RDLC*, *RRLC* schemes can achieve N only if N is no more than the total number of paths minus the average number of path failures each time slot. In our simulation, the condition is that $N \leq L - f_{\max}/2$. Therefore, in Figure 4(a), the throughput of the proposed *RDLC*, *RRLC* schemes can achieve N. Obviously, N does not have impact on the throughput of the unreliable communication scheme and the *LCW* scheme. We can see in Figure 4(a) that the throughput of the proposed *RDLC*, *RRLC* schemes outperforms the other two schemes when N is sufficiently large.

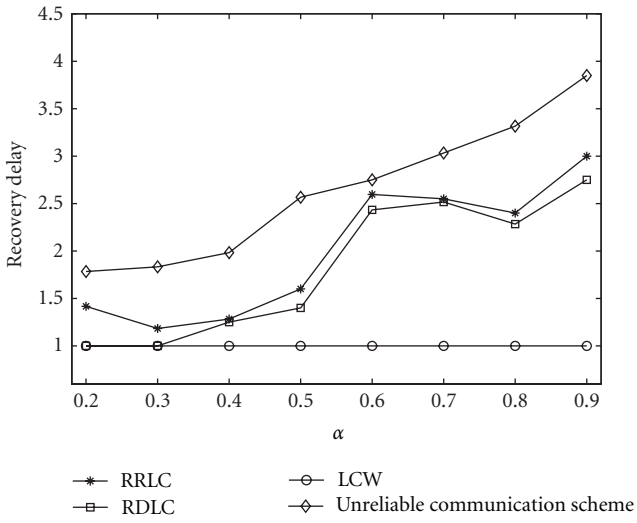
Figure 4(b) shows that the values of N also do not have impact on the recovery delay of the unreliable communication scheme and the *LCW* scheme. On the other hand, the recovery delays of the proposed *RDLC*, *RRLC* schemes increase with the increase of achievable throughput. However, by selecting a suitable value of N, a transmission can achieve a higher throughput with a lower delay compared with the unreliable communication scheme and the *LCW* scheme.

Finally, in Figure 5, we set $L = 16$, $\alpha = 0.4$, $f_{\max} = \lfloor \alpha L \rfloor$, $N = \lceil M - f_{\max}/2 \rceil - 1$.

In Figure 5(a), the throughput of the proposed *RDLC*, *RRLC* schemes, the unreliable communication scheme, and the *LCW* scheme do not change because the throughput of them does not related with the size of the finite field. However, as we showed in Section 4, the size of the finite field has impact on the decoding probability that the destination to recovery the original packets when using the *RRLC* scheme. Therefore, the increase of the size of the finite field will increase the decoding probability and reduce the recovery delay. The trends of the recovery delay of the *RRLC* scheme shown in Figure 5(a) are consistent with our

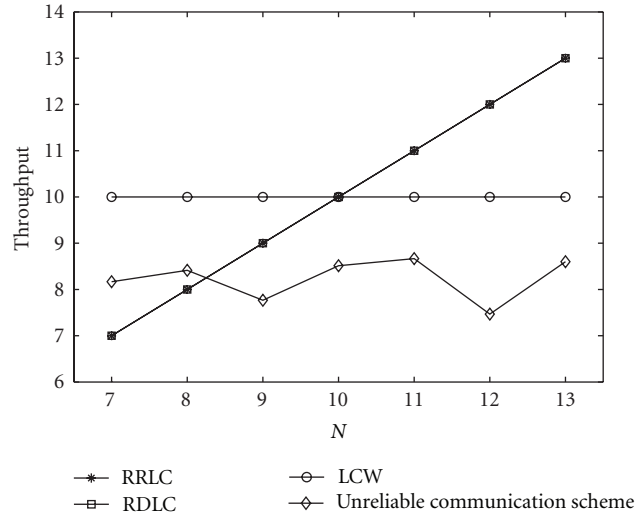


(a) Throughput

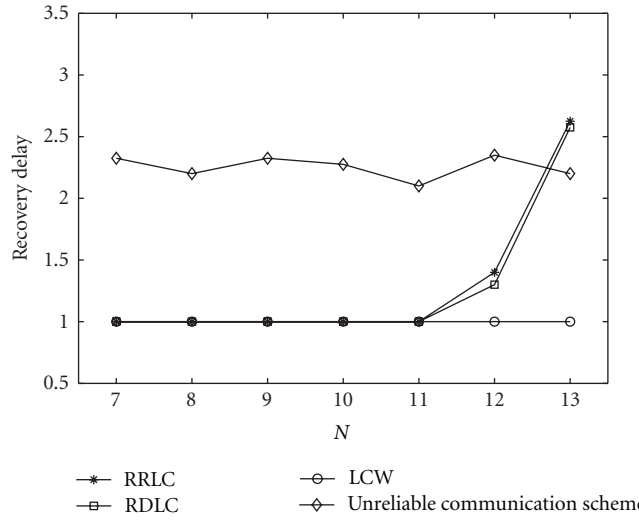


(b) Transmission delay

FIGURE 3: $L = 16$, $q = 2^2$, $f_{\max} = \lfloor \alpha L \rfloor$, $N = \lceil M - f_{\max}/2 \rceil - 1$.



(a) Throughput



(b) Transmission delay

FIGURE 4: $L = 16$, $q = 2^2$, $\alpha = 0.4$, $f_{\max} = \lfloor \alpha L \rfloor$, $N < \lceil M - f_{\max}/2 \rceil$.

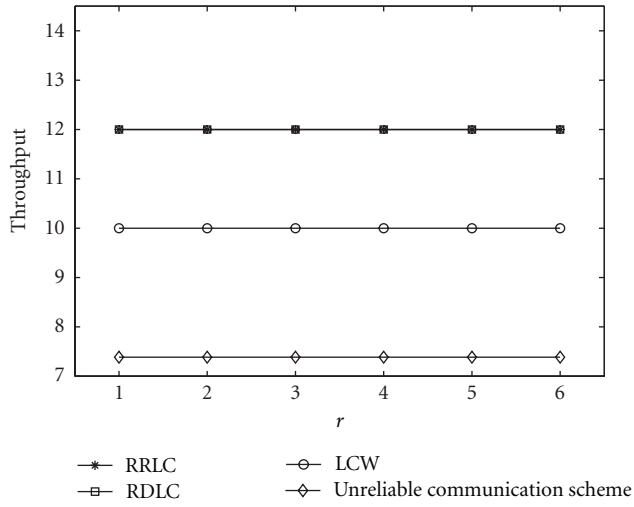
theoretical analysis. The recovery delay of the other three schemes does not related with the size of the finite field.

6. Conclusion

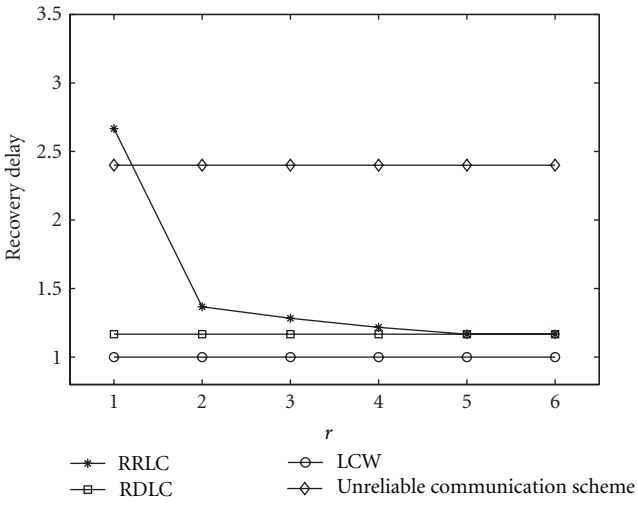
In this paper, we have explored the power of LNC to provide reliable communication for multiple failures in WSNs. The key idea is to design the LNC scheme based on average number of path failures, in which part of original data packets sent previously is still encoded into the following time slots, in order to improve the network throughput by utilizing the redundant coded data packets transmitted in the transmission rounds with fewer failures happening simultaneously in the WSN. Specifically, we first give the design of deterministic LNC scheme based on the average number of path failures, by which the network throughput is significantly improved comparing with the traditional approaches designed based on the maximum number of path failures. We also have investigated the behavior of the random LNC, when it is applied to the reliable communication problem studied in this paper. We have given the performance evaluation, which demonstrates the effectiveness of the proposed LNC schemes.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants no. 61202378, 61070169, 61070170, and 61201212, the Fundamental Research Funds for the Central Universities under Grant no. 2012HGBZ0640, the Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, the Specialized Research Foundation for the Doctoral Program of Higher Education of China no. 20103201110018, and the Application Foundation Research of Suzhou of China no. SYG201118, SYG201238, and SYG201239.



(a) Throughput



(b) Transmission delay

FIGURE 5: $L = 16$, $\alpha = 0.4$, $f_{\max} = \lfloor \alpha L \rfloor$, $N = \lceil M - f_{\max}/2 \rceil - 1$.

References

[1] I. F. Akyildiz and X. Wang, "A survey on wireless mesh networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23–S30, 2005.

[2] M. L. Sichitiu, "Wireless mesh networks: opportunities and challenges," in *Proceedings of the 6th World Wireless Congress (WWC '05)*, pp. 318–323, Palo Alto, Calif, USA, May 2005.

[3] L. M. L. Oliveira, A. F. De Sousa, and J. J. P. C. Rodrigues, "Routing and mobility approaches in IPv6 over LoWPAN mesh networks," *International Journal of Communication Systems*, vol. 24, no. 11, pp. 1445–1466, 2011.

[4] I. Bekmezci and F. Alagz, "Energy efficient, delay sensitive, fault tolerant wireless sensor network for military monitoring," *International Journal of Distributed Sensor Networks*, vol. 5, no. 6, pp. 729–747, 2009.

[5] N. Jain and D. P. Agrawal, "Current trends in wireless sensor network design," *International Journal of Distributed Sensor Networks*, vol. 1, no. 1, pp. 101–122, 2005.

[6] W. Ye, J. Heidemann, and D. Estrin, "A flexible and reliable radio communication stack on motes," Tech. Rep., USC Information Sciences Institute, 2002.

[7] A. Woo and D. E. Culler, "A transmission control scheme for media access in sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 221–235, New York, NY, USA, July 2001.

[8] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493–506, 2004.

[9] B. Kusy, C. Richter, W. Hu et al., "Radio diversity for reliable communication in WSNs," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '11)*, pp. 270–281, April 2011.

[10] S. Chalasani and V. Rajaravivarma, "Survivability in optical networks," in *proceedings of the 35th Southeastern Ymposium on System Theory*, pp. 6–10, 2003.

[11] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[12] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.

[13] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[14] Z. Li, B. Li, and L. C. Lau, "On achieving maximum multicast throughput in undirected networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2467–2485, 2006.

[15] 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)*, p. 442, July 2003.

[16] A. E. Kamal, "1+N protection in mesh networks using network coding over p-Cycles," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–6, December 2006.

[17] S. A. Aly, A. E. Kamal, and O. M. Al-Kofahi, "Network protection codes: providing self-healing in autonomic networks using network coding," *Computer Networks*, vol. 56, no. 1, pp. 99–111, 2012.

[18] O. M. Al-Kofahi and A. Kamal, "Network coding-based protection of many-to-one wireless flows," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 797–813, 2009.

[19] A. E. Kamal, "A generalized strategy for 1+N protection," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 5155–5159, May 2008.

[20] S. Dai, J. Wang, X. Zhang, and S. Li, "Network coding-based 1+N protection scheme in hybrid wireless-optical broadband access networks," in *Proceedings of the 6th International ICST Conference on Communications and Networking in China (CHINACOM '11)*, pp. 1013–1020, 2011.

[21] S. Dai, X. Zhang, J. Wang, and J. Wang, "An efficient coding scheme designed for n+k protection in wireless mesh networks," *IEEE Communications Letters*, vol. 16, no. 8, pp. 1266–1269, 2012.

[22] L. Lima, M. Médard, and J. Barros, "Random linear network coding: a free cipher?" in *proceedings of the IEEE International Symposium on Information Theory (ISIT '07)*, pp. 546–550, June 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 614206, 12 pages
doi:10.1155/2012/614206

Research Article

Network Coded Wireless Cooperative Multicast with Minimum Transmission Cost

Xiumin Wang,[1] Jin Wang,[2] and Shukui Zhang[2,3]

[1] *School of Computer and Information, Hefei University of Technology, Hefei 230009, China*
[2] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[3] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

Correspondence should be addressed to Jin Wang, wjin1985@suda.edu.cn

Received 27 July 2012; Accepted 6 September 2012

Academic Editor: Yong Sun

We study multicasting over wireless lossy links. Instead of downloading all the data from the source node, we allow the destination nodes themselves to locally exchange the packets, as local communication within a cluster achieves higher packet reception probability with less transmission cost. However, when shall we stop the transmission from the source node? If the source stops too early, the destination nodes locally cannot reconstruct all the original packets, while if the source stops too late, the benefit of cooperative data exchange cannot be fully exploited. In this paper, we propose a network coded hybrid source and cooperative exchange scheme to determine when to stop the source sending and start the exchange process, so as to minimize the total transmission cost. For the case when the clusters are predefined, we derive the expected total transmission cost with our hybrid scheme. Our theoretical results show that under a special condition, the source node should keep sending the packets until all the destinations get the complete information. For the case when the clusters are not predefined, we propose a cluster division algorithm such that the destination nodes within each cluster can conduct data exchange locally with energy efficiency. Finally, simulation results demonstrate the effectiveness of the proposed scheme.

1. Introduction

Over the past decades, wireless sensor networks have attracted a great deal of research attentions [1–3]. Once deployed, sensors are expected to operate for a long period of time, and it is impractical to reach these sensors physically. However, it is quite often necessary to update the software running on those sensors or add new functionalities to the sensors [4, 5], which needs to reliably multicast large data objects with energy efficiency [5, 6]. Particularly, in wireless lossy sensor networks, multicasting packets from a single source node is still a challenge problem due to the heterogeneous lossy links to the destination nodes. To satisfy all the destination nodes, the source node needs to keep sending until the destination node with the worst packet reception link successfully receives all the packets, which is inefficient for the source node.

Recently, cooperative data exchange [7–11] has become a promising approach to achieve the efficient data communications. Instead of downloading all the packets from the source node (e.g., the server) [12, 13], cooperative data exchange allows the destination nodes to cooperatively exchange their received packets among themselves, once the destination nodes collectively hold all the packets. Compared with pure source-dominated multicast, cooperative data exchange has two main benefits: (a) short-range communication among destination nodes is often more reliable and consumes less transmission cost, (b) the bandwidth saved at the source node can serve more other nodes in the system.

However, most of the existing works on cooperative data exchange problem never consider when to stop downloading the data from the source node. On the one hand, if the source node stops too early, the destination nodes themselves may be unable to collectively reconstruct the complete packets. On the other hand, if the source node stops too late, the benefit of cooperative data exchange cannot be fully utilized. In addition, although the average transmission cost with cooperative data exchange is lower than with source transmission, the sum of the transmission costs within multiple clusters may exceed the cost with pure source

sending. To sum up, for wireless multicasting, it is important for us to consider when to stop the source sending and start the cooperative data exchange, so as to reduce the total transmission cost or energy consumption [1–3, 14, 15].

So far, only the work in [16] studies the hybrid source transmission and cooperative data exchange for wireless multicasting, and it stops the source sending exactly once the destination nodes within each cluster collectively hold the complete information. Although the scheme in [16] performs well in reducing the transmission delay, it is still possible that the cooperative data exchange by multiple clusters may consume more transmission cost/energy than source-dominated scheme. As discussed above, the cooperative data exchange needs to be conducted separately in each cluster, and thus the sum of their transmission costs may exceed the transmission cost with pure source transmission. In addition, the predefined clusters in [16] might not perform well, as the destination nodes in some clusters may collect the complete information more quickly than the destinations in other clusters, which may stop the source transmission too late for the other clusters.

Recent work also shows that network coding [17–19] can improve the network throughput and reliability, especially for wireless lossy networks. Instead of sending/forwarding the original packets directly, network coding allows the source/transmitting node to linearly combine multiple packets together. With this approach, each transmitted packet has almost the same contribution in reconstructing the original packets, and hence improves wireless reliability [20]. Considering the benefits of network coding, we assume that the packets sent from the source node or the packets exchanged among the destination nodes are all linear encoded at the source/transmitting nodes before sending.

In this paper, given the heterogeneous link loss probability and the transmission cost of the source transmission and data exchange among the destinations, we aim to design a hybrid source and cooperative data exchange scheme to minimize the total transmission cost consumed during multicasting. For the case when the clusters are predefined, we determine when the source should stop sending the packets to the destination nodes and the cooperative data exchange should start. For the case when the clusters are not predefined, we consider how to divide the destinations into the clusters. The main contribution of the paper can be concluded as follows.

(i) We theoretically derive the expected total transmission cost required with traditional source-dominated scheme and our hybrid transmission scheme.

(ii) Our analysis shows that under a special condition, the source node should keep sending the packets until all the destinations get the complete information, so as to reduce the total transmission cost.

(iii) We also propose an efficient algorithm to determine how to group the destination nodes into the clusters, so as to make sure that the destination nodes within each cluster can collectively recover all the original packets with energy efficiency.

(iv) We compare the performance of the proposed scheme with some existing schemes. Simulation results show that the proposed scheme can significantly reduce the total transmission cost.

The rest of the papers are organized as follows. In Section 2, we introduce the background and some related works. The system model and the problem description are presented in Section 3. The expected total transmission costs with source-dominated and our scheme are discussed in Section 4. In Section 5, we consider how to group the destinations into clusters after receiving the sufficient number of packets from the source. The simulation results are presented in Section 6. We conclude the paper in Section 7.

2. Background and Related Work

In this section, we provide a brief introduction to the existing wireless network coded multicast scheme and summarize some related works.

2.1. Wireless Network Coding. Network coding was originally proposed in information theory [17] and recently has become a promising approach to improve the network performance in throughput [21, 22], reliability [19, 23], security [18, 24], and so forth. Instead of forwarding the original packets directly, network coding allows the source node/intermediate node to combine multiple packets together before sending it out. The work in [25] ensures that all the encoded packets generated by the same peer are linearly independent with a high probability, if we use linear network coding based on a sufficient large field size.

Network coding in wireless lossy networks was also considered in the literature. The work in [21] proposed a first wireless network coding architecture, named COPE. By exploiting the broadcast nature of wireless medium, each node stores the overheard packets for a while. When a node transmits a packet, it uses its knowledge of what its neighbors have overheard to perform opportunistic coding [21, 22]. After receiving an encoded packet, multiple neighbors can decode their wanted packets with their overheard packets. In other words, the sender or transmitting node can deliver "multiple packets" to different neighbors in a single transmission, which thus improves the throughput. The work in [19, 23] theoretically shows that network coding significantly reduces the expected number of retransmissions in lossy networks compared to traditional ARQ scheme. The work in [5] considers the impact of both wireless unreliable communication and sleep scheduling of sensor nodes and proposes a deterministic code design at the source node so as to accomplish the data dissemination process at the earliest time.

2.2. Reliable Multicast in Lossy Wireless Networks. Traditionally, wireless single-hop multicast mainly focuses on source-dominated transmission, where the source node keeps sending the packets until all the destination nodes in the system obtain the complete information. However, the performance of such a source-dominated transmission

degrades significantly when the packet reception probabilities of the destination nodes are heterogeneous. In this case, the source node cannot stop sending the packets until the destination node with the worst link state successfully gets the packets, even if all the other destination nodes received the packets in a much earlier time.

Recently, cooperative data exchange [7] among the users (e.g., mobile users) becomes one of the most promising approaches in designing efficient data transmissions. In cooperative data exchange, each client initially holds a subset of the packets and wants all the packets that others have. In the literature, it is assumed that there is a common communication channel among the users to receive/send the packets from/to all other users. Cooperative data exchange is to make sure that each client finally can get the complete packets by exchanging packets among themselves through the common channel. Compared with source-dominated transmissions, cooperative data exchange appears ti have two benefits: (1) short-range communications among the users are often more reliable and faster, (2) the bandwidth saved at the server can serve more other clients in the system.

Most recent works on cooperative data exchange mainly focus on how to minimize the total number of packets to be exchanged/transmitted among the users [7–9], or the total transmission cost consumed at the users [10, 11], such that all the users in the system can finally recover/receive the complete packets. Current works also show that network coding can reduce the number of transmissions or the total transmission cost required for cooperative data exchange process [7–11].

However, in the literature, most of the works either focus on pure source-dominated multicast or focus on cooperative data exchange by assuming that each user initially holds a subset of packets. The only work that considers the hybrid architecture of source transmission and data exchange transmission is in [16]. Specifically, the source node is set to stop sending the packets once the destinations in each cluster can collectively reconstruct the original packets, followed by the cooperative data exchange within each cluster. The numerical results show that with this hybrid transmission, the transmission delay of the multicast can be reduced compared with pure source-dominated multicast. However, the source node may stop too early, as the sum of the energy consumed by the cooperative data exchange within multiple clusters may consume more energy, which is inefficient for wireless sensor networks.

3. System Model and Problem Description

In this section, we first introduce the system model of our problem. Then, we discuss the problem to be solved in two different cases: with predefined clusters and without predefined clusters.

3.1. System Model. In this paper, we consider a multicast application, where a source node s needs to send n packets to m destination nodes in $D = \{d_1, d_2, \ldots, d_m\}$. Before transmitting the packets, the source node may generate more than n encoded packets over the original n packets. It is typical to assume that with network coding, after receiving any n encoded packets, the destination node can recover the original n packets.

Suppose that the destination nodes are formed into multiple clusters in $\mathcal{C} = \{C_1, C_2, \ldots\}$, where $C_k \in \mathcal{C}$ denotes the set of destination nodes in cluster k. Without loss of generality, we assume that each pair of clusters maintain disjoint destination nodes, that is, $C_{k_1} \bigcap C_{k_2} = \emptyset$, for all $k_1 \neq k_2$ and $\bigcup_{C_k \in \mathcal{C}} C_k = D$. As in cooperative data exchange, we assume that local data exchange within each cluster is conducted through a common channel.

Our multicasting process consists of two transmission stages. In the first stage, the source node sends the packets to all the destination nodes in D. In the second stage, the destination nodes within the same cluster perform the cooperative data exchange among themselves. These two stages need to make sure that each destination finally can reconstruct the original n packets. Due to unreliable wireless communications, suppose that the packet loss probability from the source node to every destination node is l_s, and the packet loss probability between the destination nodes in cluster C_k is l_k. We also assume that the transmission cost of the source node is t_s, and the transmission cost for the data exchange within cluster C_k is $t(C_k)$. Without loss of generality, we assume that the transmission cost of the source node is higher than the transmission cost of the destination node, that is, $t_s \geq t_{C_k}$ for $\forall k$, and the reception probability of the packet sent from s is lower than the reception probability of the packet exchanged within the same cluster, that is, $l_s \geq l_k$.

In this paper, we aim to minimize the total transmission cost consumed during transmissions, while we ensure that all the destination nodes in D can successfully reconstruct the original n packets. Let x_{hs} be the number of packets that the source node sends, and x_k be the number of packets sent by the destination nodes within the cluster C_k. Thus, the total transmission cost can be written as

$$x_{hs} t_s + \sum_{c_k \in \mathcal{C}} x_k t(C_k) \tag{1}$$

which needs to be minimized.

An example of the system model is shown in Figure 1. The source node s needs to send three packets to the destinations in $D = \{d_1, d_2, \ldots, d_7\}$, and the original packets are generated into three encoded packets p_1, p_2, and p_3. The set of the packets received at each destination is given near the node after the source node sends three packets p_1, p_2, p_3, for example, the set of packets received by node d_4 is $\{p_3\}$. In the next subsections, we will discuss the problems to be considered in this example.

3.2. Problem Description with Predefined Clusters. In this section, we consider the case when the destination nodes in each cluster are predetermined.

Due to unreliable wireless communications, some packets may be lost at some destination nodes. So, one challenge problem is to determine when the source node s stops

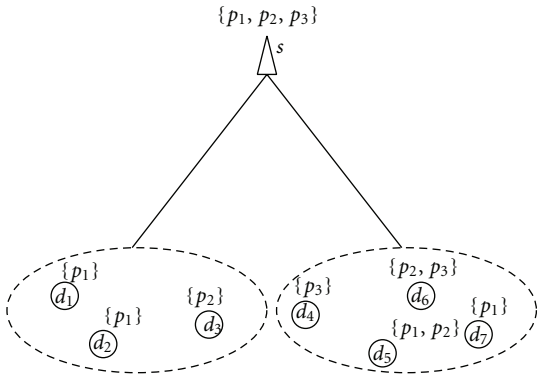


Figure 1: An example of system model after sending three packets in $\{p_1, p_2, p_3\}$ from source node s to receiver nodes in $D = \{d_1, d_2, \ldots, d_7\}$.

sending the packets such that the destination nodes within each cluster can exchange the packets among themselves, and the total transmission cost defined in (1) is minimum.

(i) If the source node s stops too early, the destination nodes in some clusters may be unable to collectively recover all the original packets.

(ii) If the source node s stops too late, the low-cost transmission among the destination nodes themselves cannot be fully utilized, which may incur high transmission cost because of high packet loss probability and high transmission cost of the source transmission.

(iii) The sum of the transmission costs by the data exchange by all the clusters may exceed the transmission cost by the pure source transmission. In this case, the source node should keep sending until all the destination nodes obtain the complete information.

As shown in Figure 1, if the source node s stops after sending p_1, p_2, p_3, the destination nodes in the left cluster (within a circle) cannot collectively reconstruct the original three packets. On contrary, if the source node s sends too many packets, all the destination nodes within each cluster may obtain the complete packets, which does not utilize the low-cost cooperative data exchange. However, if the sum of the transmission costs in these two clusters are higher than the transmission cost from the source node, it is better to let the source node keep sending until all the destination nodes get the complete information.

We refer to the above problem of determining when to stop the source transmission, as minimizing the multicast transmission cost problem with predefined clusters.

3.3. Problem Description with Nonpredefined Clusters. In the above section, we assume that the destination nodes in each cluster are given. However, the predefined clusters may not perform well, as the destination nodes in some clusters may collectively hold the complete information much later than the destination nodes in other clusters, which thus delays the starting time of the cooperative data exchange in other clusters.

Still take Figure 1 as an example, if we group the destinations in $\{d_1, d_2, \ldots, d_7\}$ into two clusters, that is, the destinations in the same circle belong to the same cluster. After the source node s sends three packets p_1, p_2, p_3, the destination nodes in the left cluster cannot recover the original packets, as they collectively have received only two packets so far. However, if we move destination node d_4 into the left cluster, the destination nodes in both clusters can reconstruct all the three original packets, and then s can stop sending the packets.

Thus, after the source node sends sufficient number of packets, we need to consider how to group the destination nodes into multiple clusters such that the destination nodes within each cluster collectively can reconstruct the original packets with energy efficiency. In this case, the number of clusters in $\mathcal{C}$ and the set of the destinations in each cluster C_k need to be determined by our algorithm.

Table 1: Main notations and their descriptions.

$\mathcal{C}$	The set of the clusters
C_k	The kth cluster in $\mathcal{C}$
D	The set of the destination nodes in the system
d_i	The ith destination node in D
l_s	Packet loss probability on the link from s to each destination
l_k	Packet loss probability of the data exchange within cluster C_k
m	The total number of destination nodes in D
n	The total number of original packets
s	The source node
t_s	The transmission cost of the source node s
$t(C_k)$	The transmission cost of the data exchange within cluster C_k
x_{hs}	Number of packets sent by s in the first stage of hybrid scheme
x_k	The number of packets exchanged within cluster C_k

4. Minimum Multicast Transmission Cost with Predefined Clusters

In this section, we consider the case when the clusters are predefined. We first analyze the expected total transmission costs with source-dominated scheme and our hybrid transmission scheme. We then formulate the problem of minimizing the total transmission cost with our hybrid transmission scheme as an integer programming.

To ease the understanding, the main notations used in the paper are given in Table 1.

4.1. Transmission Cost with Source-Dominated Transmission. With source-dominated scheme, the source node s keeps sending the encoded packets that are generated based on n original packets, until all the destination nodes in D

```
begin
    //after sending x_hs packets from the source node s
    if the destinations in D cannot recover the original packets
    then
        The source s should send more packets before the
        second stage;
    end
    else
        U = D;
        C_1 = ∅;
        not_finish=1;
        k = 1;
        while not_finish do
            find a destination d_i ∈ U which satisfies the rules
            (1), (2), and (3) and incurs the least cost to C_k;
            if There does not exist a feasible d_i ∈ U then
                C_k = C_k ⋃ U;
                not_finish=0;
            end
            else
                add d_i into C_k;
                delete d_i from U;
                if If the destinations in C_k can collectively
                recover the original packets then
                    k = k + 1;
                    C_k = ∅;
                end
            end
        end
    end
end
```

Algorithm 1: Algorithm design for cluster determination.

successfully decode the original n packets. With linear network coding, after receiving any n encoded packets from s, the destination node can decode the original n packets.

Suppose that $\overline{x_s}$ is the expected transmission cost required with the source-dominated scheme. To make sure all the destinations can recover the complete packets, the number of packets that s sends should be at least the maximum number of packets to be sent to satisfy the decoding requirement of each destination. That is

$$\overline{x_s} = \max\left\{\overline{x_1'}, \overline{x_2'}, \ldots, \overline{x_m'}\right\}, \tag{2}$$

where $\overline{x_i'}$ means the expected number of packets required to be sent by s, so as to guarantee that destination d_i can decode all the n original packets. Since the packet loss probability from s to each destination node is l_s, we can derive $\overline{x_i'}$ as follows:

$$\overline{x_i'} = \frac{n}{1 - l_s}. \tag{3}$$

In other words, we have

$$\overline{x_s} = \max\left\{\overline{x_i'} \mid \forall i\right\} = \frac{n}{1 - l_s}. \tag{4}$$

With the above equations, we can obtain the expected transmission cost as follows:

$$\overline{x_s} t_s = \frac{n t_s}{1 - l_s}. \tag{5}$$

4.2. Transmission Cost with Hybrid Transmission Scheme. With hybrid transmission scheme, there are two stages. In the first stage, the source node s sends x_{hs} encoded packets. In the second stage, the destination nodes in the same cluster cooperatively exchange the packets that they received before, until each of them can decode all the packets.

We now consider the first stage. For the success of the second stage, the number of packets sent by the source s should make sure that the destination nodes within each cluster can collectively reconstruct all the original packets. The probability that at least one of the destinations in cluster C_k can receive the current transmission is $1 - l_k^{|C_k|}$. Thus, to guarantee at least n transmissions of the packets are successfully received by the collection of the destination nodes in C_k, the expected number of packets sent by the source node s, denoted as $\overline{x_{hs}}$, should satisfy the following condition:

$$\overline{x_{hs}} \geq \frac{n}{1 - l_k^{|C_k|}}. \tag{6}$$

When considering all the clusters, (6) can be written as

$$\overline{x_{hs}} \geq \max_{C_k \in \mathcal{C}} \left\{ \frac{n}{1 - l_k^{|C_k|}} \right\}. \tag{7}$$

Note that if we set $\overline{x_{hs}} = \min_{C_k \in \mathcal{C}} \{n/(1 - l_k^{|C_k|})\}$, it means the source node s stops sending the packets exactly once the destination nodes in each cluster collectively can recover the complete packets.

We then consider the second stage. After the source node sends $\overline{x_{hs}}$ packets, the expected number of packets required by each destination in cluster C_k is

$$\overline{L} = n - \overline{x_{hs}}(1 - l_s). \tag{8}$$

Then, the expected number of packets to be sent by other destinations through cooperative data exchange, denoted as $\overline{M_k}$, such that one specific destination in cluster C_k can decode, is

$$\overline{M_k} = \frac{\overline{L}}{1 - l_k}. \tag{9}$$

According to the work in [16], the expected number of packets required to be exchanged among cluster C_k such that all the destination nodes in C_k can obtain its required $\overline{M_k}$ packets is

$$\overline{x_k} = \frac{|C_k|}{|C_k| - 1} \left(\overline{M_k} + \overline{W}_{|C_k|} \sqrt{\sigma^2_{\overline{M_k}}} \right), \tag{10}$$

where $\overline{W}_{|C_k|}$ is the expectation of the $|C_k|$-th order statistic of a sequence of $|C_k|$ normal random variables, and $\sigma^2_{\overline{M_k}}$ (given in [16]) is the variance of $\overline{M_k}$.

Thus, the total transmission cost with hybrid transmission scheme can be formulated as follows:

$$\overline{x_{hs}} t_s + \sum_{C_k \in \mathcal{C}} t(C_k) \overline{x_k}. \tag{11}$$

If we assume that the packet loss probability within each cluster is the same, that is, $l_k = l_r$ for for all $C_k \in \mathcal{C}$, and the transmission cost within each cluster is also the same, that is, $t(C_k) = t_r$ for $\forall C_k \in \mathcal{C}$, we can obtain the following theorem.

Theorem 1. *If $|\mathcal{C}| t_r(1 - l_s) \geq t_s(1 - l_r)$, to minimize the expected total transmission cost, it is better to let the source node keep sending the packets until all the destination nodes in D successfully obtain the original n packets, that is, $x_k = 0$.*

Proof. Let $\overline{x_{hs}}$ be the number of packets sent by the source node s in a hybrid transmission scheme. Since we assume that $|C| t_r(1 - l_s) \geq t_s(1 - l_r)$, we have

$$t_r \geq \frac{t_s(1 - l_r)}{|C|(1 - l_s)}. \tag{12}$$

Based on the above equation (12), we then compare the expected total transmission cost required by source-dominated scheme with that by the hybrid transmission scheme for any $x_k > 0$ as follows:

$$\begin{aligned}
&\overline{x_{hs}} t_s + \sum_{C_k \in \mathcal{C}} t(C_k) \overline{x_k} - \overline{x_s} t_s \\
&= \overline{x_{hs}} t_s + \sum_{C_k \in \mathcal{C}} t(C_k) \overline{x_k} - \frac{n t_s}{1 - l_s} \\
&\geq \overline{x_{hs}} t_s + |C| t_r \overline{M_k} - \frac{n t_s}{1 - l_s} \\
&= \overline{x_{hs}} t_s + |C| t_r \frac{n - \overline{x_{hs}}(1 - l_s)}{1 - l_r} - \frac{n t_s}{1 - l_s} \\
&\geq \overline{x_{hs}} t_s + |C| \frac{n - \overline{x_{hs}}(1 - l_s)}{1 - l_r} \frac{t_s(1 - l_r)}{|C|(1 - l_s)} - \frac{n t_s}{1 - l_s} \\
&= \overline{x_{hs}} t_s + \frac{(n - \overline{x_{hs}}(1 - l_s)) t_s}{1 - l_s} - \frac{n t_s}{1 - l_s} \\
&= \frac{\overline{x_{hs}} t_s (1 - l_s) + (n - \overline{x_{hs}}(1 - l_s)) t_s - n t_s}{1 - l_s} \\
&= 0.
\end{aligned} \tag{13}$$

From the above equation, we can see that the expected total transmission cost with hybrid scheme is not lower than source-dominated transmission scheme when $|C| t_r(1 - l_s) \geq t_s(1 - l_r)$. In other words, to reduce the total transmission cost, the source node should keep sending until all the destination nodes can successfully decode the complete information, that is, $x_k = 0$, which thus proves the theorem. □

4.3. Hybrid Transmission Scheme with Minimum Total Transmission Cost. As discussed above, in some cases, the sum of the transmission costs within all the clusters may be more than transmission cost of the source transmission. Under such circumstances, the source node should keep sending the packets until all the destination nodes get the complete information, which is a special case of our hybrid scheme, that is, $x_{hs} = x_s$.

To determine x_{hs}, the problem of minimizing the total transmission cost with hybrid transmission scheme can be formulated as follows:

$$\min \overline{x_s} t_s + \sum_{C_k \in \mathcal{C}} t(C_k) \overline{x_k} \tag{14}$$

subject to (7).

With the above integer formulation, we can obtain the best time to stop the source transmission and start the cooperative data exchange process.

5. Cluster Determination in Cooperative Data Exchange Stage

As discussed in Section 3.3, after the source transmissions in the first stage, the destination nodes in some clusters may

be unable to reconstruct the original packets if the clusters are predefined. In this section, we consider after the source node sends x_{hs} packets in the first stage, where $x_{hs} \geq n$, how to group the destination nodes into multiple clusters so as to make sure that the destination nodes in each cluster collectively can recover all the original packets.

Assume that after the source node sends x_{hs} packets, the set of the packets received by destination node d_i is denoted as H_i, for example, $H_6 = \{p_2, p_3\}$ in Figure 1. Since the total number of packets sent by the source node s in the first stage is x_{hs}, we have $H_i \subseteq \{p_1, p_2, \ldots, p_{x_{hs}}\}$. Before describing how to group the clusters, we first discuss given the exact packet reception states of the destination nodes, how many transmissions are required by the data exchange process within a local cluster.

5.1. Number of Transmissions Required for Data Exchange Process within a Local Cluster. We now discuss the number of transmissions required for data exchange within a specific cluster C_k, given the exact packet reception sets of the destination nodes in C_k.

Without loss of generality, we assume that after the source s sends x_{hs} packets, all the destination nodes in cluster C_k collectively can recover the original n packets. As in the above section, we assume that the packets sent by the source node s are linear independent with each other, that is, after receiving any n packets from s, the node can decode the original n packets. In other words, after the first stage, the total number of packets collectively received by the destination nodes within each cluster C_K should satisfy

$$\left| \bigcup_{d_i \in C_k} H_i \right| \geq n. \tag{15}$$

Before describing the number of transmissions for data exchange process within cluster C_k, we first introduce a useful existing result in [26].

Theorem 2 (see [26]). *Provided that the encoding field size is large enough, the minimum number of packets to be exchanged is given by*

$$x = \operatorname{rank}\left(\bigcup_{d_i \in C_k} H_i \right) - \min_P \frac{S_{R(P)} - \operatorname{rank}(\cup_{d_i \in C_k} H_i)}{|P| - 1}, \tag{16}$$

where $P = \{S_1, S_2, \ldots, S_{|P|}\}$ *denotes a disjoint partition of the node sets and* $S_j \in P$ *is the subset of the nodes in* C_k*, where* $2 \leq P \leq |C_k|$*, and* $\operatorname{rank}(\bigcup_{d_i \in C_k} H_i)$ *means the rank of the matrix including the encoding vectors of the packets in* $\bigcup_{d_i \in C_k} H_i$*,* $S\_R(P) = \sum_{S_j \in P} \operatorname{rank}(\bigcup_{d_i \in S_j} H_i)$.

Note that in our problem, the packets sent by the source node s are linear independent with each other. Thus, fora given cluster C_k, we can obtain that

$$\begin{aligned} \operatorname{rank}\left(\bigcup_{d_i \in C_k} H_i \right) &= n, \\ S_R(P) &= \sum_{S_j \in P} \operatorname{rank}\left(\bigcup_{d_i \in S_j} H_i \right) \\ &= \sum_{S_j \in P} \min\left\{ \left| \bigcup_{d_i \in S_j} H_i \right|, n \right\}. \end{aligned} \tag{17}$$

Given the packets received by the destination nodes in C_k, according to Theorem 2, we can then derive the number of packets to be exchanged among the destinations in C_k as follows:

$$x_k = n - \min_P \frac{\sum_{S_j \in P} \min\left\{ \left| \cup_{d_i \in S_j} H_i \right|, n \right\} - n}{|P| - 1}, \tag{18}$$

where P is a disjoint partition of the nodes in C_k and $S_j \in P$ is the jth subset of the nodes by the partition P.

5.2. Optimal Cluster Division with Minimum Transmission Cost in the Data Exchange Process. We now consider how to divide all the destination nodes into multiple clusters, so as to minimize the total transmission cost in the second stage.

For m destination nodes in D, the maximum number of clusters that can be formed is m. The problem of minimizing the total transmission cost of the data exchange within the clusters by cluster division can be formulated as follows:

$$\min_{C_k} \sum_{k=1}^{m} t(C_k) x_k \tag{19}$$

subject to

$$C_k \bigcap C_{k'} = \varnothing, \tag{20}$$

$$\bigcup_{k=1}^{m} C_k = D, \tag{21}$$

$$\left| \bigcup_{d_i \in C_k} H_i \right| = n. \tag{22}$$

In (19), $t(C_k)$ means the transmission cost if the destination nodes in C_k form a cluster, and x_k is number of packets to be sent by the data exchange within cluster C_k, which can be calculated with (18). Thus, the objective is to minimize the total transmission cost of the data exchange within all the clusters, by determining C_k. The constraints in (20) and (21) denote that each destination node should be located in one and only one cluster. The constraint in (22) represents that the destination nodes can form a cluster if and only if these destination nodes collectively can recover all the n original packets.

Although the above formulation can derive the optimal cluster division to minimize the total transmission cost in

the data exchange process, the complexity of calculating x_k is too high, as we need to enumerate all the possible partitions of the nodes in C_k, $\{P\}$. It makes the formulation difficult to be solved. Thus, in the next subsection, we propose a suboptimal algorithm to divide the clusters.

5.3. Algorithm Design. In this section, given the packet reception state of each destination node, we consider to group all the destination nodes into multiple clusters.

Without loss of generality, we suppose that after x_{hs} transmissions from source node s, all the destinations collectively can recover the original n packets. Otherwise, the source node s should send more packets before starting the second stage (i.e., cooperative data exchange). Let U be the set of all the destinations left in the system, that is, $U = D$.

We then introduce how to select the destinations in U into the kth cluster C_k. When adding a new destination d_i from U into C_k, we make the following rules.

(1) The destination d_i should be able to contribute at least one "innovative" packet to the destination nodes that have been added to C_k so far.

(2) Deleting d_i from U will not sacrifice the decoding capability of the destinations left in U, that is, (22).

(3) The sum of the transmission costs of the new cluster C_k (including d_i) and the cluster formed by the left destinations in U (excluding d_i) is smaller than the transmission cost of the cluster formed by $U \bigcup C_k$.

We then describe the process of adding new destinations to C_k as follows.

(i) We start from the destination node $d_i \in U$ that satisfies the above three rules and incurs the least transmission cost if it is added into C_k. If there does not exist such a node in U, we can obtain that the destinations in U can only form one cluster, that is, $C_k = U$. In this case, the algorithm terminates.

(ii) If we can find a feasible destination d_i, we add d_i into C_k and correspondingly delete d_i from U. We then check if the destinations in C_k can collectively recover all the original packets. If they can, we finish the cluster C_k and repeat the above process to find the $(k+1)$th cluster C_{k+1} from U.

(iii) If they still cannot recover all the original packets, we continue the above two steps.

(iv) The algorithm continues until we cannot find a feasible node $d_i \in U$ that satisfies the rules (1), (2), and (3). In this case, the destination nodes left in U should be all added to the current considered cluster C_k, that is, $C_k = C_k \bigcup U$.

The detailed process of the above algorithm is shown in Algorithm 1. With the above algorithm, we can make sure that the destination nodes within each cluster $C_k \in C$ collectively can reconstruct all the original packets.

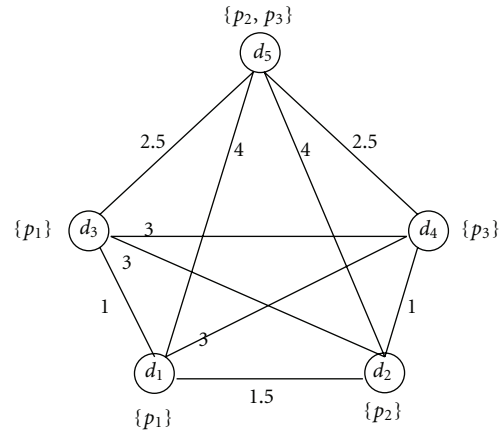


Figure 2: An example of five destination nodes.

5.4. Illustration Example. We take Figure 2 as an example to illustrate how Algorithm 1 works, where five destination nodes in $\{d_1, d_2, \ldots, d_5\}$ need to get three original packets. Assume that the first stage stops after the source node s sends three packets $\{p_1, p_2, p_3\}$. The set of the packets received by each destination in the first stage is given near the node. To simplify the understanding, we define the transmission cost of the transmission within a cluster C_k as the square of the maximum distance between every two destinations within the same cluster [25, 27], that is,

$$t(C_k) = \max_{d_i, d_{i'} \in C_k} \left\{ |d_i - d_{i'}|^2 \right\}, \tag{23}$$

where $|d_i - d_{i'}|$ denotes the distance between two nodes d_i and $d_{i'}$ (given on the edge between two nodes). Note that the above model is only an example, and our algorithm does not restrict the model of the transmission cost.

According to the Algorithm 1, initially we have $U = \{d_1, d_2, \ldots, d_5\}$. We can easily check that if all the nodes in U form only one cluster, its transmission cost is $4^2 = 16$.

We first construct the cluster C_1, by starting from any node d_1. Since d_1 satisfies all the three rules defined in the above section, we add d_1 into C_1 and delete it from U, that is, $C_1 = \{d_1\}, U = \{d_2, d_3, \ldots, d_5\}$. Since the nodes in C_1 cannot reconstruct the original three packets, we need to add more nodes to C_1. Note that although adding d_3 incurs the least transmission cost of C_1, d_3 violates the rule (1), as the packet it has is not innovative to the nodes in C_1. We then find that d_2 can be added to C_1 and it incurs the least transmission cost among the nodes left in $U \setminus \{d_3\}$, that is, $C_1 = \{d_1, d_2\}, U = \{d_3, d_4, d_5\}$. With a similar approach, d_4 will be added to C_1, that is, $C_1 = \{d_1, d_2, d_4\}, U = \{d_3, d_5\}$. Since the destinations in C_1 now can collectively reconstruct the original three packets, the construction of cluster C_1 terminates, and the transmission cost of C_1 is $t(C_1) = 3^2 = 9$.

We then construct the cluster C_2. We can easily check that any node in $U = \{d_3, d_5\}$ cannot satisfy the rule (2). Thus, the destination nodes in U cannot be divided into multiple clusters. In other words, all the destinations in U are added into C_2, that is, $C_2 = \{d_3, d_5\}$, and the

transmission cost of C_2 is $t(C_2) = 2.5^2 = 6.25$. With the above operation, the destination nodes in D form two clusters, $C_1 = \{d_1, d_2, d_4\}, C_2 = \{d_3, d_5\}$. Note that, the algorithm can start from any node, for example, if starting from node d_2, we can also get two clusters $\{d_2, d_4, d_1\}$, $\{d_3, d_5\}$.

6. Simulation Results

In the simulation, we study a connected network graph, where nodes are randomly deployed in a two-dimensional (2D) space. We use l_s to simulate the packet loss probability on the link from the source node s to the destination nodes, and l_k to be the packet loss probability by the data exchange within cluster C_k. For the transmission cost, we use t_s and $t(C_k)$ to denote the transmission cost of the packet sent from source node and the packet exchanged within cluster C_k, respectively. Generally, we set $l_s \geq l_k$ and $t_s \geq t(C_k)$.

To demonstrate the performance of our proposed scheme, we also conduct two baseline algorithms: source-dominated scheme, and the hybrid scheme by [16]. As introduced before, in source-dominated scheme, the source node keeps sending until all the destination nodes get the complete packets. The difference between the hybrid scheme proposed by [16] and ours is that the scheme in [16] stops the source sending exactly once the destination nodes in each cluster can collectively reconstruct the full packets.

We define cost gain, denoted as δ, as the ratio of the transmission cost difference with source-dominated scheme and the hybrid scheme (our hybrid scheme or the scheme in [16]) to the total transmission cost with source-dominated scheme, for example, the cost gain by our hybrid scheme can be written as

$$\delta = \frac{x_s t_s - \left(x_{hs} t_s + \sum_{C_k \in \mathcal{C}} x(C_k) t_k\right)}{x_s t_s}. \tag{24}$$

6.1. The Impact of the Transmission Cost. In this section, we conduct the simulation to investigate the impact of the transmission cost on the cost gain. In this setting, we set $|\mathcal{C}| = 2$, $l_s = 0.5, l_k = 0.2, t_s = 6, |C_k| = 5$ and vary the transmission cost of the data exchange within each cluster in $[1, 6]$.

As shown in Figure 3, the cost gain with our hybrid scheme is much better than the scheme proposed by [16]. We can observe that in some cases, for example, $t(C_k) \geq 3$, the total transmission cost with the scheme [16] is even more than source-dominated scheme. This is because, when the transmission cost of the packet exchange within a cluster is high, the sum of the transmission costs in multiple clusters exceeds the cost consumed by pure source sending. In other words, the hybrid scheme in [16] stops the source transmission too early. From the figure, we can also find that when $t(C_k)$ is no more than 3, the transmission cost with our hybrid scheme is better than the source-dominated scheme. However, when $t(C_k) \geq 4$, it is the same as the source-dominated scheme, which verifies the results of Theorem 1. In this case, the programming used in our hybrid scheme can detect and keep the source sending until all the nodes receive the complete packets.

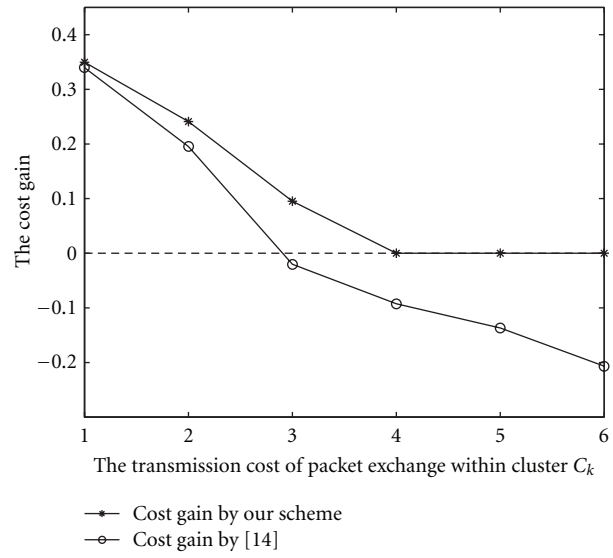


Figure 3: The impact of transmission cost of each cluster on the transmission cost gain ratio.

From Figure 3, we can also observe that with the increase of the transmission cost within a cluster, the cost gain decreases. This is reasonable, as when the transmission cost within a cluster increases, the sum of the total transmission costs of multiple clusters increases quickly and correspondingly reduces the gain.

6.2. The Impact of Packet Loss Probability. We now investigate the impact of packet loss probability on the performance of the total transmission cost. We fix $n = 100, l_k = 0.2, |\mathcal{C}| = 2, t_s = 6, t(C_k) = 3$ and vary the packet loss probability on the link from the source to the destination in $[0.4, 0.9]$.

As shown in Figure 4, the cost gain with our hybrid scheme is better than with the scheme in [16], and our hybrid scheme also always consumes less transmission cost compared with source-dominated transmission scheme. From the figure, we observe that with the increase of the packet loss probability from the source to the destination nodes, the transmission cost gains with both hybrid schemes increase. This is because, when the links from the source to the destinations nodes are too bad, the source node should stop sending the packet as early as possible.

In addition, when the packet loss probability $l_s \geq 0.8$, the cost gains with both hybrid schemes are almost the same, as our hybrid scheme may stop the source transmission at almost the same time as in [16], that is, after the destination nodes in each cluster can reconstruct the complete packets.

6.3. The Impact of the Number of Clusters. We now conduct the simulation to investigate the impact of the number of the clusters on the total transmission cost of the proposed schemes. In this setting, we fix $n = 100, l_s = 0.6, l_k = 0.2, t_s = 6, t(C_k) = 2$ and the number of clusters are varied from 1 to 7.

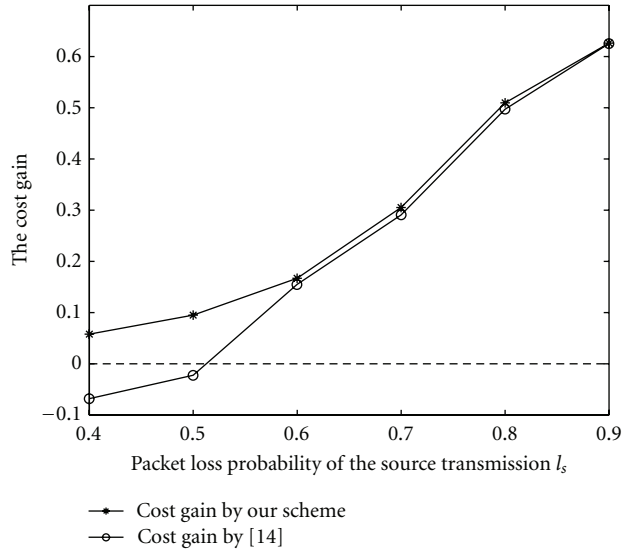


Figure 4: The impact of packet loss probability on the transmission cost gain ratio.

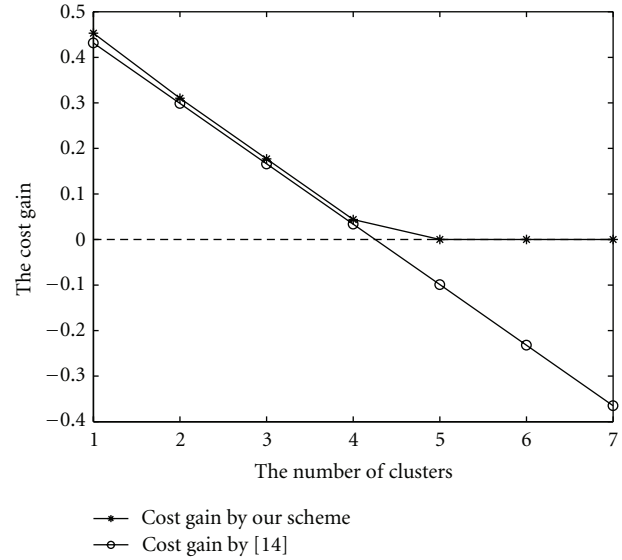
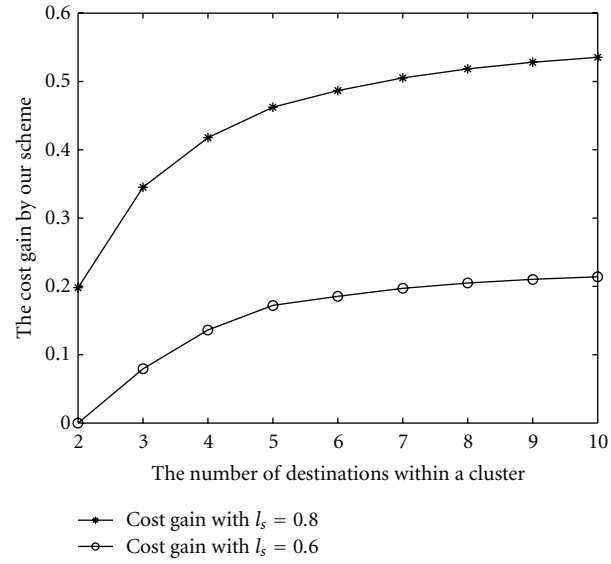
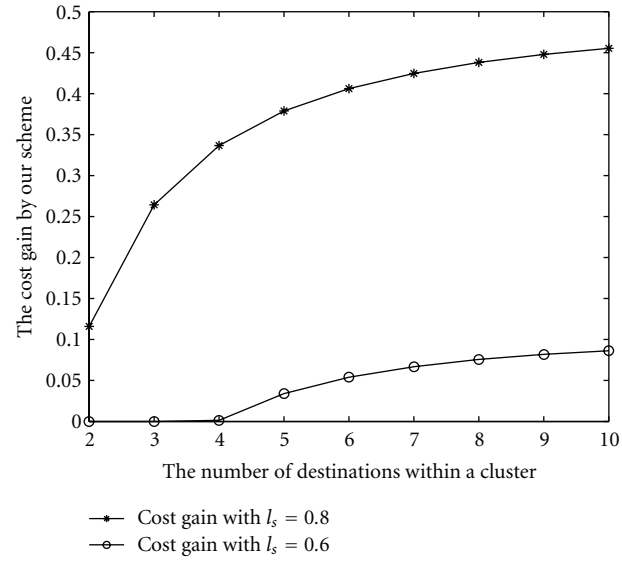


Figure 5: The impact of the number of clusters on the transmission cost gain ratio.



(a) $|\mathcal{C}| = 3$



(b) $|\mathcal{C}| = 4$

Figure 6: The impact of the number of destinations within the cluster on cost gain by our scheme.

As shown in Figure 5, when the number of clusters is few, the total transmission costs with both hybrid schemes are less than with the source-dominated scheme. This is because, when the number of clusters is few, the sum of their transmission costs does not exceed the source transmission cost. However, when the number of clusters increases, for example, more than 5, pure source transmission is better, since the sum of the transmission costs in the clusters is more than the transmission cost from the source. However, we can observe that, the transmission cost with our scheme is not higher than with source-dominated scheme in all the cases, as our hybrid scheme always chooses the best time to stop the source transmission.

6.4. The Impact of the Number of Receiver Nodes within Clusters. Finally, we investigate the impact of the number of destination nodes within each cluster on the performance of our transmission scheme.

As shown in Figure 6, we fix $n = 100, t_s = 6, t(C_k) = 2, l_k = 0.2$ and vary the number of destination nodes within each cluster in $[2, 10]$. From the figure, we can see that with the increase of the number of the destination nodes within each cluster, the cost gain with our hybrid scheme increases. This is because, with more destination nodes in a cluster, it is much more earlier that the destinations in the cluster collectively can reconstruct the original n packets. Thus, the second stage of our hybrid scheme, data exchange process,

can start earlier, which decreases the number of high-cost transmissions from the source node s.

From Figure 6, we also observe that the cost gain with the setting $l_s = 0.8$ is much higher than with the setting $l_s = 0.6$. The reason is that with source-dominated scheme, a large number of the transmissions will be wasted when the packet reception state on the links from s to the destination nodes is bad. In this case, the cooperative data exchange process in our hybrid scheme should start as early as possible, as it consumes less transmission cost. In other words, our hybrid scheme performs well especially when the packet reception from the source node is bad.

By comparing Figures 6(a) and 6(b), we can also obtain that when the number of clusters in the network increases, the cost gain of our hybrid scheme decreases. As each cluster consumes its independent transmission cost in the cooperative data exchange process, the sum of the transmission costs of the data exchange by all the clusters increases, which thus decreases the cost gain.

7. Conclusion

In this paper, we proposed a hybrid source and cooperative data exchange transmission scheme for reliable multicasting over wireless lossy links. Our hybrid scheme determines when to stop the source sending and start the cooperative data exchange, so as to minimize the total transmission cost. We theoretically derive the total transmission cost required with traditional source-dominated scheme and our hybrid scheme. We give a condition under which the source node should not stop sending the packets until all the destination nodes successfully get the complete information, so as to reduce the total transmission cost. If the clusters are predefined, we propose an efficient algorithm to divide the destination nodes into multiple clusters, such that the destination nodes within each cluster can conduct the cooperative data exchange separately with energy efficiency. Finally, simulation results demonstrate the effectiveness of the proposed scheme in reducing the total transmission cost.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities (no. 2012HGBZ0640). It was also supported in part by National Natural Science Foundation of China under Grants no. 61202378, 61070169, Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, Specialized Research Foundation for the Doctoral Program of Higher Education of China no. 20103201110018, and Application Foundation Research of Suzhou of China no. SYG201118.

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] H. M. N. D. Bandara, A. P. Jayasumana, and T. H. Illangasekare, "A top-down clustering and cluster-tree-based routing scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, Article ID 940751, 17 pages, 2011.

[3] H. Liu, X. Chu, Y. Leung, X. Jia, and P. Wan, "General maximal lifetime Sensor-Target surveillance problem and its solution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1757–1765, 2011.

[4] S. S. Kulkarni and L. Wang, "MNP: multihop network reprogramming service for sensor networks," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, pp. 7–16, usa, June 2005.

[5] X. Wang, J. Wang, and Y. Xu, "Data dissemination in wireless sensor networks with network coding," *Eurasip Journal on Wireless Communications and Networking*, vol. 2010, Article ID 465915, 2010.

[6] Y. F. Wen and W. Liao, "Minimum power multicast algorithms for wireless networks with a Lagrangian relaxation approach," *Wireless Networks*, vol. 17, no. 6, pp. 1401–1421, 2011.

[7] S. El Rouayheb, A. Sprintson, and P. Sadeghi, "On coding for cooperative data exchange," in *Proceedings of the IEEE Information Theory Workshop 2010 (ITW '10)*, pp. 1–5, January 2010.

[8] T. A. Courtade, B. Xie, and R. D. Wesel, "Optimal exchange of packets for universal recovery in broadcast networks," in *Proceedings of the IEEE Military Communications Conference (MILCOM '10)*, pp. 2250–2255, November 2010.

[9] N. Milosavljevic, S. Pawar, S. El-Rouayheb, M. Gastpar, and K. Ramchandran, "Deterministic algorithm for the cooperative data exchange problem," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '11)*, pp. 410–414, 2011.

[10] S. Tajbakhsh, P. Sadeghi, and R. Shams, "A generalized model for cost and fairness analysis in coded cooperative data exchange," in *Proceedings of the IEEE International Symposium on Network Coding (NetCod '11)*, pp. 1–6, 2011.

[11] X. Wang, W. Song, C. Yuen, and J. T. Li, "Exchanging third-party information with minimum transmission cost," in *Proceedings of the IEEE Global Communications Conference (Globecom '12)*, 2012.

[12] K. Liu and V. C. S. Lee, "RSU-based real-time data access in dynamic vehicular networks," in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC '10)*, pp. 1051–1056, September 2010.

[13] K. Liu and V. C. S. Lee, "Adaptive data dissemination for timeconstrained messages in dynamic vehicular networks," *Transportation Research C*, vol. 21, pp. 214–229, 2012.

[14] M. Al-Ameen, S. M. R. Islam, and K. Kwak, "Energy saving mechanisms for MAC protocols in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2010, Article ID 163413, 16 pages, 2010.

[15] M. Elkin, Y. Lando, Z. Nutov, M. Segal, and H. Shpungin, "Novel algorithms for the network lifetime problem in wireless settings," *Wireless Networks*, vol. 17, no. 2, pp. 397–410, 2011.

[16] B. Shrader and T. Royster, "Cooperative multicast strategies under heterogeneous link loss rates," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '11)*, pp. 1–5, December 2011.

[17] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[18] J. Wang, J. Wang, K. Lu, B. Xiao, and N. Gu, "Optimal linear network coding design for secure unicast with multiple

streams," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 1–8, March 2010.

[19] M. Ghaderi, D. Towsley, and J. Kurose, "Network coding performance for reliable multicast," in *¿Proceedings of the Military Communications Conference (MILCOM '07)*, October 2007.

[20] X. Wang, K. Wu, J. Wang, and Y. Xu, "CAPF: coded anycast packet forwarding for wireless mesh networks," *Wireless Networks*, vol. 17, no. 5, pp. 1273–1285, 2011.

[21] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 3, pp. 497–510, 2008.

[22] C. Fragouli, J. Y. Le Boudec, and J. Widmer, "Network coding: an instant primer," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, 2006.

[23] M. Ghaderi, D. Towsley, and J. Kurose, "Reliability gain of network coding in lossy wireless networks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 2171–2179, April 2008.

[24] N. Cai and R. W. Yeung, "Secure network coding on a wiretap network," *IEEE Transactions on Information Theory*, vol. 57, no. 1, pp. 424–435, 2011.

[25] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proceedings of the IEEE 19th Annual Joint Conference of Computer and Communications Societies (INFOCOM '00)*, pp. 585–594, March 2000.

[26] C. Chan, *Generating secret in a network [Ph.D. dissertation]*, Massachusetts Institute of Technology, 2010.

[27] Y. Kim and G. De Veciana, "Is rate adaptation beneficial for inter-session network coding?" in *Proceedings of the IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 635–646, 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 950296, 10 pages
doi:10.1155/2012/950296

Research Article

Low-Complexity Decoding Algorithms for Distributed Space-Time Coded Regenerative Relay Systems

Chao Zhang[1] and Huarui Yin[2]

[1] *School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*
[2] *Department of Electronics Engineering and Information Science, University of Science and Technology of China, China*

Correspondence should be addressed to Chao Zhang, chaozhang@mail.xjtu.edu.cn

Received 5 June 2012; Revised 22 July 2012; Accepted 9 August 2012

Academic Editor: Hongli Xu

We examine decoding structure for distributed space-time coded regenerative relay networks. Given the possible demodulation error at the regenerative relays, we provide a general framework of error aware decoder, where the receiver exploits the demodulation error probability of relays to improve the system performance. Considering the high computational complexity of optimal Maximum Likelihood (ML) decoder, we also propose two low-complexity decoders, Max-Log decoder and Max-Log-Sphere decoder. Computational complexities of these three decoders are also analyzed. Simulation results show that error aware decoders can improve system performance greatly without high system overload and Max-Log decoder and Max-Log-Sphere decoder can drastically reduce the decoding complexity with negligible performance degradation.

1. Introduction

Relay-assisted communication is a promising strategy that exploits spatial diversity available among a collection of distributed single antenna terminals for both centralized and decentralized wireless networks. In most relay networks, a two-stage relaying strategy is used. In the first stage, a source transmits and all relays listen; in the second stage, the relays cooperate to forward the source symbols to the destination. Generally speaking, the relay functions can be separated into two types, regenerative and nonregenerative. If the relay processes the received signal, we call it regenerative relay, such as Decode-and-Forward (DCF) [1] and Demodulation-and-Forward (DMF) [2]. Otherwise, we call nonregenerative relay, such as Amplify-and-Forward (AF) [1].

It is well known that the channel between source and relay is unreliable because of fading and noise. The relay receives an attenuated version of the source signal. AF relaying scheme amplifies noise. DCF scheme always using cyclic redundancy check (CRC) will cause interruptions when the relay detects errors from the received message. DMF scheme is a tradeoff between AF and DCF in relay processing. Relay can always keep a transmit link from the source and detects and possibly decodes the source signal [3]. Moreover, the DCF scheme can also be considered as a special case of DMF if we consider the null signal as one choice of the modulation constellation. Therefore, in this paper, we treat DMF as the object to be studied for regenerative relay networks. However, DMF relay has an important disadvantage, which is the error produced in relay's Maximum Likelihood demodulation degrades the effective SNR at the destination significantly, which is called error propagation [4]. For distributed space-time coding system in regenerative relay networks, the degradation is more drastic [5, 6]. In [3], we proposed a threshold-based scheme to minimize the error propagation, which is an active mechanism equipped in relays but subject to the large computation complexity.

In this paper, we intend to investigate the ML decoding structure where the destination is able to be aware of the error probability at the relays. Since the error probability at relay is a monotonic decreasing function of received SNR at relay, the destination can estimate the error probability through training sequences which is transmitted by source and amplified by relay. Meanwhile, each relay also transmits its training sequence to estimate the relay-destination

channel [5, 7]. Therefore, error aware distributed space-time decoding is reasonable. After analyzing the conditional likelihood function, we give a general framework of error aware decoder for regenerative relay networks. Because the proposed ML decoder is composed of multiple likelihood function generators, the computational complexity is too large to be affordable in some cases. Due to max-log approximation, we provide a Max-Log decoder based on Csiszár-Tusnady algorithm [8]. Moreover, to reduce the complexity further, we also propose a Max-Log-Sphere decoder which combines max-log approximation and sphere decoding. In addition, we analyze complexities of these decoders in terms of elementary operation number. Finally, simulations verify the low complexity and improved performance of our proposed decoders.

2. System Model

We consider a wireless network with N randomly placed relay nodes, relay $i = 1,\ldots,N$, one source node S, and a destination node D. Each node is equipped with only a single antenna and uses the Half-duplex mode. Denote the channel from the source to the ith relay as f_i and the channel from the ith relay to the destination as h_i. Assume that $\{f_i\}$ and $\{h_i\}$ are independent complex Gaussian random variables with zero-mean and variance δ_{si}^2 and δ_{id}^2, respectively. Receiver noise is assumed as complex Gaussian random variable with zero-mean and unit-variance. We assume a block fading channel model, where channel gain stays constant during a time block and changes from block to block [1]. We also assume that the instantaneous channel is unknown to the transmitting node but perfectly known at receiving node. Assume that the source wishes to send the signal $\mathbf{s} = [s_1, s_2, \ldots, s_T]^T$ to the destination, where $s_i \in \mathcal{A}$ and $\mathcal{A}$ is a finite constellation with average power $1/T$. Here T is the signal block length. Hence, $E\{\mathbf{s}^H\mathbf{s}\} = 1$. Assume $\mathbf{s}$ is in the codebook $\mathcal{S} = \{\mathbf{s}_1,\ldots,\mathbf{s}_L\}$, where $L \geq 2$ is the cardinality of the codebook. For the convenience of expression, we consider all transmit power is unit.

During the first stage, the source node transmits $\mathbf{s}_l, l \in [0, L-1]$ to all relays, then each relay tries to demodulate the received signal. Denote the demodulated symbol vector at the ith relay is $\hat{\mathbf{s}}_i$, and $\hat{\mathbf{s}}_i \in \mathcal{S}$. By (6.5) of [9], the demodulation probability can be written as

$$P(\hat{\mathbf{s}}_i \mid f_i, \mathbf{s}_l) = \begin{cases} \mathcal{Q}\left(\sqrt{\frac{1}{2}|f_i|^2|\mathbf{s}_l - \hat{\mathbf{s}}_i|^2}\right), & \text{if } \hat{\mathbf{s}}_i \neq \mathbf{s}_l \\ 1 - \sum\limits_{1\leq j\leq L-1, j\neq l} \mathcal{Q}\left(\sqrt{\frac{1}{2}|f_i|^2|\mathbf{s}_l - \mathbf{s}_i|^2}\right), & \text{if } \hat{\mathbf{s}}_i = \mathbf{s}_l, \end{cases} \tag{1}$$

where $\mathcal{Q}(x) = (1/\sqrt{2\pi})\int_x^\infty e^{-t^2/2}dt$. If the source node transmits $\mathbf{s}_l$, the ith relay decode its received signal as $\mathbf{s}_i \neq \mathbf{s}_l$ with probability $\mathcal{Q}(\sqrt{(1/2)|f_i|^2|\mathbf{s}_l - \hat{\mathbf{s}}_i|^2})$. Obviously, the probability of decoding successfully at the ith relay is $1 - \sum_{1\leq j\leq L-1, j\neq l}\mathcal{Q}(\sqrt{(1/2)|f_i|^2|\mathbf{s}_l - \mathbf{s}_j|^2})$.

At the ith relay, the received signal $\hat{\mathbf{s}}_i$ is mapped onto a $T \times 1$ vector (not necessary), $\mathcal{F}_i(\hat{\mathbf{s}}_i)$, as processed at one antenna of colocated space-time coding transmitter. We assume the map function $\mathcal{F}_i$ is invertible. Therefore, there are L possible transmitted vectors to the destination for the ith relay, because $\hat{\mathbf{s}}_i$ could be any vector in $\mathcal{S}$. Herein, we assume all mapping functions $\mathcal{F}_i, i = 1,\ldots,N$, are different with each other. Then, all relays transmit the mapped vectors to the destination. At the destination, the received signal is

$$\mathbf{Y} = [\mathcal{F}_1(\hat{\mathbf{s}}_1),\ldots,\mathcal{F}_N(\hat{\mathbf{s}}_N)]\mathbf{H} + \mathbf{N}, \tag{2}$$

where $\mathbf{Y} = [y_1,\ldots,y_T]^T$ is the received signal, $\mathbf{H} = [h_1,\ldots,h_N]^T$ is the relay to destination channel vector, and $\mathbf{N}$ is Gaussian white noise. Define a codebook $\mathcal{C} = \{[\mathcal{F}_1(\hat{\mathbf{s}}_1),\ldots,\mathcal{F}_N(\hat{\mathbf{s}}_N)]\}$. Clearly, $\mathcal{C}$ includes L^N elements. We define the kth element of $\mathcal{C}$ is $\mathbf{C}_k = [\mathbf{c}_{1k},\ldots,\mathbf{c}_{Nk}]$, where $\mathbf{c}_{ik} \in \{\mathcal{F}_i(\hat{\mathbf{s}}_i)\}$. Thus, if $\mathbf{C}_k$ is transmitted, we can express (2) as

$$\mathbf{Y} = \mathbf{C}_k\mathbf{H} + \mathbf{N}. \tag{3}$$

Denote the inverse function of $\mathcal{F}$ as $\mathcal{F}^{-1}$. Then, we have

$$P(\mathbf{C}_k \mid \mathbf{F}, \mathbf{s}_l) = \prod_{i=1}^{N} P(\mathcal{F}_i^{-1}(\mathbf{c}_{ik}) \mid f_i, \mathbf{s}_l), \tag{4}$$

where $\mathbf{F} = [f_1,\ldots,f_N]^T$. So, by (1), we can derive the exact value of (4). Given a $\mathbf{C}_k$, the conditional probability density function of $\mathbf{Y}$ is

$$P(\mathbf{Y} \mid \mathbf{H}, \mathbf{C}_k) = \frac{1}{\pi^N}\exp\left(-(\mathbf{Y} - \mathbf{C}_k\mathbf{H})^{\mathcal{H}}(\mathbf{Y} - \mathbf{C}_k\mathbf{H})\right). \tag{5}$$

3. Error Aware Maximum Likelihood Decoder

In this section, we provide a general Maximum Likelihood decoder for distributed space-time coded regenerative relay networks. First of all, destination should know the channel information in this relay networks. The channels from relays to the destination $h_i, i = 1,\ldots,N$, can be estimated through pilot symbols which are transmitted by each relay before data transmission [10]. Herein, we assume all the estimators are ideally accurate without error. The effect of estimation error will be checked in simulations. To let the destination know the demodulation error probability of each relay, we propose the following extra channel estimation scheme.

Step 1. The source transmits its pilot symbol to all relays through channels $f_1,\ldots,f_N$. Without demodulating, each relay maps the noise version signal to a vector like the scheme proposed in [11].

Step 2. Each relay transmits the vector to the destination like the Amplify-and-Forward based distributed space-time coding [12].

Step 3. The cascaded channel between source and destination carried by amplified pilots, that is, $f_ih_i, i = 1,\ldots,N$, can be estimated at the destination like [11].

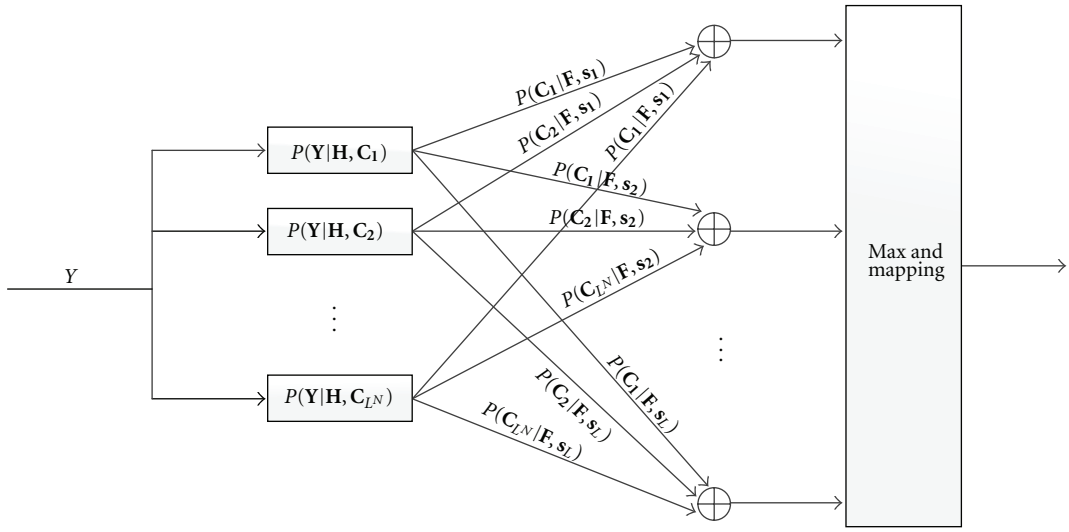


Figure 1: Error aware ML decoder.

Therefore, f_i also can be estimated by the above channel estimation scheme. It is difficult to analyze the effect of channel estimation error in error aware decoders, but we simulate that in Section 6. In following context, we just assume there is no channel estimation error to allow us to focus only on structures of error aware decoders. Because the signal vector set $\mathcal{S}$ and all mapping functions $\{\mathcal{F}_i\}$ are known at the destination as a prior knowledge, (4) can be derived at the destination.

If the transmitted signal is $\mathbf{s}_l$, the likelihood function is

$$P(\mathbf{Y} \mid \mathbf{F}, \mathbf{H}, \mathbf{s}_l) = \sum_{k=1}^{L^N} P(\mathbf{Y} \mid \mathbf{H}, \mathbf{C}_k) P(\mathbf{C}_k \mid \mathbf{F}, \mathbf{s}_l). \quad (6)$$

Therefore, the error aware ML decoder is

$$\arg \max_{1 \le l \le L} \{P(\mathbf{Y} \mid \mathbf{F}, \mathbf{H}, \mathbf{s}_l)\}. \quad (7)$$

Utilize (4)–(6), then (7) is derived. By (5), $P(\mathbf{Y} \mid \mathbf{H}, \mathbf{C}_k)$ is independent of $\mathbf{s}_l$, so that according to the estimated channel $\mathbf{H}$, $P(\mathbf{Y} \mid \mathbf{H}, \mathbf{C}_k)$ can be calculated first. Then, using the amplified pilot, $P(\mathbf{C}_k \mid \mathbf{F}, \mathbf{s}_l)$ is also derived. Therefore, the ML decoder can be built in Figure 1, where we show the structure of error aware ML decoder for regenerative distributed space-time coding. Note that there are L adders and L^N likelihood function generators.

The complexity of the error aware ML decoder equals to that of L^N colocated space-time decoders it is too large to be affordable if the signal block length, modulation order, and the number of relay are considerablely large. Reference [2] considered a piecewise-linear approximation to solve a similar problem, but in this case it is also too complicated to design an approximation function.

3.1. Optimality of Error Aware Decoder. To prove the optimality of our proposed error aware receiver, we need to analyze the error performance difference between the error aware decoder and nonerror aware decoder (traditional receiver). Unfortunately, it is difficult to derive the exact error performances of error aware decoder and nonerror aware decoder. To illustrate the optimality, we try to give following two points (in Figure 2).

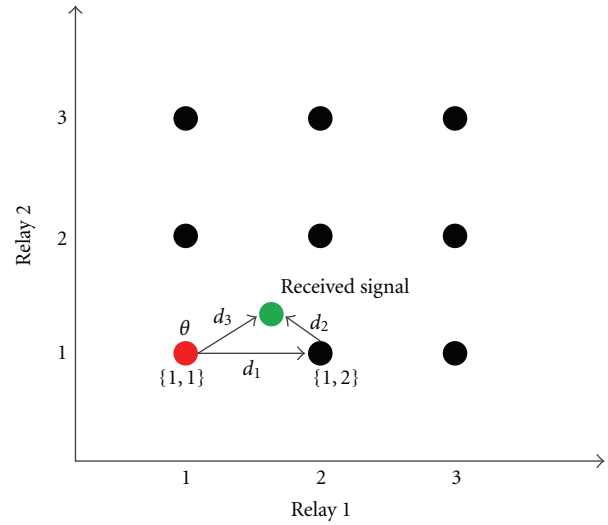


Figure 2: Illustration of the signal space.

(1) *Receiver Rule.* Since the error aware decoder is based on ML rule, it should be the optimal receiver [10].

(2) *Signal Space Description.* To express clearly, we set the source transmit a symbol $s \in \mathcal{A}$ and $\mathcal{A} = \{1, 2, 3\}$. We assume 1 is transmitted by the source. Consider there are 2 relays in the relay networks. Then, it is obvious that there could be 3^2 possible decoding combinations at the relay network, that is, $\{1,1\}$, $\{1,2\}$, $\{2,1\}$,..., $\{3,2\}$, $\{3,3\}$. In the nonerror aware decoder, combinations $\{1,1\}$, $\{2,2\}$, and $\{3,3\}$ are considered. In following, we consider two cases.

Case 1 (No error happens at relays). In this case, relays decode the received message as 1, which means decode set is $\{1,1\}$ (red ball). According to (6), error aware decoder is equivalent to the nonerror aware decoder. Therefore, both have the same error performances.

Case 2 (Error happens at relays). In our interested situation, the impact of noise is very slight therefore, the system SNR is very high and we could only consider the closest symbols as errors. As a result, our candidates are also limited between $\{1,2\}$ and $\{2,1\}$ in error aware decoder. As we know, the error performance is a Q-function of the distance between the transmitted symbol and the received symbols on the signal space [10]. Therefore, the error performance of nonerror aware receiver could be expressed as $Q(\sqrt{d_3^2\text{SNR}})$. By (6), the error performance of error aware decoder is $Q(\sqrt{d_1^2\text{SNR}})Q(\sqrt{d_2^2\text{SNR}})$. As we consider high SNR regime, then there is $Q(x) \approx \exp(-x^2/2)$. Then the error performance of error aware decoder can be approximated as $\exp(-(d_1^2+d_2^2)\text{SNR}/2)$. Denote the angle between d_1 and d_3 as θ. Because the received signal is so close to $\{1,2\}$, then $\cos\theta > 0$. By the law of cosines, we have $\exp(-(d_1^2 + d_2^2)\text{SNR}/2) < \exp(-d_3^2\text{SNR}/2)$. For $\{2,1\}$, we can obtain the same result. So we can prove that error aware decoder outperforms nonerror aware decoder.

4. Low-Complexity Error Aware Decoders

In this section, we will introduce two low-complexity error aware decoders through analyzing and simplifying the structure of ML decoder. The simplifying process we used herein can be extended for more general cases to obtain low-complexity decoders. First, we use Max-Log approximation to derive a Max-Log error aware decoder which can work with Csiszár-Tusnady algorithm. Second, to reduce the complexity further, sphere decoding also is combined into the Max-Log decoder, which is called Max-Log-Sphere decoder.

4.1. Error Aware Max-Log Decoder. Substitute (6) into (7), there is

$$\arg\max_{1\le l\le L}\left\{\sum_{k=1}^{L^N} P(\mathbf{Y} \mid \mathbf{H}, \mathbf{C}_k)P(\mathbf{C}_k \mid \mathbf{F}, \mathbf{s}_l)\right\}. \tag{8}$$

Because $\log(x)$ is an increasing monotonic function, the ML decoder can be rewritten as

$$\arg\max_{1\le l\le L}\left\{\log\left(\sum_{k=1}^{L^N}\exp\left\{-\frac{1}{2}\|\mathbf{Y}-\mathbf{C}_k\mathbf{H}\|^2+\lambda_{k,l}\right\}\right)\right\}. \tag{9}$$

According to (5) and max-log approximation in [13], we derive

$$\arg\max_{1\le l\le L}\left\{\arg\max_{1\le l\le L^N}\left\{-\|\mathbf{Y}-\mathbf{C}_k\mathbf{H}\|^2+\lambda_{k,l}\right\}\right\}, \tag{10}$$

where $\lambda_{k,l} = \log(P(\mathbf{C}_k|\mathbf{F}, \mathbf{s}_l))$.

We can see that decoding distributed space-time code becomes searching a two-dimension array, which is indexed by (k,l). Intuitively, this decoder also needs L^N ML detector like ML decoder. The only difference is that calculating the likelihood function of each symbol vector does not need cross-computation. However, double maximization problem can take advantage of Csiszár-Tusnady algorithm to reduce computing [8]. Because the set $\{\|\mathbf{Y}-\mathbf{C}_k\mathbf{H}\|^2\}$ is a set of distance measure which is one-one mapped to a probability distribution set and $\{\lambda_{k,l}\}$ is the set of probability distribution, moreover, $\{\lambda_{k,l}\} \le 0$, then (10) can be seemed to seek the vector which has the minimum sum distance which equals to the distance from $\mathbf{s}_l$ to $\mathbf{C}_k$ plus the distance from $\mathbf{C}_k$ to $\mathbf{Y}$. Thus, Csiszár-Tusnady algorithm does converge to the maximum element [8]. We summarize the iterative Max-Log decoder as follows (Figure 3).

4.2. Error Aware Max-Log-Sphere Decoder. If the length of vector $\mathbf{s}$ and the constellation size are sufficiently large, Max-Log decoder is also subject to the implementation. The largest computation is required for searching code set $\mathcal{C}$ with cardinality L^N. Reducing the decoder complexity depends on searching $\mathcal{C}$.

To state the Max-Log-Sphere decoder, we first find the real-valued equivalent of (3), Define

$$\begin{aligned}
\widetilde{\mathbf{Y}} &= \left[\mathcal{R}\{\mathbf{Y}\}^T, \mathcal{I}\{\mathbf{Y}\}^T\right]^T_{2T\times 1},\\
\widetilde{\mathbf{H}} &= \left[\mathcal{R}\{\mathbf{H}\}^T, \mathcal{I}\{\mathbf{H}\}^T\right]^T_{2N\times 1},\\
\widetilde{\mathbf{N}} &= \left[\mathcal{R}\{\mathbf{N}\}^T, \mathcal{I}\{\mathbf{N}\}^T\right]^T_{2T\times 1},\\
\widetilde{\mathbf{C}}_k &= \begin{bmatrix}\mathcal{R}\{\mathbf{C}_k\} & \mathcal{I}\{\mathbf{C}_k\}\\ -\mathcal{I}\{\mathbf{C}_k\} & \mathcal{R}\{\mathbf{C}_k\}\end{bmatrix}_{2T\times 2N},
\end{aligned} \tag{11}$$

where $\mathcal{R}\{\cdot\}$ and $\mathcal{I}\{\cdot\}$ denote real part and imaginary part. By (10), we yield

$$\arg\min_{1\le l\le L}\left\{\min_{1\le k\le L^N}\left\{\left\|\widetilde{\mathbf{Y}}-\widetilde{\mathbf{C}}_k\widetilde{\mathbf{H}}\right\|^2-\lambda_{k,l}\right\}\right\}. \tag{12}$$

For a specific $\mathbf{s}_l$, the decoding object is

$$\begin{aligned}
&\arg\min_{1\le l\le L^N}\left\{\left\|\widetilde{\mathbf{Y}}-\widetilde{\mathbf{C}}_k\widetilde{\mathbf{H}}\right\|^2-\lambda_{k,l}\right\}\\
&\quad=\arg\min_{1\le l\le L^N}\left\{\left\|\widetilde{\mathbf{Y}}-\left(\widetilde{\mathbf{H}}^T\otimes\mathbf{I}\right)\ \text{Vec}\left\{\widetilde{\mathbf{C}}_k\right\}\right\|^2-\lambda_{k,l}\right\},
\end{aligned} \tag{13}$$

where $\otimes$ is the Kronecker product operation. Obviously, we can use sphere decoding method [14, 15] to searching $\mathbf{C}_k$, which minimizes (13). Note that $\|\widetilde{\mathbf{N}}\|^2 = \|\widetilde{\mathbf{Y}}-\widetilde{\mathbf{C}}\widetilde{\mathbf{H}}\|^2$ is an χ^2 random variable with $2N$ degrees of freedom. We choose the radius r to be a linear function of the variance of $\|\widetilde{\mathbf{N}}\|^2$

$$r^2 = 2\alpha N, \tag{14}$$

where the coefficient α is chosen in such a way that with a high probability P_{fp} we can find a lattice inside a sphere

$$\int_0^{2\alpha N}\frac{x^{N-1}}{\Gamma(N)}e^{-x}dx, \tag{15}$$

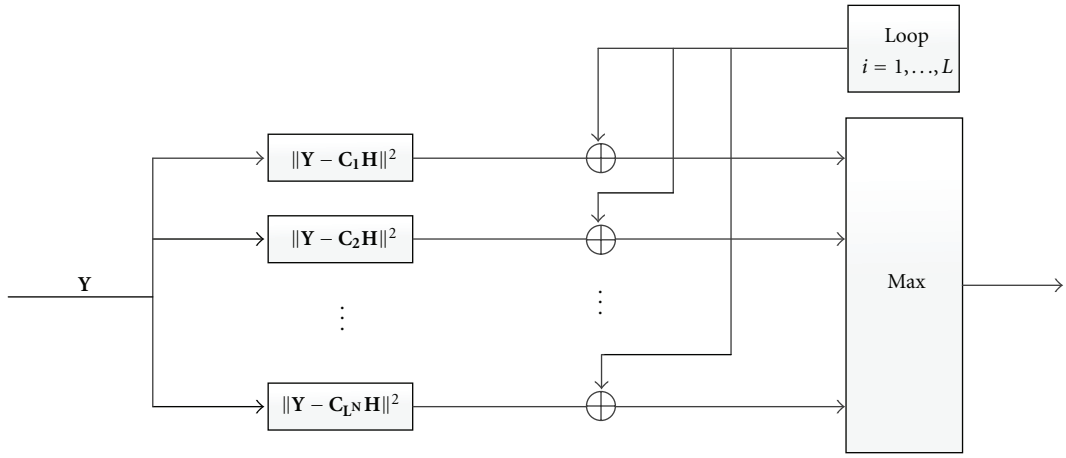


Figure 3: Max-Log decoder.

(1) **Initialization**: Set error probability set $\mathcal{P}$. Take any element of $\mathbf{C}_k \in \mathcal{C}$ and compute $\mathbf{D}_k = -\|\mathbf{Y} - \mathbf{C}_k\mathbf{H}\|^2$.
(2) **Step 1**: Find the l that makes $\mathbf{D}_k + \lambda_{k,l}$ is maximum for the chosen k, where $\lambda_{k,l} \in \mathcal{P}$ and is calculated by (4).
(3) **Step 2**: Fix l and find a $\mathbf{D}_n \in \mathcal{C}$ which makes $\mathbf{D}_n + \lambda_{n,l}$ is maximum.
(4) **Decision**: If $n = k$, goto End, otherwise, goto Step 1.
(5) **End**: $\mathbf{s}_l$ is the decoded vector.

Algorithm 1

where $\Gamma(N) = \int_0^\infty t^N e^{-t} dt$. Note that the radius is chosen based on the noise not on channel efficiency. As stated in [14], this point has a beneficial effect on the computational complexity.

For expression convenience, we define $\widetilde{\mathbf{H}}^T \otimes \mathbf{I} = \mathbf{B}$ and Vec $\{\widetilde{\mathbf{C}}_k\} = \mathbf{X}$ with size $4NT \times 1$. Therefore, searching $\mathbf{X}$ is equal to searching $\mathbf{C}_k$. Applying the idea of the Fincke Pohst algorithm (See Algorithm 1), we search for the point $\mathbf{X}$ that belongs to the geometric body described by

$$r'^2_{k,l} \geq \left(\mathbf{X} - \hat{\mathbf{X}}\right)^H \mathbf{U}^H \mathbf{U}\left(\mathbf{X} - \hat{\mathbf{X}}\right) - \sum_{i=1}^{N} \log(P(\mathbf{c}_{ik} \mid f_i, \mathbf{s}_l)), \tag{16}$$

where $\hat{\mathbf{X}} = \mathbf{B}^\dagger \mathbf{Y} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{A}^T\mathbf{Y}$ and $\mathbf{U}$ is the low triangular matrix obtained from QR factorization of $\mathbf{B}$. The search radius $r'_{k,l}$ is chosen according to the statistical properties of noise and the decoding error at relays. Denote $4NT = M$, then, a necessary condition for x_M, the Mth element of $\mathbf{X}_k$, is

$$b^2_{M,M}(x_M - \hat{x}_M)^2 - \delta_{k,l,M} \leq r'^2_{k,l}, \tag{17}$$

where $\delta_{k,l,M}$ is defined as

$$\delta_{k,l,M} = \frac{1}{2N} \log P\left(\mathbf{C}_{j,k} f_j, \mathbf{s}_l\right), \qquad j = \frac{1}{2} \bmod (M, 2T). \tag{18}$$

Herein we allocate the additional weight $\delta_{k,l,m}$ averagely over N relays. Moreover, we define

$$r'^2_{k,l,M-1} = r'^2_{k,l} - b^2_{M,M}(x_M - \hat{x}_M)^2 + \delta_{k,l,M}, \tag{19}$$

and a new necessary condition can be written as

$$b^2_{M-1,M-1}\left(x_{M-1} - \hat{x}_{M-1} + \frac{b_{M-1,M}}{b_{M-1,M-1}}(x_M - \hat{x}_M)\right)^2 - \delta_{k,l,M-1} \leq r'^2_{k,l,M-1}. \tag{20}$$

In a similar fashion, one proceeds for x_{M-2}, and so on, and until all components of vector $\mathbf{X}$ are found. Note that the dominant difference between Max-Log-Sphere decoder and sphere decoder proposed in [14] is that radius varies according to all possible code words. In this Max-Log-Sphere decoder, for each k, we just check $\mathbf{X}_k$ whether to meet its radius. If there exists more than one $\mathbf{X}_k$ can meet the constraint (17) for x_m, keep these survival code word and go to next code word. If some of these code words cannot meet the new constraint, then drop them. That is to say for each lattice we must try all possible radiuses. For a specific $\mathbf{s}_l$, Max-Log-Sphere decoder can be summarized as Figure 4.

After terminating the decoder algorithm for $\mathbf{s}_l$ (See Algorithm 2), select the $\mathbf{C}_k$ which achieves the minimum distance to $\widetilde{\mathbf{Y}}$. Then through L Max-Log-Sphere decoder with $l = 1, \ldots, L$, choose the $\mathbf{s}_l$ which minimizes the distance to $\mathbf{Y}$.

Note that Max-Log-Sphere decoder needs estimating the noise variance of the receiver. However, Max-Log decoder using Eulerian distance and error probability is more realizable. Hence, there is a tradeoff between computational complexity and implementation to choose which one is suitable.

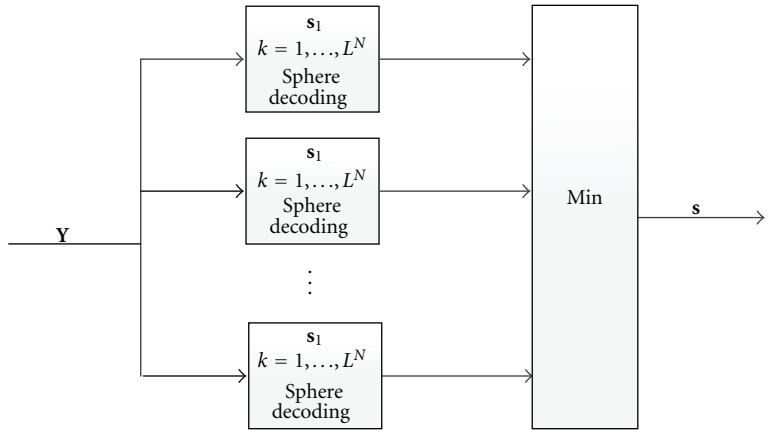


Figure 4: Max-Log-Sphere decoder.

Input $\mathbf{B}, \mathbf{Y}, \hat{\mathbf{X}}, r, \lambda_{k,l}, d$, Set $k = 1$.
(1) Set $m = M$, $r'^2_{k,l,M} = r^2 - \|\tilde{\mathbf{Y}}\|^2 + \|\mathbf{B}\hat{\mathbf{X}}\|$, $\hat{x}_{M|M+1} = \hat{x}_M$.
(2) (Set bounds for x_m) Set $z_k = r'_{k,l,m}/b_{m,m}$,
$UB(x_m) = \lfloor z_k + \hat{x}_{m|m+1} \rfloor$, $x_m = \lceil -z_k + \hat{x}_{m|m+1} \rceil - d$.
(3) (Check x_m) Set $x_m = \mathbf{X}_k(m)$,
if $b^2_{m,m}(x_m - \hat{x}_{m|m+1})^2 > r'^2_{k,l,m} + \delta_{k,l,m}$ and $x_m \leq UB(x_m)$, go to (5), else to (4).
(4) (Increase k) $k = k + 1$, If $k = L^N + 1$, terminate algorithm, else go to (1).
(5) (Decrease m) If $m = 1$ go to (6). Else $m = m - 1$, $\hat{x}_{m,m-1} = \hat{x}_m + \sum_{j=m+1}^{M}(b_{k\cdot j}/b_{m,m})(x_j - \hat{x}_j)$,
$r'^2_{k,l,m} = r'^2_{k,l,m+1} - r^2_{m+1,m+1}(x_{m+1} - \hat{x}_{m+1|m+2})^2 + \delta_{k,l,m+1}$, and go to (2)
(6) Solution found for k. Save k, $\mathbf{X}_k$ and exact distance $d_{k,l}$. and set $k = k + 1$, if $k = L^N + 1$, terminate algorithm, else go to (1).

Algorithm 2

5. Computational Complexity Analysis

In this section, we analyze and compare the computational complexity of above three decoders. We use the average numbers of real elementary operation, C_p (including addition, subtraction, multiplication, and division), as a measure for computational complexity.

5.1. Complexity of ML Decoder. By (5), it easy to know that compute $P(\mathbf{Y} \mid \mathbf{H}, \mathbf{C}_k)$ needs $TN+5T+3N+1$ times additions and $4NT + 4T$ multiplications. Similarly, observe (4), we also can figure out that compute $P(\mathbf{c}_k \mid \mathbf{F}, \mathbf{s})$ needs $N - 1$ multiplications. Therefore, there are $(L^N - 1)+L^N(TN+5T+3N-1)$ additions and $L^N(4NT+4T+N-1)$ multiplications to obtain $P(\mathbf{Y} \mid \mathbf{F}, \mathbf{H}, \mathbf{s}_l)$. As a result, it needs

$$C_p^{\mathrm{ML}} = L^{N+1}(5NT + 9T + 4N - 1) - L. \tag{21}$$

operations (additions and multiplications) to perform ML decoder.

5.2. Complexity of Max-Log Decoder. Recall (10), $\|\mathbf{Y} - \mathbf{C}_k\mathbf{H}\|^2$ needs $TN+5T+3N-1$ additions and $4NT+4T$ multiplications and $\lambda_{k,l}$ needs $N - 1$ multiplications. Therefore, Max-Log decoder needs

$$C_p^{\mathrm{Max-log}} = L^N(5NT + 9T + 3N - 1) + L^{N+1}(N - 1) + L. \tag{22}$$

real operations. To compare the complexities of ML decoder and Max-Log decoder, we have

$$C_p^{\mathrm{ML}} - C_p^{\mathrm{Max-log}} = L^{N+1}\left[(5N + 9T + 3N - 1)\left(1 - \frac{1}{L}\right) + 1 - \frac{2}{L^N}\right]. \tag{23}$$

As stated in system model, $L \geq 2$ and $N \geq 1$, so there is $C_p^{\mathrm{ML}} > C_p^{\mathrm{Max-log}}$. That is to say ML decoder has higher complexity than Max-Log decoder.

5.3. Complexity of Max-Log-Sphere Decoder. Max-Log-Sphere decoder for a $\mathbf{s}_l$ has L^N radiuses but each radius is only assigned for searching one possible $\mathbf{C}_k$. According to [14, 15], an arbitrary lattice point $\mathbf{X}_k$ that belongs to an m dimensional sphere of radius $r_{k,l}$ around the transmitted

point $\mathbf{X}_t$ is given by the following incomplete Gamma function:

$$\gamma\left(\frac{2\alpha N+\lambda_{k,l}}{2\left(1+\Pr\left\|\mathbf{X}_t^m-\mathbf{X}_k^m\right\|^2\right)},\frac{2N-2T+m}{2}\right) = \int_0^{2(1+\Pr\|\mathbf{X}_t^m-\mathbf{X}_k^m\|^2)}\frac{x^{((2N-2T+m)/2)-1}}{\Gamma((2N-2T+m)/2)}e^{-x}dx, \tag{24}$$

where $\mathbf{X}^m = [x_{M-m}, x_{M-m+1}, \ldots, x_M]^T$, and Pr is the relay transmit power, which is assumed as unit in above context. The number of elementary operations that the Max-Log-Sphere decoder performs per each visited point in dimension m is

$$C_p(k,l,m) = 2m + 12. \tag{25}$$

Denote (24) as $P_{k,l}$; therefore, C_p of Max-Log-Sphere decoder is yielded as

$$C_p = \sum_{l=1}^{L}\sum_{k=1}^{L^N}\sum_{m=1}^{M} C_p(k,l,m)P_{k,l}. \tag{26}$$

It is difficult to compare Max-Log-Sphere decoder with other decoders, but in next section we will show simulation results to illustrate the differences.

6. Simulation Results

In this section, we provide the simulation results to show the proposed error aware decoders. We denote the total power noise ratio as the system signal-noise ratio (SNR) indicator. And half of total power is assigned for source transmit power, and another half is equally divided by all relays. In this simulation, we adopt distributed linear dispersion code proposed in [12] as the coding scheme for its simplicity, where $\mathcal{F}_i(s) = \mathbf{A}_i\mathbf{s}$ and $\mathbf{A}_i$ is a random unitary matrix. For Max-Log-Sphere decoder, herein we set $P_{fp} = 0.99$. All other parameters are the same with system model. We also should claim the nonerror aware decoder is

$$\arg\min_{\mathbf{s}}\|\mathbf{Y}-\mathbf{C}(\mathbf{s})\mathbf{H}\|^2, \tag{27}$$

where $\mathbf{C}(\mathbf{s}) = [\mathbf{A}_1\mathbf{s}, \mathbf{A}_2\mathbf{s}, \ldots, \mathbf{A}_N\mathbf{s}]$.

6.1. Performance Comparison with Ideal Receivers. Figure 5 demonstrates bit error rate (BER) performances of different decoders where two relays are employed and the signal modulation is BPSK. That is to say $T = N = 2$. We can see that at high SNR regime error aware decoders achieve almost 6 dB gain than nonerror aware decoder and outperform AF scheme-based ML decoder about 3 dB. Thus it is worthy to bring slight system overhead for delivering channel estimation to improve the system performance. Over all SNR range, Max-Log decoder and Max-Log-Sphere decoder have nearly the same performance with ML decoder. Therefore, the degradation of Max-Log approximation is negligible. Carefully observing, we found that the slope of

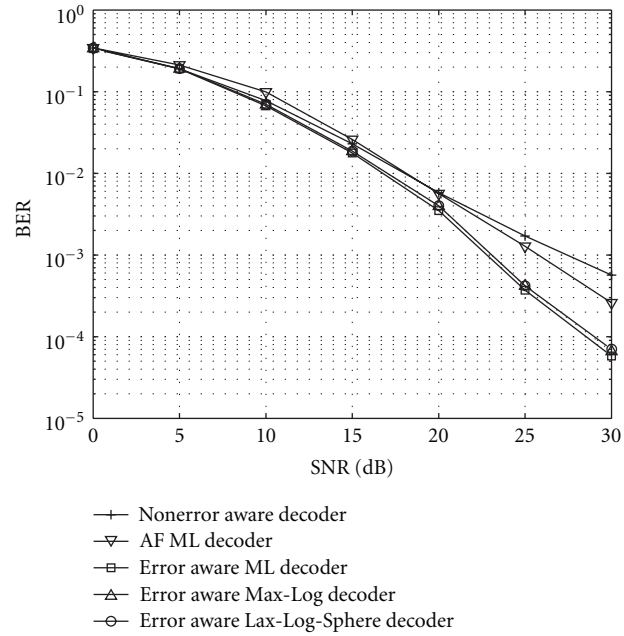


Figure 5: BER performance of error aware decoders (2 relays, ideal receiver).

BER curve decreases. The reason is that the hard-decision error at relay limits the systems performance even though SNR is enough high.

In Figure 6, we also simulate a 4-relay network to show the BER performance of that decoders. Herein, $T = N = 4$ and modulation is QPSK. Similarly, error aware decoders can bring about 7 dB power gain than nonerror aware decoder at 22 dB SNR. We can see that it is different from Figure 5 that error aware decoders only achieve about 1.5 dB gain than AF-based ML decoder. The reason is that high-order modulation incurs more error after decoding at relays and enlarges errorpropagation so that deceases the possible gain of error aware decoder. And in this case, the differences of three error aware decoders are more slight. It is interesting that the slope of BER curve does not decrease here. That is because more relays bring more error conditions and consume more power. Therefore, the slope decreasing threshold is larger than 2 relay with BPSK system. From both two figures, we can assert that error aware decoders can improve the system performance efficiently with little system cost.

For distributed space-time coded (DSTC) relay networks, [12] had proved that the maximum achievable diversity order is $\min\{N, T\}$. Reference [16] addressed that demodulate-and-forward scheme in a relay network where direct link is available can only achieve half of maximum diversity. In our simulations, relay network with nonerror aware decoder has an even less diversity, that is, the diversity of nonerror aware decoder in Figure 5 is 1 and in Figure 6 is only 1.2. That is because there is no direct link in our model and direct link which does not produce demodulation error. Adding a direct link can increase the system diversity by one but adding on one relay could not

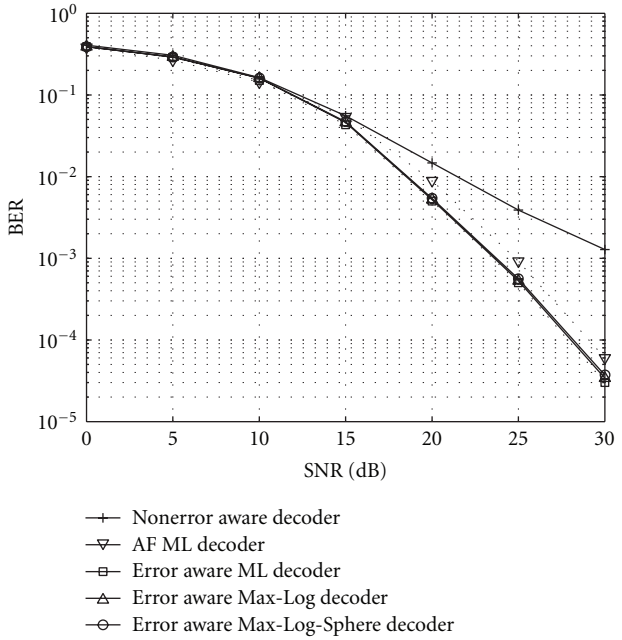


Figure 6: BER performance of error aware decoders (4 relays, ideal receiver).

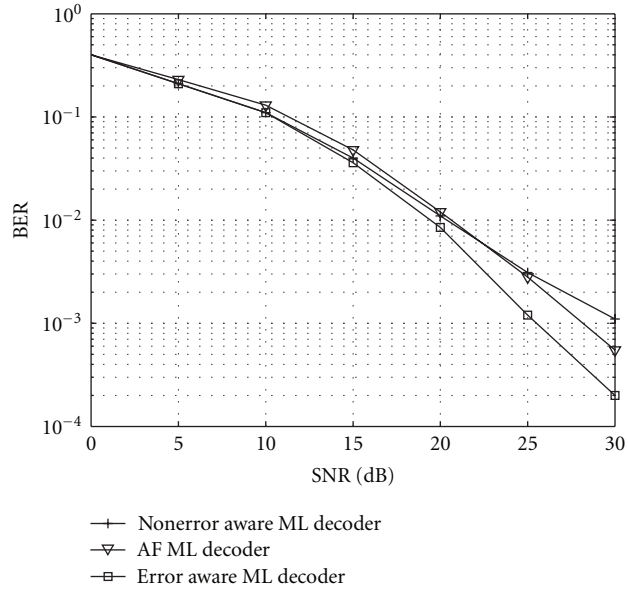


Figure 7: BER performance of error aware decoders (2 relays, practical receiver).

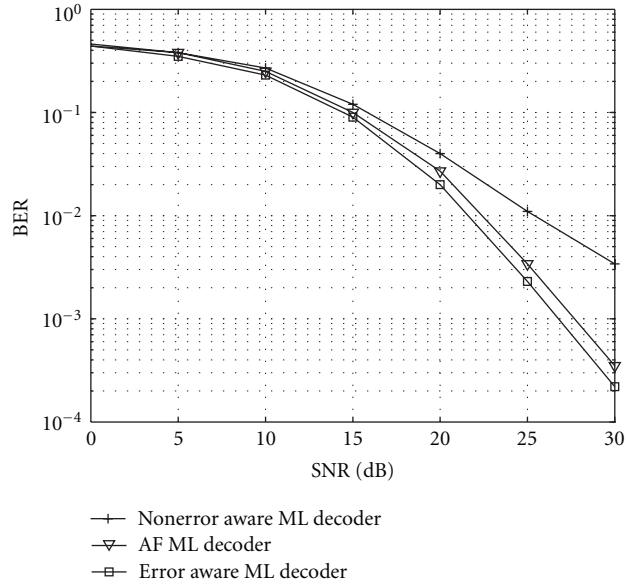


Figure 8: BER performance of error aware decoders (4 relays, ideal receiver).

gain advantages, but even get worse. Essentially speaking, demodulation error limits the diversity growing. On the other hand, error aware decoder has a larger diversity than nonerror aware decoder. That is to say error aware decoder gains advantages from the available error probabilities. Since there are also demodulation errors at relays, error aware decoder cannot achieve the full diversity at finite SNR.

6.2. Performance Comparison with Practical Receiver. In order to validate the practical performance of our proposed error aware decoders, we also consider a practical receiver at the destination, where channel state information is generated by channel estimator. It means that channel state information is not perfect and has estimation error. We set the transmit power of pilot symbols used to estimate channel equal to the transmit power of data symbols. The procedure of channel estimation follows that 3-step scheme described in Section 3. Channel estimators are built on minimum mean square error (MMSE) rule [11]. In addition, the performance degradation of low-complexity decoders is incurred by less searching in codebook. Moreover, both Figures 5 and 6 prove that low-complexity decoders achieve similarly performance compared with the error aware ML decoder. Therefore, in following simulation, we do not draw the performance of all three error aware decoders but error aware ML to compare with other schemes.

Figures 7 and 8 give the BER performances of different decoders with practical receivers where channel state information is not perfect. Clearly, the channel estimation error does not change performance relationship among nonerror aware decoder, AF-based ML decoder, and error aware ML decoder. Comparing Figures 5 and 7, we can see that the performance gain obtained by error aware decoder as compared to AF-based ML decoder decreases from 3 dB to 2.5 dB. We also can find that the gain of error aware decoder over nonerror aware decoder decreases from 6 dB to 5 dB through comparing Figures 6 and 8. That is to say that the uncertainty of channel state information does degrade the performance of our proposed error aware decoder but the degradation is limited. Error aware decoder still outperforms nonerror aware decoder. In summary, our proposed error aware decoder works well in practical receivers.

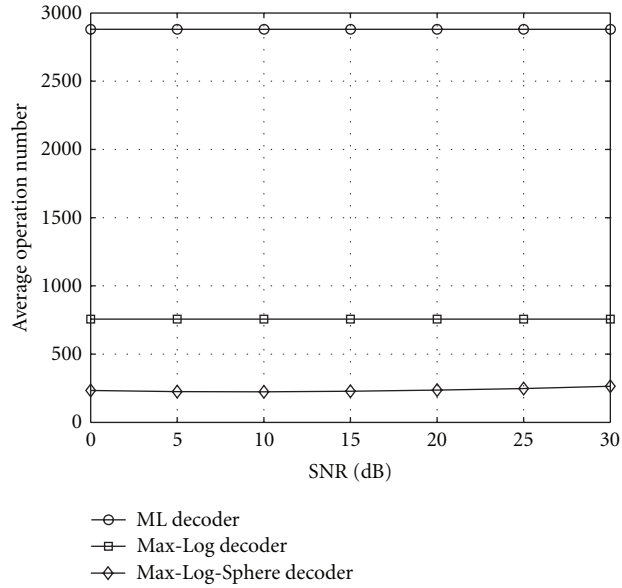


Figure 9: Average operation number of three decoders with 2 relay, BPSK.

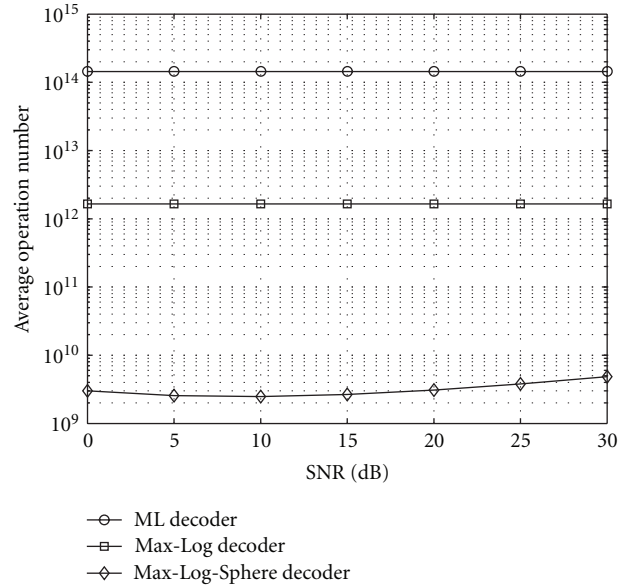


Figure 10: Average operation number of three decoders with 4 relay, QPSK.

6.3. Complexity Comparison. We will show the computational complexity of three error aware decoders by elementary operation number. Note that the operation number of Max-Log-Sphere decoder varies with unitary matrices $\mathbf{A}_i$ and the channel realization because of (13). We average the elementary operation number over 1000 channel realizations.

In Figure 9, we show the average operation number of these three decoders when 2 relays are employed. Obviously, C_ps of ML decoder and Max-Log decoder are independent of SNR. Max-Log decoder has a lower complexity than ML decoder. Max-Log-Sphere decoder needs far smaller operation number than that of ML decoder and Max-Log decoder. Of course, for 2-relay network, the operation number of ML decoder is trivial compared with current hardware computing rate. However, for 4 relays with QPSK modulation scheme, it is too large to be affordable. Figure 10 gives the elementary operation number in this case. We can see that ML decoder has $1.4404e + 014$ operations! The operation number of Max-Log decoder is nearly 1% of that of ML decoder. It is notable that Max-Log-Sphere decoder needs only 0.1% operation number of Max-Log decoder. Therefore, Max-Log-Sphere decoder achieves the same BER performance with optimal ML decoder but costs drastically low computation. Although Max-Log-Sphere has an attractive performance, the noise variance should be estimated first to calculate searching radius [14]. Max-Log decoder just utilizes Eulerian distance and error probabilities; therefore, it is a good tradeoff for decoding structure between implementation and computational complexity. We can choose one of them due to different receivers.

7. Conclusion

In this paper, we provide a general framework of error aware distributed space-time decoder for regenerative relay networks. Through two-stage pilot symbols, the destination can estimate not only the relay-destination channel but also the error probability happening at relays. Using these estimated error, Maximum Likelihood decoder is provided. To reduce computational complexity, Max-Log decoder and Max-Log-Sphere decoder are also proposed by max-log approximation. Simulations show that error aware decoders can improve the performance drastically. Max-Log-Sphere decoder can achieve the same performance with ML decoder and needs far lower computational complexity. Without noise estimating, Max-Log decoder can make a good tradeoff between implementation and computational complexity.

Acknowledgments

This work is supported by National Nature and Science Funding of China (no. 61102082), National High-tech R&D Program (863 Program, no. 2011AA01A105), National Science and Technology Major Project of China (no. 2012ZX03001032-002), the Fundamental Research Funds for the Central Universities, the Specialized Research Fund for the Doctoral Program of Higher Education (no. 20110201120011), and the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University (no. 2011D14).

References

[1] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[2] H. Li and Q. Zhao, "Distributed modulation for cooperative wireless communications," *IEEE Signal Processing Magazine*, vol. 23, no. 5, pp. 30–36, 2006.

[3] C. Zhang, J. Zhang, H. Yin, and G. Wei, "Selective relaying schemes for distributed space-time coded regenerative relay networks," *IET Communications*, vol. 4, no. 6, pp. 967–979, 2010.

[4] F. A. Onat, A. Adinoyi, Y. Fan, H. Yanikomeroglu, and J. S. Thompson, "Optimum threshold for SNR-based selective digital relaying schemes in cooperative wireless networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 970–975, March 2007.

[5] G. Scutari and S. Barbarossa, "Distributed space-time coding for regenerative relay networks," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2387–2399, 2005.

[6] R. Hoshyar and R. Tafazolli, "A pre-BSC model for distributed turbo codes," in *Proceedings of the 69th Vehicular Technology Conference (VTC '09)*, April 2009.

[7] M. N. Khormuji and E. G. Larsson, "Receiver design for wireless relay channels with regenerative relays," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 4034–4039, June 2007.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.

[9] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press, 2003.

[10] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.

[11] F. Gao, T. Cui, and A. Nallanathan, "On channel estimation and optimal training design for amplify and forward relay networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 5, pp. 1907–1916, 2008.

[12] Y. Jing and B. Hassibi, "Distributed space-time coding in wireless relay networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 12, pp. 3524–3536, 2006.

[13] S. Lin D and J. Costello Jr., *Error Control Coding*, Pearson Education, 2nd edition, 2004.

[14] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Transactions on Sginal Processing*, vol. 53, no. 8, pp. 2806–2818, 2005.

[15] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 2299–2311, 2004.

[16] D. Chen and J. N. Laneman, "Modulation and demodulation for cooperative diversity in wireless systems," *IEEE Transactions on Wireless Communications*, vol. 5, no. 7, pp. 1785–1794, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 504205, 11 pages
doi:10.1155/2012/504205

Research Article

A Diagnosis-Based Clustering and Multipath Routing Protocol for Wireless Sensor Networks

Wenjun Liu,[1] Shukui Zhang,[1,2] and Jianxi Fan[1]

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

Correspondence should be addressed to Jianxi Fan, jxfan@suda.edu.cn

Received 29 March 2012; Accepted 29 May 2012

Academic Editor: Yong Sun

In wireless sensor networks, it is of great importance for fault diagnosis to ensure the gathering information accuracy and reduce energy additionally consumed by faulty nodes, for the deployment of a large number of sensor nodes in hostile environment. In this paper, we propose an energy-efficient data collection protocol which consists of clustering and multipath routing. Clustering based on fault diagnosis eliminates the possibility of cluster heads (CHs) acting by faulty nodes which reduce energy consumption and fault information transmission. Multipath routing provided by directed acyclic graph (DAG) increases system fault tolerance. Furthermore, clustering and multihop routing consider residual energy and routing cost, respectively; thus balanced energy consumption is achieved. Performance analysis shows that the message complexity disseminated in clustering and fault diagnosis is acceptable. Simulations demonstrate that the protocol has better energy efficiency compared with other related protocols.

1. Introduction

In recent years, Wireless Sensor Networks (WSNs) have become an attractive technology for a large number of applications, ranging from monitoring to event detection and target tracking [1]. To design and deploy successful WSNs, many issues need to be resolved such as deployment strategies, energy conservation, fault-tolerant routing in dynamic environments, localization, and fault diagnosis. To extend the network lifetime as long as possible, energy efficiency becomes one of the basic tenets in the WSNs protocol design. There are several possible solutions to balance energy consumption, such as deployment optimization [2], topology control [3], and data aggregation [4].

Among these schemes, clustering provides an effective way for promoting energy efficiency [5–9]. In clustering schemes, sensor nodes are organized into clusters and a main node is selected as the cluster head (CH) of a cluster, and the other nodes are called cluster members (CMs). Each CM collects local data from the environment periodically and then sends it to the CH. When the data from all the CMs arrives, the CHs aggregate the data and send it to the BS via single-hop or multihop. When the network is partitioned into clusters, data transmission can be classified into two stages, that is, intra- and intercluster communication. Mhatre and Rosenberg have shown that multihop intercluster communication mode is usually more energy efficient because of the characteristics of wireless channel [10]. Thus it is better to let CHs cooperate with each other to forward their data.

Due to the low cost and the deployment of a large number of sensor nodes in uncontrolled or even harsh environments, it is common for nodes to become faulty. The existence of these faulty nodes in WSNs brings the data collection protocol many adverse effects such as nonuniform distribution of the clustering effect and inaccuracy of the information collected. In addition, too many faulty nodes directly affect the connectivity of the network, resulting in premature network partition, which is an important factor affecting network lifetime. How to identify faulty nodes and eliminate the impact of these nodes gradually attracts more and more attentions.

Multipath routing between a source and a destination is a promising routing scheme to achieve robustness, load balancing, bandwidth aggregation, congestion reduction, and security compared to the single shortest-path routing that is usually used in most networks [11]. Techniques developed for multipath routing are often based on employing multiple

spanning trees or directed acyclic graphs (DAGs) [12, 13]. Both of them offer resiliency to single-link failure. Due to the relative instability of communication links in WSNs, it is necessary to explore the feasibility of the multipath routing.

In this paper, we analyzed the opportunities and challenges of fault diagnosis based on comparison model in WSNs, and so far there is little work done for this owing to the inherent characteristic of WSNs. Furthermore, we design and implement a fault diagnosis-based clustering and multipath routing protocol (FDCM) for wireless sensor networks. FDCM mainly includes two phases: fault diagnosis-based clustering and multipath routing selection. The features making FDCM distinct are as follows. (i) Clustering based on fault diagnosis eliminates the possibility of CH acting by faulty nodes which reduce energy consumption and fault information transmission. (ii) The constructed DAG provides multipath routing approach, which increases system fault tolerance. (iii) Clustering and multihop routing consider residual energy and transmission cost, respectively, which balance energy consumption and promote network efficiency. Consequently, the communication overhead and network lifetime of FDCM are desirable.

The rest of the paper is organized as follows. The next section describes the related work. Section 3 introduces the network model and related terminologies at first, after which is the fault diagnosis model based on comparison model. Subsequently, it analyzes the fault diagnosis addressing WSNs and provides several requirements. The system design including clustering construction and multihop routing is detailed in Section 4. The simulation results are given in Section 5. Finally, the conclusion and the future work are drawn in Section 6.

2. Related Work

Clustering provides an effective way for prolonging the lifetime of WSNs. Heinzelman et al. [5] first proposed a clustering protocol called LEACH for periodical data gathering applications. It is an application-specific data dissemination protocol that uses clustering to prolong the network lifetime. HEED [6] introduced a variable known as cluster radius which defines the transmission power to be used for intracluster broadcast. EEUC [8] and EADUC [9] introduced cluster head competitive algorithms which extend LEACH and HEED by choosing CHs with more residual energy. Both of them achieve well distribution of CHs.

As faults are inevitable in every distributed computer system, especially in WSNs which consist of a large number of capacity-limited nodes, it is important to be able to determine which of them is working and which is faulty. Comparison-based diagnosis is a realistic approach to detect faulty nodes based on the outputs of tasks executed by system nodes. The model is based on comparisons of the outcomes returned by different units executing the same task and uses the invalidation rule of the generalized Maeng and Malek (gMM) model [14, 15]. Comparison-based diagnosis initially used for multiprocessor system has been firstly applied to mobile ad hoc networks (MANETs) by Chessa and Santi [16]. Later, Elhadef et al. considered the problems of self-diagnosis of wireless mesh networks (WMNs) and MANETs using the comparison approach [17, 18].

For WSNs, traditional comparison-based fault diagnosis protocols for multiprocessor systems, WMNs and MANETs are not suitable without changing. To the best of our knowledge, so far fault diagnosis based on the comparison model has not yet been applied to WSN efficiently. Chen et al. proposed a distributed fault-detection algorithm to locate the faulty sensors [19]. It calculated the measurement difference between neighbor sensors at different times to find if the current measurement of a sensor is different from its previous measurement. Wang et al. provided a cluster-based real-time fault diagnosis aggregation algorithm for WSNs [20]. The protocol is based on the comparison approach aiming at achieving a correct and complete diagnosis for hierarchical WSNs. They assumed that each sensor can transmit data to any other sensor and can communicate directly with the BS, which is unrealistic in practice.

3. System Models

3.1. Network Model. In this paper, we consider a sensor network consisting of N static and homogeneous sensor nodes uniformly deployed over a vast field to continuously monitor the environment. The communication topology of WSN is usually represented by the graph $G = (V, E)$, where each vertex $v \in V$ represents a sensor node and each edge $(u, v) \in E$ represents a communication link. For any vertex $v \in V$, $N(v)$ is the set of all vertices that are adjacent to v in G. We denote the ith sensor by s_i and the corresponding sensor node set $S = \{s_1, s_2, \ldots, s_N\}$. Assume that links are bidirectional in nature, which may be realized using two unidirectional links. We denote a bidirectional link between nodes s_i and s_j as $s_i - s_j$, while the directed link from s_i to s_j is denoted by $s_i \rightarrow s_j$. When a link fails, it means that both directed edges have failed. For graph terminology and notation not defined here we refer the reader to [21]. Moreover, we make the following assumptions about the sensor nodes and the underlying network model.

(1) There is a unique identifier for every node. The computing, storage, and energy power of sensors are limited. Nodes are capable of operating in an active mode or a low-power sleeping mode.

(2) There is a stationary base station (BS) located far from the sensing field. BS distributes control messages in one-hop mode, and its energy and computing capability are not limited.

(3) Nodes are location-unaware, but a node can compute the approximate distance to another node based on the received signal strength, if the transmitting power is known.

(4) All nodes are static and homogeneous which are organized as clusters. CMs communicate with CH with one-hop manner, while the communication between CHs and BS is relayed by other CHs.

(5) Proper data aggregation mechanism is adopted for energy saving, and there exists a MAC protocol which is executed to solve contentions, providing reliable one-hop broadcast over logical links.

All of these assumptions are typical for wireless sensor networks, which means that our model is general, that is, not unrealistic. We use a simplified model for the communication energy dissipation [22]. Both the free space (d^2 power loss) and the multipath fading (d^4 power loss) channel models are used, depending on the distance between the transmitter and receiver. The energy spent for transmission of an l-bit packet over distance d is

$$E_{\mathrm{Tx}}(l,d) = \begin{cases} l \times E_{\mathrm{elec}} + l \times \varepsilon_{\mathrm{fs}} \times d^2, & d < d_0 \\ l \times E_{\mathrm{elec}} + l \times \varepsilon_{\mathrm{mp}} \times d^4, & d \geq d_0. \end{cases} \quad (1)$$

The electronics energy, E_{elec}, depends on factors such as the digital coding, modulation, whereas the amplifier energy, $\varepsilon_{\mathrm{fs}}d^2$ or $\varepsilon_{\mathrm{mp}}d^4$, depends on the transmission distance and the acceptable bit error rate. To receive this message, the radio expends energy:

$$E_{\mathrm{Rx}}(l) = l \times E_{\mathrm{elec}}. \quad (2)$$

3.2. The Diagnosis Model. Each node in the system can be in one of two states: faulty or fault-free. There are different classifications for faulty type. Based on duration, faults can be classified as permanent, intermittent, and transient. A transient fault will eventually disappear without any apparent intervention, whereas a permanent one will remain unless it is repaired and/or removed by external administrator. Based on how a failed node behaves once it has failed, faults can be either hard or soft. When a node is hard-faulted, it cannot communicate with the rest of the system. In WSNs, a node can be hard-faulted either because it is crashed or due to battery depletion. Soft faults are subtle, since a soft-faulted node continues to operate and to communicate with the other nodes in the system, although with altered behaviors. In this paper, we utilize the invalidation rule of the gMM model [14, 15] that is summarized in Table 1.

In the gMM model, diagnosis is based upon comparison of the results generated by test tasks assigned to pairs of units with a common neighbor. Let u be a unit adjacent to both unit v and w. If nodes u, v, and w are fault-free, then the results agree and the comparison outcome is 0. If unit u is fault-free and any unit v or w is faulty, then the results disagree and the comparison outcome is 1. If unit u is faulty, then the comparison outcome may be not reliable (0 or 1), regardless of the state of v and w. Assuming that the topology of the network does not change during the diagnosis executing, comparison-based approach relies on the following operations.

(1) *Test Request Generation.* In order to test adjacent nodes, each node u generates a test sequence number i, a test task T_i, the expected result $R_{u,i}$ and sends the test request message TEST_REQ(u, i, T_i) to its neighbors $N(u)$ at time t. Then, node u sends a message T_{out} to initiate the timer. Node u expects to receive response message that comes from its neighbors within this time bound.

Table 1: The invalidation rule of the gMM model.

u	v	w	Comparison result of v and w by u
Fault-free	Fault-free	Fault-free	0
Fault-free	Faulty	Fault-free	1
Fault-free	Fault-free	Faulty	1
Fault-free	Faulty	Faulty	1
Faulty	Any	Any	Not reliable

(2) *Test Request Reception.* Any node v in $N(u)$, upon receiving TEST_REQ, generates the test result $R_{v,i}$ for T_i and sends test response message TEST_RES$(u, i, R_{v,i})$ to $N(v)$ at time t', with $t < t' < t + T_{\mathrm{out}}$. Note that (u, i) is known as header of message, which is used to uniquely identify the test task and the sender.

(3) *Test Response Reception.* Any node w in $N(v)$ at time t', upon receiving TEST_RES$(u, i, R_{v,i})$, does the following.

Case 1. If $w = u$, that is, w is the testing node itself, it compares $R_{v,i}$ with the expected result $R_{u,i}$ and generates the comparison outcome. Node v is diagnosed as fault-free if the outcome is 0, as faulty otherwise.

Case 2. If $w \neq u$, this means that w is not the testing node, we should check whether w is u's neighbor. The following two cases arise.

Case 2.1. $w \in N(u)$ at time t. In this case, $w \in N(v) \cap N(u)$, that is, w and the tester node u share at least one common adjacent node v. Node w received the test request TEST_REQ from u and the test response TEST_RES from v, hence it can compare $R_{v,i}$ with $R_{w,i}$. Node v is diagnosed as fault-free if the comparison outcome is 0, as faulty otherwise. Figure 1(a) illustrates this case.

Case 2.2. $w \notin N(u)$ at time t. If there exists some $z \in N(u)$ such that $R_{z,i} = R_{v,i}$ then both nodes are diagnosed as fault-free; otherwise, if node z (node v) has been diagnosed as fault-free, then node v (node z) is diagnosed as faulty. Otherwise, the test result $R_{v,i}$ is stored. Figure 1(b) illustrates this case.

(4) *Timeout Reception.* After sending its test response message, node u initiates a timer to T_{out} in order to guarantee that its neighbors will response within this time bound. Once this bound expires, the testing node u receives the timeout message from the timer and diagnoses all the nodes that did not reply to the test request as faulty.

3.3. Problems and Requirements. The inherent characteristics of WSNs make the direct applications of existing fault diagnostic models that have been pervasively applied in

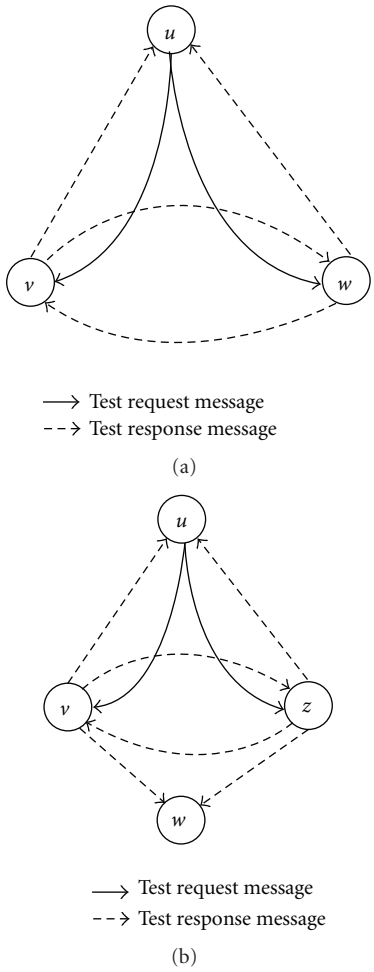


Figure 1: (a) Node w received u's test task and v's corresponding response reply. (b) Node w received v and z's response corresponding to u's test task.

traditional multiprocessor systems, WMNs, and MANETs are unrealistic. In the following, we sum up these problems and give the corresponding analysis.

On the one hand, we require a suitable fault diagnosis mechanism for WSNs which is energy-efficient and message complexity acceptable. Sensor nodes are limited in battery energy, computing, and storage capacity, and the protocol design need to consider the energy efficiency of the network nodes. That is, the amount of calculation performed by nodes should be moderate and the traffic generated by the exchange of messages should be reasonable. Clearly, there should be some mechanisms to isolate messages flooding in the networks. Owing to instability of nodes, the number of faulty nodes is far more than the traditional multiprocessor systems or MANETs. Besides, constrained by wireless communication capacity, the structure properties in terms of regularity, connectivity, and so forth, of the communication topology are worse than interconnection networks. As a result, the traditional diagnostic measures cannot meet the fault diagnosis requirements in WSNs.

On the other hand, an excellent clustering mechanism provides a good basis for fault diagnosis. Clustering WSNs, CMs only perform the sensing tasks, while in addition to collecting data from CMs, CHs also have to fulfill a variety of other tasks, such as data aggregation, cluster maintenance, and communication with BS via multihop or one-hop way. Considering the efficient mechanisms for CHs to reduce and balance energy consumption and to ensure the CHs fault-free is crucial, especially in fault diagnosis applications. Besides, nodes are left unattended after deployment requires adaptive node fault processing mechanism. For instance, the fault conditions can be transmitted to the BS, which are uniformly processed according to the actual requirements.

In this paper, we summarize the protocol design as clustering problem based on fault diagnosis and multipath routing problem based upon DAG. Since the networking nodes in WSNs are very limited in resources, clustering should not only have small size, but also be constructed with low communication overhead and computation cost. In addition, the amounts of communication and computation should be scalable as the networks are typically deployed with large network size. For the multipath routing problem, the backbone network composed of CHs can be abstracted as an edge and vertex weighted graph $G(V, E, W, R)$. We view the CHs as vertexes set V, the communication links as edges set E, the communication cost (edge weight) as W: $E(G) \rightarrow R^+$, and the residual energy (vertex weight) as R: $V(G) \rightarrow R^+$. In order to provide solutions for the two problems, we believe the following requirements should be met.

(1) Clustering should be completely distributed with accepted message complexity. Each node independently makes its decisions based on local information and results into well-distributed CHs over the sensing field.

(2) At the end of clustering, each node is either a cluster head or a member node.

(3) Using message complexity acceptable diagnostic mechanism to eliminate the harmful influence and to avoid unnecessary energy consumption imposed by faulty nodes.

(4) Utilizing multipath routing mechanism to make the gathering data transmission reliable. In particular, it is necessary to select the cost-aware or energy-aware communication path among all of the possible paths.

(5) Avoiding excessive energy consumption of CHs and maintaining the energy consumption balance of network nodes.

4. System Design

Our fault diagnosis-based clustering and multipath routing data collection protocol (FDCM) can be divided into two phases mainly as follows. Phase (I): clustering construction based on fault diagnosis; Phase (II): resilient multipath routing selection. In the following, we explain how FDCM works in detail.

4.1. Clustering Construction Based on Fault Diagnosis. In network deployment phase, BS broadcasts a HELLO message to all the nodes in the network at a certain power level which includes a certain number of candidate CHs selected in advance. After receiving this message each sensor node checks whether it is a candidate CH. Each selected CH computes the approximate distance to the BS based on the received signal strength and then executes a distributed cluster head competitive algorithm similar to EEUC [8]. Our CH competition is primarily based on the fault status and residual energy of candidate CHs. The size of cluster is controlled by competition radius R, which is a constant tuned by typical situation. In addition, let s_i denote a CM and c_i represents any cluster i, respectively.

Definition 1. Candidate CH s_i's adjacent CH set N_{CH} is given by $s_i.N_{\text{CH}} = \{s_j \mid s_j \text{ is candidate CH, and } \text{dist}(s_i, s_j) < R\}$. Furthermore, if the nodes in $s_i.N_{\text{CH}}$ are fault-free, then the set is denoted by $s_i.N_{\text{FF\_CH}}$, otherwise, $s_i.N_{F\_\text{CH}}$. Obviously, $s_i.N_{\text{CH}} = s_i.N_{\text{FF\_CH}} \cup s_i.N_{F\_\text{CH}}$.

Before clustering, BS selects predefined candidate CHs randomly on certain probability to compete for final CHs. For the sake of saving energy, nodes that fail to be candidate CHs keep sleeping until the cluster head competition stage ends.

The distributed clustering algorithm which is initiated by BS and executed by each candidate CH is presented in Algorithm 1. First, each candidate CH broadcasts a COMPETE_CH(s_i.ID, $s_i.R$, s_i.RE) message which contains its node ID, competition radius R, and residual energy RE. After the construction of N_{CH} has finished in lines 2–4, each candidate CH checks the fault status of its N_{CH} based on comparison model approach in lines 5–24. Candidate CH s_i generates a test sequence number i and the correspondent test task T_i and sends a test request message TEST_REQ(s_i.ID, i, T_i) to its adjacent candidate CH set $s_i.N_{\text{CH}}$. Node s_i waits for the responses of $s_i.N_{\text{CH}}$ and diagnoses their status according to the comparison model. Lines 25–36 describe the CH competition process. The candidate CHs with faulty status (soft-faulty nodes) have no qualification for the competition. If s_i belongs to $s_i.N_{\text{CH}}$ and s_i receives a FINAL_CH message from s_j, then s_i will give up the competition immediately. After that the fault-free candidate CH makes a decision whether it can act as a final CH. In particular, if the constructed adjacent set N_{CH} is null, then the candidate CH becomes final cluster head immediately. Once fault-free s_i finds that its residual energy is more than all the nodes in its S_{CH}, it will win the competition.

After all the final CHs have been elected, immediately, previous sleeping nodes now are waked up and each CM chooses their closest CH with the largest signal strength received. All the CMs register with the CH by sending a JOIN_CLU message. In order to determine the status of CMs in each cluster, the CH sends a test quest TEST_REQ to its member nodes. These nodes compute the tasks and feed back the results to the sender. The lower layer fault diagnosis algorithm based on comparison protocol is presented in Algorithm 2. Once the faulty nodes are determined, they will be ordered to turn dead. The final CH sets up a TDMA schedule and transmits it to the nodes in the cluster. After the TDMA schedule is known by all nodes in the cluster, the clustering phase is completed and the data transmission stage begins. Based on the execution of Algorithm 1, we can draw the following theorem.

Theorem 2. *In clustering stage, FDCM has a message exchange complexity of $O(1)$ per node and $O(N)$ for entire network.*

Proof. During the execution of clustering algorithm, each candidate CH sends a COMPETE_CH message at first. In order to identify the faulty candidate CHs, which will be deprived of the eligibility of final CHs, each of them sends TEST_REQ and receives TEST_RES message one after another. If it becomes a final CH then broadcasts a FINAL_CH message to declare its win, otherwise broadcasts a QUIT_ELECT message to exit. After the declaration of winning the election, each regular node broadcasts a JOIN_CLU message. So each node has the message complexity of $O(1)$.

Assume that network size is N, the number of candidate CHs is M and the number of final CHs/clusters is N_c ($N_c \leq M \leq N$). In clustering stage, the overall message overhead is $3M + N_c + (M - N_c) + (N - N_c) = 4M + N - N_c = O(N)$.

The theorem is proved. □

From Theorem 2 we can conclude that the clustering stage has a low message complexity both for individual node and entire network, thus requirement (1) is satisfied.

Theorem 3. *At the end of clustering phase, any fault-free node either is a final CH or a CM. Furthermore, each CM exactly belongs to a cluster, and only one final CH is allowed in each competition range.*

Proof. During the execution of Algorithm 1, for the nodes in sensor network there are at most four states in total: *RegularNode*, *CandidateCH*, *FinalCH* and *deadNode*. Here the status of *RegularNode* and *FinalCH* means it is a CM and CH, respectively.

In the following we first show that any node is either a final CH or a CM after execution of Algorithm 1. Initially, in addition to the selected candidate CHs in advance, the remainder nodes are all regular nodes. For candidate CHs, in lines 5–24, each of them knows the fault status of its adjacent CHs. If the node is determined as faulty nodes, then they quit the competition process, immediately. In lines 33–36, the faulty candidate CHs are ordered to turn dead (*deadNode*), and they do not participate in the subsequent work any more. For any candidate CH s_i, if it has not any adjacent node, then Algorithm 1 executes line 26 and s_i becomes a final CH at once. Furthermore, in lines 25–36 s_i either becomes a final CH (*FinalCH*) or becomes a CM (*RegularNode*) mutually exclusive.

After the election of final CHs has finished, each CM registers with only one CH based on received signal strength, thus each CM exactly belongs to a cluster. The competition process shows that for any candidate CH s_i's adjacent CH

1. Candidate CH s_i broadcasts a competition message COMPETE_CH(s_i.ID, s_i.R, s_i.RE);
2. On receiving a COMPETE_CH from s_j;
3. **if** (dist(s_i, s_j) < R) **then**
4. $s_i.N_{CH} \leftarrow s_i.N_{CH} \cup s_j$;
5. s_i generates test sequence number i and the test task T_i
6. s_i broadcasts a TEST_REQ(s_i.ID, i, T_i);
7. On receiving a TEST_REQ(s_j.ID, j, T_j) from s_j;
8. s_i generates the test result $R(s_i, j)$ for task T_j;
9. s_i broadcasts the Test_RES(s_j, j, R_{si}) to $s_i.N_{CH}$;
10. On receiving a TEST_RES from s_j initiated by s_k
11. **if** ($s_k = s_i$) **then**
12. **if** ($R(s_j, j) = R(s_i, j)$) **then**
13. $s_i.N_{FF\_CH} \leftarrow s_i.N_{FF\_CH} \cup \{s_j\}$;
14. **else if** $s_i.N_{F\_CH} \leftarrow s_i.N_{F\_CH} \cup \{s_j\}$;
15. **else if** ($s_k \in s_i.N_{CH}$) **then**
16. **if** ($R(s_j, j) = R(s_i, j)$) **then**
17. $s_i.N_{FF\_CH} \leftarrow s_i.N_{FF\_CH} \cup \{s_j\}$;
18. **else** $s_i.N_{F\_CH} \leftarrow s_i.N_{F\_CH} \cup \{s_j\}$;
19. **else if** ($s_k \notin s_i.N_{CH}$)
20. **if** (s_i has received a TEST_RES from $s_z \in s_i.N_{CH}$ and $R(s_j, j) = R(s_z, j)$) **then**
21. $s_i.N_{FF\_CH} \leftarrow s_i.N_{FF\_CH} \cup \{s_j, s_z\}$;
22. **else if** (s_i received a TEST_RES from $s_z \in s_i.N_{CH}$ and $R(s_j, j) \neq R(s_z, j)$ and $s_z \in s_i.N_{FF\_CH}$) **then**
23. $s_i.N_{F\_CH} \leftarrow s_i.N_{F\_CH} \cup \{s_j\}$;
24. **else** store the test response;
25. **if** ($s_i.N_{CH}$ = NULL) **then** s_i.state ← *finalCH*;
26. **while** (s_i.state = *candidateCH*)
27. **if** ($s_j \in s_i.N_{FF\_CH}$ and $s_i.E_{RE} > s_j.E_{RE}$) **then**
28. broadcast FINAL_CH(s_i.ID);
29. s_i.state ← *finalCH*;
30. On receiving a FINAL_CH from s_j;
31. **if** ($s_j \in s_i.N_{CH}$) **then**
32. broadcast QUIT_ELECT(s_i.ID);
33. s_i.state ← *RegularNode*;
34. On receiving a QUIT_ELECT from s_j
35. **if** ($s_j \in s_i.N_{CH}$) **then**
36. remove s_j from $s_i.N_{CH}$;
37. **end while**

Algorithm 1: The distributed CH election algorithm based on fault diagnosis executed by candidate CH s_i.

set N_{CH} if s_i wins the competition, then the nodes in $s_i.N_{CH}$ quit competition. Otherwise, if s_i receives a message, it quits competition too, so only one final CH is allowed in each competition range.

To sum up, the theorem is proved. □

Fault diagnosis in this paper refers to all faulty nodes within the sensor network are identified correctly and these faulty nodes are ordered to stop working, and the fault conditions about each cluster are reported to BS via data transmission by CHs. Upon receiving fault information, BS takes appropriate actions, such as forbid faulty nodes taking part in the final CH election in the next round. According to the description and analysis above, the fault diagnosis process consists of two phases. First, it eliminates the candidate CHs to participate in the final CH competition in the process

1. generate test task T_j with sequence number j;
2. broadcast test request TEST_REQ(c_i.CH, j, T_j) to its member nodes c_i.CMs;
3. generate the expected result $R(c_i.CH, j)$ for T_j;
4. set the timer to be T_{out};
5. Initialize the faulty CM set $c_i.S_{F\_CM}$ and fault-free CM set $c_i.S_{FF\_CM}$ with NULL;
6. Each CM sends test response TEST_RES(c_i.CM, j, T_j) in random back-off time $t' < T_{out}$;
7. **while** receiving test response TEST_RES from any node c_i.CM in c_i.CMs
8. **if** ($R(c_i.CH, j) = R(c_i.CM, j)$) **then**
9. $c_i.S_{FF\_CM} \leftarrow c_i.S_{FF\_CM} \cup \{c_i.CM\}$;
10. **else**
11. $c_i.S_{F\_CM} \leftarrow c_i.S_{F\_CM} \cup \{c_i.CM\}$;
12. **end while**
13. send TURNDEAD message to $c_i.S_{F\_CM}$;

Algorithm 2: Lower layer diagnosis algorithm executed by cluster head c_i.CH.

of clustering. Secondly, after the clustering is finished, the fault diagnosis is done based on special comparison model between CH and CMs. These two phases together complete the diagnosis of all faulty nodes in the network correctly. A diagnostic message can be a test request, a test response, or a timeout message. The message complexity of diagnostic algorithm is presented in Theorem 4.

Theorem 4. *The communication complexity of our comparison-based fault diagnostic approach is* $O(N+M(1+d_{max}))$, *where* M *and* d_{max} *denotes the number of candidate CH and its maximum degree, respectively.*

Proof. According to Theorem 3, at the end of the clustering stage, any fault-free node is either a final CH or a CM. Each candidate CH s_i generates at most one test request TEST_REQ. In turn, the test request generates at most $|s_i.N_{CH}| < d_{max}$ test responses. The communication complexity during CH competition is $O(M(1 + d_{max}))$. The fault diagnosis of CMs is done by CH generates and sends test request TEST_REQ. The test request sent by CH generates N_c test responses. CHs obtain the fault status of their CMs by comparison of test tasks. The communication complexity of each cluster is $O(N_c)$ and for entire network, it is $O(N)$. Thus, the overall communication complexity is $O(N+M(1+d_{max}))$. □

4.2. Resilient Multipath Routing. Before delivering their data to BS, each CH first aggregates the sensing data from its CMs, and then sends the data packet plus the fault information to BS via a multihop fault-tolerant path. We assume that any two CHs within their communication scope can communicate with each other and consider them as neighboring. Each CH has the information about its neighbors. The distributed multipath routing consists of three subphases: Construct Connected Network, Choose Next-hop Neighbor and Route Maintenance.

4.2.1. Construct Connected Network. Assume c_i.CH is the source CH, from where the sensing data and fault information is aggregated; c_j.CH is a candidate relay CH. The factors that are taken into account by each CH for constructing communication path are summarized as:

Condition 1. $\text{dist}(c_i.\text{CH}, c_j.\text{CH}) < 2R$: the communication radius is defined as two times of cluster radius. The distance relation ensures that connectivity of constructed network (graph), since any two CHs have communication link (edge) only if they meet with this condition.

Condition 2. $\text{dist}(\text{BS}, c_j.\text{CH}) < \text{dist}(\text{BS}, c_i.\text{CH})$: the distance relation ensures that the constructed routing paths moving toward BS from the source c_i.CH. The selection of any relay CH always approaches BS geographically, which provides direction for data transmission.

Condition 3. $c_i.\text{CH}.E_{\text{RE}} > E_{\text{relay}}$: the energy relation ensures that the relay nodes should have enough residual energy for data transmission in practice. Relay nodes' residual energy is greater than the energy sum of receiving and sending date packets. Therefore, for balancing network energy consumption, it is necessary to protect the relay nodes' residual energy and give priority to the use of the relay nodes with more remaining energy.

According to the model described in Section 3, we have mapped multipath routing problem into finding communication path in weighted graph $G(V, E, W, R)$. On the basis of this mapping, we assume that there is a logical communication link between any two nodes which meet conditions 1–3 synchronously. Note that the links direction is always from sources to BS, thus a directed connected cost network (vertex and edge weighted digraph) is built. In order to model and depict mapped multipath routing problem, the definition of DAG is given followed by a theorem which satisfies requirement (4).

Definition 5. A directed acyclic graph (DAG) is a directed graph with no directed cycles. We say that a DAG is rooted at r if it is the only node in the DAG that has no outgoing edges. Every other node has at least one outgoing edge.

Theorem 6. *The mapped weighted graph G(V, E, W, R) which meets Conditions 1 and 2 synchronously is a connected DAG rooted at BS. That is, the graph is connected, directed and without directed cycles. Furthermore, for any vertex in G, it is BS reachable.*

Proof. As stated before, there is an edge between any two vertexes which meet with Condition 1. For any two adjacent vertexes u and $v \in V$, edge $(u, v) \in E$. Condition 2 ensures that the direction of edge (u, v) is always from u to v if $\text{dist}(u, \text{BS}) \geq \text{dist}(v, \text{BS})$, according to distance relation. Therefore, graph G is directed.

Suppose that the mapping produces at least two connected components $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that any $v_1 \in V_1$ cannot communicate with any $v_2 \in V_2$. Note that a connected component is a maximal connected subgraph of G. Without loss of generality, assume that V_2 lies on the right of V_1 and $\text{BS} \in V_2$, we have the following two cases.

Case 1. There is not any relay CH in V_2 such that v_1 reaches BS via it. There must be a cluster head $v_1 \in V_1$ which is able to communicate with $v_2 = \text{BS} \in V_2$.

Case 2. There exists at least a relay CH $v_2 \in V_2$ such that v_1 reaches BS via v_2. Then v_1 communicates with it firstly.

So both cases there must be at least a directed edge (v_1, v_2) connects these two connected components, which contradicts with the initial assumption that a cluster head in one component cannot communicate with the one in the other component. Therefore, V_1 and V_2 are connected.

Condition 2 ensures that the route moving toward BS from source CHs via relay CHs. Without loss of generality, we suppose that there is a directed circle $C = \langle v_i \rightarrow v_j \rightarrow \cdots \rightarrow v_k \rightarrow v_i \rangle$ in the digraph, so we have $\text{dist}(v_i, v_i) = 0$. Due to condition (2), we have $\text{dist}(v_i, v_j) > 0, \ldots, \text{dist}(v_k, v_i) > 0$. The total distance on the directed circle C is $\text{dist}(v_i, v_j) + \cdots + \text{dist}(v_k, v_i) > 0$. Since $\text{dist}(v_i, v_j) + \cdots + \text{dist}(v_k, v_i) = \text{dist}(v_i, v_i)$, thus we have $\text{dist}(v_i, v_i) > 0$. This is a contradiction.

Therefore, the theorem is proved. □

4.2.2. Next-Hop Neighbor Choosing. The reachability relation in a DAG forms a partial order, with which a routing path can be constructed, that is, the DAG provides a multipath routing mechanism. With Theorem 6, any CH can transmit its data along relay CHs to BS. Any node or link fails, based on the candidate nodes and links in DAG another one will be chosen. According to different requirements, the selection of the next hop node gives priority to the minimum energy strategy or the maximum residual energy strategy or any other factors. The former forwards packets along the minimum energy path to BS; while the latter farces packets to move toward the BS considering more residual energy of the node on routing path. The decision is made depending on the connected network model and the information gathered in clustering stage.

As an example, the distributed algorithm looking for next-hop neighbor for any cluster head c_i.CH is presented in Algorithm 3. Each CH chooses its next-hop neighbor independently according to the distance to BS. Initially, c_i.CH chooses a neighbor, which is the nearest to BS within its communication range. In lines 5–9, if more than two neighbors have the same distance, then algorithm selects the one with more residual energy for the sake of balancing energy once again. When the cluster head cannot choose its neighbor any more, the network becomes partitioned. From the view of entire network, a spanning tree with root BS which has minimum hop counts to BS is received.

Figure 2 illustrates the communication path selection process. Each cluster head has 2 J initial energy, and the communication cost is denoted by edge weight, the residual energy is represented by vertex weight. The DAG shows the available communication paths. For sensor node v_4, if the minimum energy first, it selects v_3, v_6 as the next hop node

```
1. while network is not partitioned;
2.  for any c_j.CH ∈ c_i.N_FF_CH
3.      c_k.CH = min(dist(c_j. N_CH, BS));
4.      c_i.CH.nextHop ← c_k.CH;
5.      if dist(c_j.CH, BS) = dist(c_k.CH, BS) then
6.       if (c_i.CH. E_RE > c_k.CH. E_RE) then
7.          c_i.CH.nextHop ← c_j.CH;
8.       else
9.          c_i.CH.nextHop ← c_k.CH;
10. end while
```

Algorithm 3: Distributed multihop routing selection for cluster head c_i.CH.

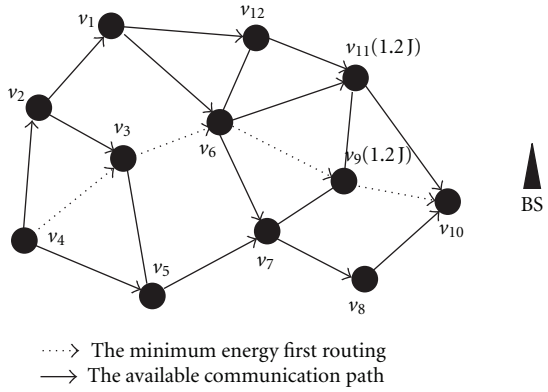


Figure 2: Intercluster multihop paths selection among DAG.

one after another. Note that when node v_6 looking for its next hop node and find that there are two nodes v_9 and v_{11} with the same distance to BS. At this moment, it selects node v_9 which has more residual energy. On the contrary, if the highest residual first, algorithm not necessarily selects the paths which approach BS more quickly but have more residual energy. Finally, from the global view, algorithm outputs a relative optimal communication spanning tree about the graph G whose root is BS.

4.2.3. Route Maintenance. As stated previously, in the connected network each CH always chooses the neighboring CH with the minimum distance to BS as the next-hop routing node independently. The rest neighbors are maintained in its routing table in order of their distance to BS. If the optimal neighbor is unavailable due to node or link failure, then the node chooses a suboptimal one in its routing table, thus providing a robust routing. In addition, this multipath routing selection mechanism to some extent guarantees that the malicious attacker in network cannot obtain the communication path by listening in the signal simply. After each CH determined its next-hop neighbor, CHs are ready to start transmitting sensing data.

In intercluster multihop routing stage, Algorithm 3 finds a routing path, which approaches BS more quickly among all the available paths. While in clustering stage, candidate CH competition takes into residual energy into account. In round based protocol, both stages progress alternatively and thus provide network balanced energy consumption among all sensor nodes which meets requirement (5).

Table 2: Simulation parameters.

Parameter	Value
Sensor field	100 m × 100 m
BS location	(170,50)
Number of nodes	100
Initial energy of nodes	2 J
Data packet size	500 bytes
E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/(bit·m^2)
ε_{mp}	0.0013 pJ/(bit·m^4)
R	30 m
d_0	86 m

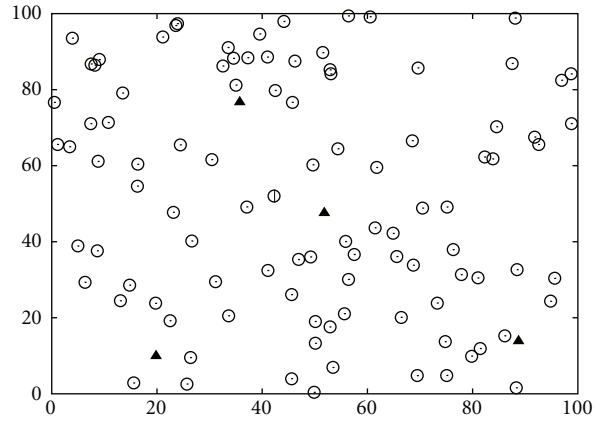


Figure 3: Wireless sensor network deployment. Hollow circles denote fault-free nodes, and triangles show faulty nodes.

5. Simulations

5.1. Simulation Settings. In this section, we evaluate the performance of FDCM with simulations. Because LEACH [5], HEED [6], and EEUC [8] are the most similar clustering protocols, we use them for comparisons. For fault diagnosis-based clustering, CRFDA [20] is compared. One hundred of sensor nodes are randomly distributed over the region of 100 m × 100 m as showed in Figure 3. The number of candidate CHs is set as 20% of the total nodes. The BS is located far away from the region, at point (50, 175). The simulation parameters are listed in Table 2 .

In our paper we make the following assumptions. (1) Nodes that are detected as faulty will turn into dead mode, that is, they will no longer generate information and consume energy. (2) During the network lifetime, nodes may be faulty at any time. The data sending by soft-faulty nodes is invalid. (3) Sensor nodes have idealized sensing capabilities. Ideal MAC layer conditions are assumed, that is, perfect transmission of data on a node-to-node wireless link. (4) In diagnosis process, we use the uniform rules to generate test Task T_i, and ignoring the energy consumed by its implementation.

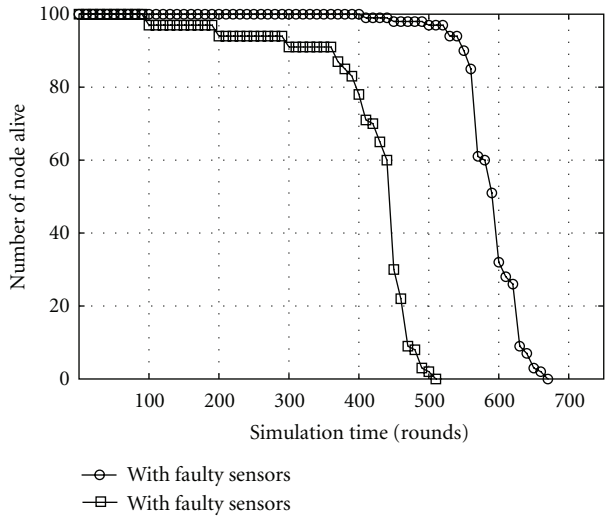


Figure 4: The number of fault-free nodes changes with time.

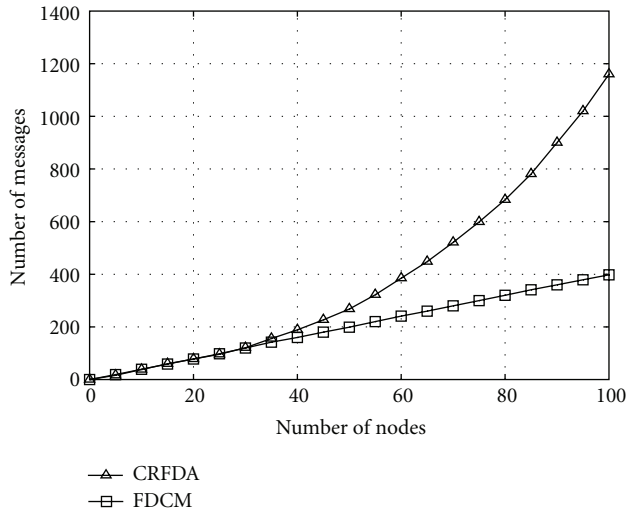


Figure 5: Comparison of communication complexity.

5.2. Simulation Results. Since our protocol is round based, we assume that sensor network randomly generates certain number of faulty nodes when it is running in a certain round. In ideal situation, we expect that the number of the fault-free nodes in network decreases correspondingly comparing with previous round. In our simulation, assume that 3 faulty nodes present in 100th, 200th, 300th round, respectively, and then the result of algorithms execution is shown in Figure 4. One of the curves indicates the result when there are faulty nodes in the network; while the other is fault-free case. From the figure, we can see that the number of fault-free nodes is reduced with faulty nodes arise when the network is running in each round. This means FDCM can correctly detect the faulty nodes in the network. Besides, by comparing, the existence of faulty nodes decreases the network lifetime observably.

Communication complexity is an important measurement for fault diagnosis efficiency. That is one of reasons why fault diagnosis mechanisms with high complexity in traditional MANETs are unrealistic using for WSNs directly. By means of introducing fault diagnosis based on the comparison model in the process of the candidate CHs running for final CHs, FDCM eliminates the chance of faulty nodes to participate in the election; Then, within each cluster only $O(N)$ message complexity needed to complete the fault diagnosis of the cluster members. The isolation of diagnosis boundary avoids large-scale message diffusion throughout the entire network, which made the message complexity reduced significantly. Figure 5 confirmed this by comparing the message complexity of different protocols. It shows that the message complexity of FDCM increases nearly linearly. When faulty nodes come into being, if they do not be diagnosed and excluded from the network, they will consume additional energy of other fault-free CHs. Then, it will shorten the lifetime of the entire network. In simulation, we monitor the number of nodes alive changing with time (round). We find that the network lifetime of FDCM may be influenced by various factors in different situations. The evaluation of the lifetime comparison between the cost-aware and the energy-aware is shown in Figure 6.

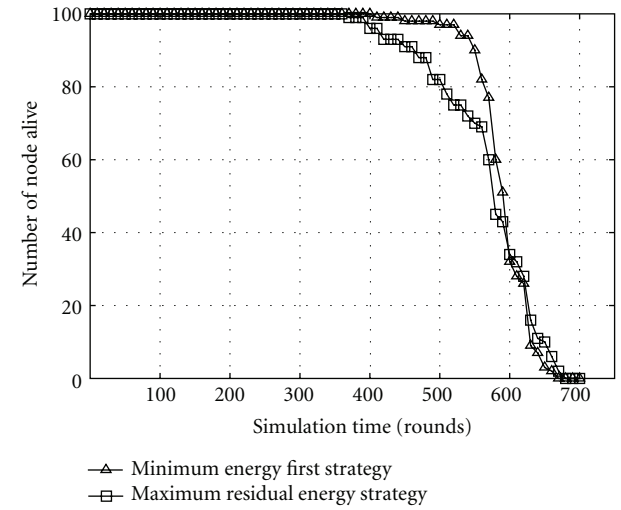


Figure 6: Comparison of network lifetime with different routing modes.

In contrast with similar clustering protocols, we run LEACH, HEED, and EEUC to compare their performance in network lifetime. As shown in Figure 7, FDCM and EEUC perform far better than LEACH and HEED in prolonging network lifetime attributed to the consideration of energy conservation. In FDCM, a certain amount of energy is spent by the nodes involving in fault diagnosis; however, this eliminates additional energy consumption caused by the faulty nodes. More importantly, the existence of fault diagnosis ensures the correctness of the information collected.

6. Conclusions

In wireless sensor network, it is of great importance for fault diagnosis to ensure the gathering information accuracy and

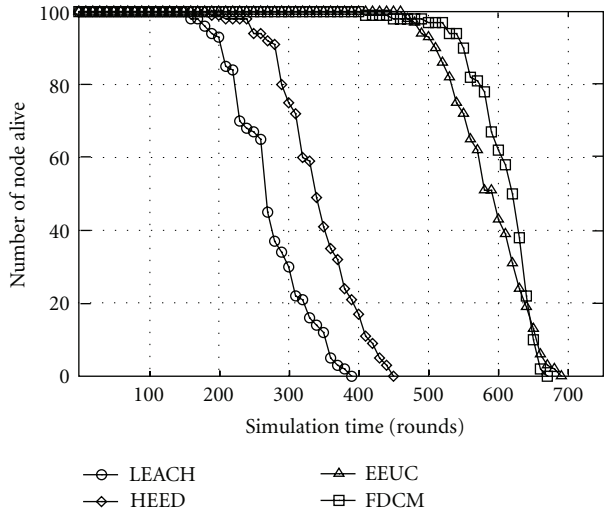


Figure 7: Comparison of Network lifetime with different protocols.

reduce energy additionally consumed by faulty nodes, for the deployment of a large number of sensor nodes in hostile environment. For the inherent characteristics of sensor networks, this paper analyzes the issues and challenges of comparison based-fault diagnosis model for wireless sensor networks and gives the relevant design requirements.

As a complete data collection protocol, the proposed protocol mainly consists of fault diagnosis-based clustering and multipath routing. During clustering stage, a fault diagnosis approach based on comparison model is introduced. Fault diagnosis of the network nodes consists of two phases. At first, it eliminates the faulty candidate CHs to participate in the final CH competition in the process of clustering. Secondly, after the clustering is finished, the fault diagnosis is done based on special comparison model between CH and CMs. CH sends a test request message to its members and according to their responses to determine the fault status of these nodes, failure nodes are ordered to turn dead. These two phases together complete the diagnosis of all faulty nodes in the network.

In Multipath routing stage, communication characteristics impose certain conditions, which map the original abstract communication graph into the DAG. The new graph determines the feasible multipath communication path of any node to transfer data to the BS. In particular, we give an algorithm greedy select next hop neighbor which has the minimum distance to BS. When multiple nodes are optional, then the node with the highest residual energy is preferential. If any node in the routing path fails, then select an available path in the DAG depending on the highest residual energy or have the minimum distance to BS until the data transfer to the BS. Note that the transmitted data including node fault status, which can be used in the next round as a basis of cluster first election, that is, faulty nodes will lose the possibility of acting as the candidate CH.

For future work, we will consider two new directions. First, we intend to improve our algorithm effectiveness and obtain better performance, such as more accurate diagnosis and lower response time. Second, on condition that acceptable message complexity, we will study the possibility of new diagnosis approach which is appropriate for dynamic topology.

Acknowledgments

This work is supported by National Natural Science Foundation of China (nos. 61070169, 61170021), Natural Science Foundation of Jiangsu Province (no. BK2011376), Specialized Research Foundation for the Doctor al Program of Higher Education of China (no. 20103201110018), Application Foundation Research of Suzhou of China (nos. SYG201034, SYG201240), and sponsored by Qing Lan Project.

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] 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.

[3] H. M. Ammari and S. K. Das, "Promoting heterogeneity, mobility, and energy-aware Voronoi diagram in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 7, pp. 995–1008, 2008.

[4] H. Zhang and H. Shen, "Balancing energy consumption to maximize network lifetime in data-gathering sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, pp. 1526–1539, 2009.

[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] 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.

[7] S. Soro and W. B. Heinzelman, "Prolonging the lifetime of wireless sensor networks via unequal clustering," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, April 2005.

[8] 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.

[9] J. Yu, Y. Qi, G. Wang, and X. Gu, "An energy-aware distributed unequal clustering protocol for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 202145, 8 pages, 2011.

[10] 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.

[11] Z. Ye, S. V. Krishnamurthy, and S. K. Tripathi, "A framework for reliable routing in mobile ad hoc networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (IEEE INFOCOM)*, pp. 270–280, April 2003.

[12] J. Tsai and T. Moors, "A review of multipath routing protocols: from wireless ad hoc to mesh networks," in *Proceedings of the*

ACoRN Early Career Researcher Workshop on Wireless Multihop Networking, pp. 17–22, July 2006.

[13] T. Elhourani and S. Ramasubramanian, "Independent directed acyclic graphs for resilient multipath routing," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 153–162, 2012.

[14] M. Malek, "A comparison connection assignment for diagnosis of multiprocessor systems," in *Proceedings of the 7th International Symposium on Computer Architecture*, pp. 31–36, 1980.

[15] A. Sengupta and A. T. Dahbura, "On self-diagnosable multiprocessor systems: diagnosis by the comparison approach," *IEEE Transactions on Computers*, vol. 41, no. 11, pp. 1386–1395, 1992.

[16] S. Chessa and P. Santi, "Comparison-based system-level fault diagnosis in ad hoc networks," in *Proceedings of the 20th IEEE Symposium on Reliable Distributed Systems (SRDS '01)*, pp. 257–266, New Orleans, La, USA, October 2001.

[17] M. Elhadef, A. Boukerche, and H. Elkadiki, "Self-diagnosing wireless mesh and ad-hoc networks using an adaptable comparison-based approach," in *Proceedings of the 2nd International Conference on Availability, Reliability and Security (ARES '07)*, pp. 983–990, April 2007.

[18] M. Elhadef, A. Boukerche, and H. Elkadiki, "A distributed fault identification protocol for wireless and mobile ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 68, no. 3, pp. 321–335, 2008.

[19] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proceedings of the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS '06)*, pp. 65–71, September 2006.

[20] W. Wang, B. Wang, Z. Liu, and L. Guo, "A cluster-based real-time fault diagnosis aggregation algorithm for wireless sensor networks," *Information Technology Journal*, vol. 10, no. 1, pp. 80–88, 2011.

[21] D. B. West, *Introduction to Graph Theory*, Prentice Hall, New York, NY, USA, 2001.

[22] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS-33)*, p. 10, January 2000.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 852027, 15 pages
doi:10.1155/2012/852027

Research Article

On Guaranteed Detectability for Surveillance Sensor Networks

Yanmin Zhu[1, 2]

[1] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*
[2] *Shanghai Key Lab of Scalable Computing and Systems, Shanghai 200240, China*

Correspondence should be addressed to Yanmin Zhu, yzhu@cs.sjtu.edu.cn

Received 22 February 2012; Revised 21 May 2012; Accepted 26 May 2012

Academic Editor: Shan Lin

Surveillance is an important class of applications for wireless senor networks (WSNs), whose central task is to detect events of interest. Existing approaches seriously suffer from blind spots and low energy efficiency. In this paper, we propose a fully distributed algorithm GAP for energy-efficient event detection for surveillance applications. Employing the probabilistic approach, GAP actively tunes the active probability and minimizes the energy consumption of each sensor. The unique features of GAP are threefold. First, it provides guaranteed detectability for any event occurring in the sensing field. Second, it exposes a convenient interface of the user to specify the desired detectability. Finally, it supports differentiated service to empower better surveillance for critical spots. Without relying on costly time synchronization, GAP is a lightweight distributed protocol and is truly scalable to network scale and sensor density. Theoretical analysis and comprehensive simulation experiments are conducted, which jointly demonstrate that GAP is able to provide guaranteed detectability while significantly prolonging the system lifetime compared with other schemes.

1. Introduction

Recent rapid advances in wireless sensor networks (WSNs) have made it possible to develop a wide multitude of compelling applications, ranging from battlefield surveillance, habitat monitoring, and radiation prevention [1] to pollution detection [2]. Surveillance, whose central task is to detect events of interest, is an important class of applications for WSNs. The most important performance goal for surveillance applications is high detectability. Among the others, fire detection in large-scale forest is a good example of surveillance applications.

It is well known that tiny sensor nodes are subject to stringent energy constraint since they are powered by small batteries. This implies that a sensor node has only a short lifetime. It is usually impractical, if not impossible, to recharge or replace batteries after sensor nodes are deployed to a remote or event hostile environment. However, applications require the network to sustain surveillance operations for a long lifetime. It has been challenging to obtain a long-lived network with tiny short-lived sensor nodes. To achieve high detectability, the straightforward way is to keep all sensors active such that an event is sure to be detected. However, it is obvious that this scheme suffers from low energy efficiency, especially when sensor density is high. In this paper, we focus on energy-efficient surveillance using networked sensors.

The fundamental approach to conserving energy is to power off sensors. Many research efforts have been made for energy-efficient surveillance using sensor networks. In general, they select a subset of sensor nodes to keep vigilant for event detection and put the others in power-save mode. In PEAS [3], a sensor probes neighbors to check if there are active neighbors. Upon receiving acknowledgement from an active neighbor, it goes to sleep. Network provisioning [4] identifies a redundant sensor whose sensing coverage is jointly covered by its active neighbors. Yan et al. [5] noted the underestimation problem that exists in [4] and proposed a randomized algorithm to determine an active schedule of the sensors.

There are two major drawbacks with the existing methods. First, the algorithms' failing to provide full coverage over the sensing field suffers from blind spots that are not covered by any active sensors. Events occurring in these blind spots will not be detected, leading to serious surveillance quality degradation. Second, the algorithms suffer from the critical problem of unbalanced energy consumption. According to

the existing algorithms, a set of sensors are selected to stay active for full sensing coverage while the other sensors turn to power-save mode. The consequence is that the selected active sensors will be depleted earlier. If an active sensor becomes unavailable due to power depletion or physical damage, the area covered by this sensor will become a blind spot.

In addition to identifying the limitations of existing approaches, we have two important observations for many realistic surveillance applications.

(i) It is unnecessary for many applications, such as habitat monitoring, to have perfect detectability (i.e., 100% detectability). For example, a wild animal of interest may cause a number of events in the field. It is sufficient for the animal study to capture some of the events. In practice, different applications usually have varying requirements on detectability. It is clear that more sensitive applications are usually in need of higher detectability.

(ii) An event can occur at any unpredictable time at any place within the sensing field. It is impossible to predict the location where the next potential event occurs.

In response to the two observations, the system design should satisfy two important requirements for such surveillance applications.

(i) The system should allow the users to customize the desirable detectability, given different applications.

(ii) The system should guarantee that the detectability of any event occurring in the sensing field is larger than the requirement posed by the user.

To the best of our knowledge, no existing algorithms can successfully satisfy the requirements mentioned above. In this paper, we propose a novel-distributed algorithm GAP to provide guaranteed detectability for any event in the whole sensing field. The algorithm exposes a convenient interference for the users to specify the desirable minimum detectability. We devise a simple yet effective metric to realize the design goal of providing guaranteed detectability of any potential event. Exploiting the probabilistic approach, the algorithm allows the sensors to be active probabilistically for effective energy conservation. The algorithm actively minimizes the active probability of each sensor, which is also adaptive to its neighborhood of sensor deployment. Energy consumption of the sensors is finely balanced. At the same time; however, the detectability of any event in the sensing field is dynamically maintained.

The contributions we have made in this paper are highlighted as follows.

(1) We develop the fully distributed GAP algorithm that can provide guaranteed detectability for any event. Not relying on costly time synchronization, this algorithm is lightweight and fully distributed, supporting truly scalability with network scale and sensor density.

(2) The GAP algorithm empowers differentiated surveillance in terms of detectability and detection degree, which greatly enhances its practical applicability.

(3) We conduct both theoretical analysis and comprehensive simulations to validate the design and study the performance of the GAP algorithm.

The remainder of the paper is structured as follows. In Section 2, we discuss related work. In Section 3, we introduce the system model and some preliminaries. In Section 4, after detailing the GAP design, we discuss several design issues and present algorithm analysis. To study the performance of GAP, we conduct comprehensive experiments and discuss the results in Section 5. In Section 6, we give some discussions about the algorithm design. Finally, we conclude the paper and introduce future work.

2. Related Work

In this section we review related work and discuss the difference of our work from existing studies.

2.1. Duty Cycling in Sensor Networks. It has been the subject of extensive research to conserve energy in WSNs through power management or duty cycling. With duty cycling, a sensor node periodically enters power-saving mode for energy saving and wakes up for sensing and communication tasks. Three power-saving protocols for mobile ad hoc networks were developed in [4], which provide different tradeoffs between energy efficiency and neighbor discovery latency. Without relying on time synchronization, asynchronous wakeup [6] is advantageous. However, it comes with the cost of increased packet delivery latency. Different tradeoffs between packet delivery latency and energy saving were studied in [7].

Low duty cycling has been recognized as an effective technique to realize operation longevity in sensor networks. In [8], the scheduling problem of multiple tasks in low-duty-cycled sensor networks is studied. In [9], the energy fairness of asynchronous duty cycling sensor networks is explored. Our work in this paper also adopts duty cycling for every saving but has a focus on the determination of duty cycles for each sensor node making sure that the event detection of any possible event is guaranteed.

2.2. Power Management of Sensor Networks. With power management, a subset of sensor nodes are selected for sensing or communication purposes. In surveillance applications, a number of algorithms were proposed to select a subset of sensor nodes to stay vigilant for event detection while the others remain in power save mode. PEAS [3] selects active sensors by active probing. Each sensor probes its neighborhood. If there is an active sensor responding its probing, the sensor decides to sleep; otherwise, it stays active. PEAS does not providing full sensing coverage and therefore suffers from the blind spot problem. Network-provisioning [10] identifies a sensor to be in sleep mode if sensing coverage of this sensor is jointly covered by its active neighbors. Several

efforts, for example, the one in [11], take both sensing coverage and network connectivity into account. These algorithms provide full sensing coverage and meanwhile maintain network connectivity.

Tian and Georgana [4] noted the underestimation problem that exists in [4] and proposed a randomized algorithm to determine an active schedule of each sensor. According to this algorithm, each sensor is activated for event detection periodically. Thus, this algorithm solves, to some extent, the problem of unbalanced energy consumption. A probabilistic approach has been proposed for event detection in the context of object tracking [2], which can also mitigate the problem of unbalanced energy consumption. However, these algorithms cannot provide guaranteed detectability for the sensing field.

Shih et al. [12] propose to use a small-scale sensor network to monitor epilepsy. It is reported that 21 scalp electrodes are needed and 18 data streams or channels are generated. In order to save the energy consumption of the data processing device that is battery powered and attached to a user, they propose an automated way to construct detectors that use fewer channels, and thus fewer electrodes.

2.3. Energy-Efficient Event Detection. As is widely known, event detection [13] is an important class of applications of sensor networks. Exploiting the inherent property of event persistence, some algorithms [14, 15] try to detect events with low duty-cycled sensor networks. In [16], a two-stage optimization was proposed to minimize detection latency. In the first stage, a density control algorithm is applied to select a set of active nodes. In the second stage, an optimization procedure is executed to schedule wakeups of the sensors, which relies on accurate location information. A testbed of 70 sensors was deployed to detect and track the positions of moving vehicles [17]. In this system, 5% of deployed motes serve as sentries and nonsentries operate at a 4% duty cycle. An improved system with a combination of duty cycle scheduling, sentry service, and tripwire service was recently reported in [18]. With low duty cycling, the lifetime of the system can be significantly extended by up to 900%.

Different from these existing studies, our paper specially considers the guarantee of event detection performance while conserving energy consumption on sensor nodes.

2.4. Energy Harvesting in Sensor Networks. Recently, energy harvesting techniques are becoming very promising technology for long-term applications of sensor networks. In [19], Gu et al. point out that it is unnecessary to conserve energy in sensor networks with energy harvest ability, and instead it is more important to balance energy supply and energy consumption. They propose a middleware to control the RF activity with the objective of minimizing communication delay.

Zhu et al. [20] notice the leakage problem of energy capacitors. They propose leakage-aware feedback control techniques to match local and network-wide activity of sensor nodes that obtain dynamic energy supply from environments.

In [21], a system called eShare is described to support energy sharing among multiple-embedded sensor devices. They design energy routers for energy storage and routing devices. Energy access and network protocols are also designed. To improve sharing efficiency subject to energy leakage, an energy charging and discharging mechanism is devised.

The preliminary result of this research has previously been reported in [15] and in this paper we consolidate the research with investigation on design issues, algorithm analysis, and discussions.

3. System Model and Preliminaries

In this section, we first describe the system model and formally state the problem. Second, we define the necessary notations and make several simplifying assumptions. Third, we devise a metric that helps realize the effective guarantee of required detectability for any event. Finally, we analyze the detectability of the nonadaptive scheme in which sensors stay active blindly, and reveal the necessity of adaptive control on sensor active probability.

3.1. System Model and Problem Statement. We consider the sensors are deployed in a square field F with side length L according to a 2-dimensional Poisson process with rate n/L^2. Under this deployment, the number of sensors in any given region of area A is Poisson distributed with rate nA/L^2. The number of sensors in disjoint regions is independent. Usually, a random uniform distribution of points over a region can be approximated by a 2-dimensional Poisson process. Note that the actual number of nodes deployed in the field needs not to be n. A random uniform deployment can be approximated by a 2-dimensional Poisson deployment when the number of deployed sensors is sufficiently large.

The power consumption of a sensor node lies in three major units: *processor*, *sensing device*, and *radio transceiver*. Ideally, each unit has separate power control [10]. The duty cycle of the transceiver is subject to the control of communication protocols. Therefore, we assume it is given and concentrate on the study of duty cycling of the sensing device. The transceiver does not necessarily has the same duty cycle as the sensing device. The consequent advantage is the increased flexibility for our protocol to work with different communication protocols. It is important to note that a sensor node can actually be attached with multiple-sensing devices of different types. For simplification; however, we assume that a sensor node is equipped with a single-sensing device throughout the analysis and the protocol design. Nevertheless, the protocol can be easily extended to support the situation where a sensor node has multiple-sensing devices. Later, we call a sensor node just a sensor for short if it is not confused with the sensing device.

It should be noted that such an power model assumes that all the units can be independently controlled. In some cases, however, a sensor node may not be able to independently control the power consumption of each unit. In such cases, a sensor node has only one unit and has a single duty cycle. Our power model is general and covers such cases.

The objectives of the system design are twofold. First, users should be enabled to specify the lowest detectability (denoted by v_0) for any event in the sensing field. The system needs to ensure that the detectability of such a random event is greater than the required detectability. Second, event detection of the sensors should be energy-efficient such that the system can continue to be functional for a very long lifetime.

To accomplish these objectives, we have identified the key issues in the system design as follows.

(1) The system needs an effective way to realize the goal of providing guaranteed detectability for any possible event.

(2) The algorithm should minimize the active probability of every sensor, thus minimizing the energy consumption of the sensor.

(3) The power consumption of the sensors should be balanced such that as few blind spots as possible are introduced.

3.2. Notations and Assumptions. In the rest of this paper, we adopt the notations in Table 1 and make the following assumptions.

(i) *Binary Detection Model.* Each sensor has a detection range. An event is reliably detected by an active sensor if it resides in the range of an active sensor. More sophisticated models suggest that the detection probability is related to the distance between the sensor and the event. We assume that the detection range in our binary detection model is selected such that an event can be detected with high probability if its distance to the sensor is less than the detection range.

(ii) *Location Awareness.* Each sensor has the knowledge of its location. A good number of power-efficient algorithms have been proposed for practical localization in large-scale WSNs [22].

(iii) *High Density.* There are sufficient sensors deployed in the sensing field such that any point in the sensing field is covered by at least one sensor.

In the protocol design, we assume that the sensor network is deployed in a two-dimensional plane. However, the proposed protocol can be extended to a three-dimension space without much difficulty.

3.3. Realizing Detectability Guarantee. To realize the goal of providing guaranteed detectability for any event, we devise a simple but effective metric: *point coverage*. Its precise definition is given as follows.

Definition 1. For a point p within the sensing field, the point coverage of p, denoted by $\varsigma(p)$, is defined as the probability that p is covered by at least one sensor at any time.

The point coverage of p is dependant on the number of covering sensors and the active probabilities of these sensors.

Table 1: Notations employed in this paper.

Notation	Description
F	The sensing field
n	The sensor deployment rate
r	The detection range
$v(e)$	The detectability of physical event e
v_0	The lowest detectability requirement
$\varsigma(p)$	The point coverage of point p
ς_0	The necessary point coverage
$\omega(Q)$	The active probability of sensor Q
$\omega(Q, p)$	The needed active probability of sensor Q for point p
$t(e)$	The life of physical event e
$p(e)$	The point at which event e occurs
t_0	The minimum time detecting and processing an event
$S(p)$	The set of sensors covering point p
$U(Q)$	The set of grid points within the detection vicinity of sensor Q

Let $\omega(Q)$ denote by the probability that sensor Q is active at any time. It is apparent that a sensor has a longer lifetime if it has a lower active probability. In the sensing field, a point can be in the detection range of many sensors. Let $S(p)$ denote the set of sensors that cover point p. The point coverage of p is given by

$$\varsigma(p) = 1 - \prod_{\forall Q \in S(p)} (1 - \omega(Q)). \tag{1}$$

With the concept of point coverage, we show that Objective A is implied if we achieve Objective B.

Objective A. to guarantee that the detectability of any event is larger than the minimum requirement.

Objective B. to ensure that the point coverage of any point in the field is larger than a given value.

Let $t(e)$ denote the lifetime of event e. The detectability of this event depends on both the event life and the point coverage of the location where the event resides. Let t_0 be the minimum necessary time required for a sensor to detect and process an event. Then, the detectability of e is given by

$$v(e) = 1 - (1 - \varsigma(p(e)))^{t(e)/t_0}. \tag{2}$$

Event life $t(e)$ is usually a random variable. Given the probability density distribution of $t(e)$, denoted by $f_{t(e)}$, we can compute the expected detectability of a random event

$$E(v(e)) = \int_{t=0}^{\infty} \left(1 - (1 - \varsigma(p(e)))^{t/t_0}\right) f(t)dt. \tag{3}$$

It is apparent that the expected detectability monotonously increases with the increasing point coverage of $p(e)$. To ensure that the detectability of the event is greater than the required detectability,

$$E(v(e)) \geq v_0, \tag{4}$$

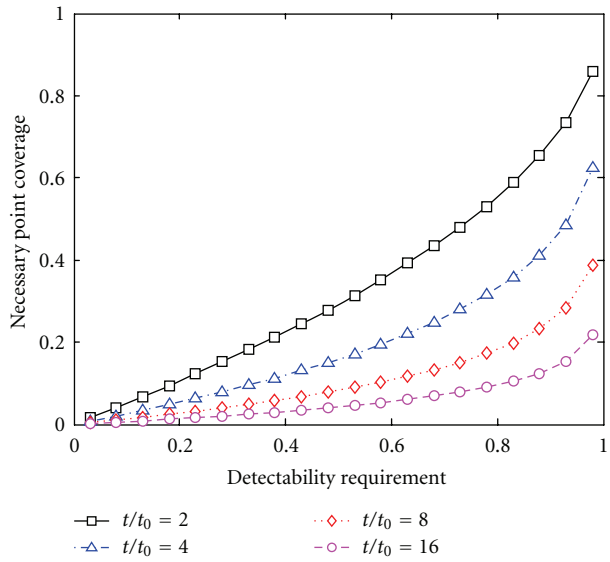


FIGURE 1: Necessary point coverage as a function of detectability requirement.

we can calculate the minimum point coverage (ς_0) such that,

$$\int_{t=0}^{\infty} (1 - (1 - \varsigma_0)^{t/t_0}) f(t) dt = v_0. \tag{5}$$

Based on the monotony of the expected detectability as a function of point coverage, we can develop a numerical procedure to obtain the desired ς_0. Let

$$\varsigma_0 = g(v_0). \tag{6}$$

It then becomes obvious that the expected detectability of e is guaranteed to be greater than the required v_0 as long as the point coverage of $p(e)$ is maintained above ς_0. Considering the arbitrary selection of the event, we conclude that by providing the guaranteed minimum point coverage of any point within the sensing field,

$$\varsigma(p) \geq \varsigma_0, \quad \forall p \in F, \tag{7}$$

the system is able to ensure that the detectability of any event is greater than the required v_0.

For pictorial study, we plot the necessary point coverage as a function of the required detectability in Figure 1. For simplification, we consider the lifetime of events as a fixed value. We can see that the necessary point coverage increases when the required detectability becomes higher. However, when the event lifetime becomes larger, the necessary point coverage can be dramatically reduced.

3.4. Detectability Analysis for Nonadaptive Scheme. There is a straightforward solution (called NAS) to provid guaranteed detectability to the sensing field; that is, guaranteeing that the point coverage of any point is greater than ς_0. According to this scheme, every sensor has the identical active probability of ς_0. When the deployment density is sufficiently large, it is obvious that this scheme can successfully provide the guaranteed detectability. However, this scheme does not scale as more sensors are deployed in the sense that additional sensor deployment does not result in extended system lifetime. We illustrate the problem by analyzing the actual detectability of any event achieved by NAS. Firstly, we study the point coverage as a function of number of deployed sensors.

Let point p be an arbitrary point in the field. Note that we do not consider the special case of points on the edge. The number of sensors covering p (denoted by N) is a random number. Since the sensors are deployed according to a 2-dimensional Poisson process, N has a Poison distribution. The probability mass function of N is given by

$$\Pr(N = k) = \frac{1}{k!}\lambda^k e^{-\lambda}, \quad \text{where } \lambda = \frac{n\pi r^2}{L^2}. \tag{8}$$

Theorem 2. *The expected point coverage of a point in the sensing field is given by*

$$E(\varsigma(p)) = 1 - e^{-\lambda v_0}. \tag{9}$$

Proof. Let point p be an arbitrary point in the field. The point coverage of p is given by

$$\varsigma(p) = 1 - (1 - \varsigma_0)^N. \tag{10}$$

The point coverage of p is actually a random variable since it relies on the number of covering sensors. We are interested in the expected $\varsigma(p)$. We condition on N to compute this expected value,

$$\begin{aligned} E(\varsigma(p)) &= \sum_{i=1}^{n} \left(\left(1 - (1 - \varsigma_0)^i\right) \times \Pr(N = i) \right) \\ &= 1 - e^{-\lambda \varsigma_0}. \end{aligned} \tag{11}$$

□

Theorem 3. *The expected detectability of any event occurring in the sensing field is given by*

$$\begin{aligned} E(v(e)) &= \int_{t=0}^{\infty} (h(i,t) \cdot \Pr(N = i)) f(t) dt, \\ \text{where } h(i,t) &= 1 - (1 - \varsigma_0)^{(i \cdot t)/t_0}. \end{aligned} \tag{12}$$

Proof. We consider an arbitrary event e occurring point p in the sensing field. Suppose the number of sensors coving p is N and the event life of e is t. According to (10) and (2), we can obtain the detectability of e,

$$v(e) = 1 - (1 - \varsigma_0)^{(N \cdot t)/t_0}. \tag{13}$$

To compute the expected detectability, we first condition on N and then consider the probability density of t. This completes the proof. □

To study the expected detectability when the system parameters vary, we plot the expected detectability as a function of the number of sensor nodes. For simplification, we

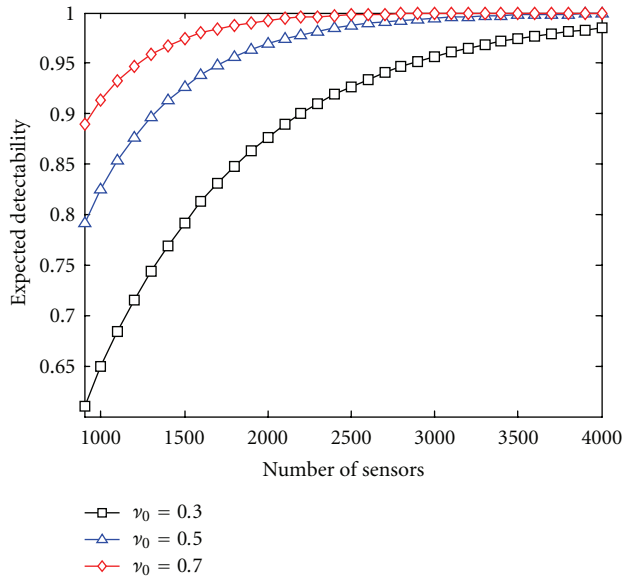


Figure 2: Expected detectability as a function of number of deployed sensors; $t(e) = t_0$

consider the lifetime of events as a fixed value equal to t_0. We set the field side to 300 m. The detection range of the sensor is 10 m. We vary the number of sensors from 4000 to 10000. In this case, the expected detectability is given by

$$E(v(e)) = 1 - e^{-\lambda\varsigma_0}. \tag{14}$$

Figure 2 shows the expected detectability as a function of the number of deployed sensors under different detectability requirements. We can see that the actual detectability is dramatically larger than the required detectability even when the density of the sensors are relatively low. With the increasing number of sensors, the detectability quickly converges to one. This suggests that NAS is not scalable to the sensor density, thus wasting precious energy.

4. Design of GAP

In this section, we first give an overview of the design of GAP. Next, we describe the detailed design. Third, we discuss some design issues. Finally, we present the analysis of the algorithm.

4.1. Overview. There are two critical design goals for GAP. On one hand, it should ensure that the point coverage of any point in the sensing field is not less than ς_0. On the other hand, it needs to reduce energy consumption of every sensor, thereby extending the system lifetime as much as possible. The algorithm adopts a probabilistic approach, where every sensor probabilistically stays active. At any time, a sensor Q is active with probability of $\omega(Q)$ and is in power save mode with probability of $1 - \omega(Q)$.

The central issue of the GAP design is the determination of the active probability of each sensor. It is intuitive that the active probability should be minimized for the purpose of higher energy efficiency. However, at the same time it should be sufficiently large to ensure that point coverage of any point is above ς_0. This poses a rigid requirement on the algorithm design. To exploit the dense deployment and balance energy consumption of the sensors, GAP adaptively tunes the active probability of every sensor such that the active probability is minimized but is adequate to ensure that the lowest point coverage within its detection vicinity is not less than ς_0.

The GAP algorithm consists of two phases. In the first phase, each sensor conservatively selects an initial active probability based on the neighborhood information. The initial probability is so sufficiently large that the point coverage of any point is larger than ς_0. To solve the energy waste introduced by the conservativeness, in the second phase, each sensor executes an iterative refinement procedure to reduce active probability for better energy efficiency. The refinement procedure terminates in finite number of steps. As there are infinitely points in the sensing field, we divide the field into virtual grids, as shown in Figure 4. We consider grid points and will later show that these grid points are sufficient in providing guaranteed detectability for any point. Figure 3 depicts the state transition diagram of the proposed algorithm.

4.2. Design Details

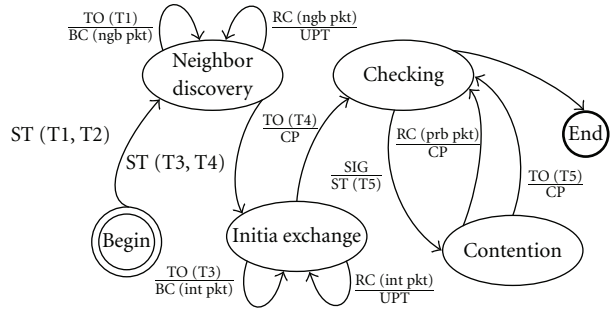
4.2.1. Neighbor Discovery. At the beginning, each sensor discovers its neighbors within $2r$ distance from itself by exchanging HELLO messages with each other. A HELLO message encloses the ID and the location of the sensor. For a given sensor, a neighbor is a *detection neighbor* (distinguished from a communication neighbor) if its distance to the neighbor is less than $2r$. Every sensor maintains a table for its detection neighbors. Upon receiving a HELLO, the sensor records the sender in the table if the sender is a detection neighbor; otherwise, this packet is silently dropped. Note that such small HELLO messages can be piggybacked through other protocol packets for energy efficiency, such as localization messages in the initialization process. The time period for neighbor discovery should be sufficiently long such that each sensor can broadcast its HELLO message.

4.2.2. Initial Probability Selection. After neighbor discovery, the sensors start to compute its initial active probability. The initial active probability guarantees that the point coverage of any point in the field is greater than ς_0.

The point coverage of p is given by

$$\varsigma(p) = 1 - \prod_{B \in S(p)} (1 - \omega(B)). \tag{15}$$

To guarantee that the point coverage is not less than ς_0, each sensor initially computes the probability needed for every single grid point within its detection vicinity, and then calculates the active probability needed at the sensor. Each sensor Q considers a grid point p and computes the active probability needed for p, denoted by $\omega(Q, p)$. With the consideration of energy balance, we let the sensors covering p



TO (Ti): Timeout of timer Ti

ST (Ti): Set timer Ti

BC (pkt): Broadcast pkt

RC (pkt): Receive pkt

UPT: Update active probability

SIG: Reduction is significant

Non-SIG: Reduction is nonsignificant

CP: Compute new active probability

Figure 3: State transition diagram of the GAP algorithm.

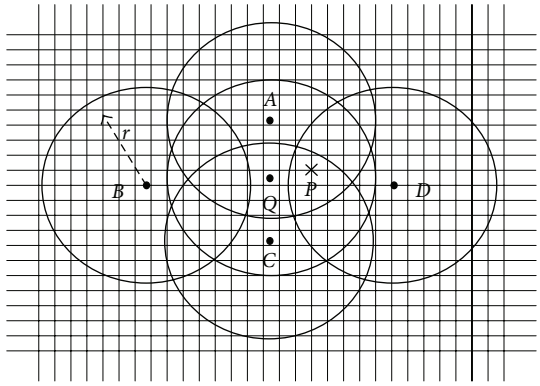


Figure 4: Grid points layout for active probability determination.

play an equally important role in detecting events at p. Thus, Q figures out the number of detection neighbors that cover p by checking the table of detection neighbors. Then Q is able to compute $\omega(Q, p)$,

$$\omega(Q, p) = 1 - \sqrt[k]{1 - \varsigma_0}, \quad \text{where } k = |S(p)| \geq 1. \tag{16}$$

To compute the initial active probability for sensor Q, it takes the maximum of the active probabilities for all grid points within its detection vicinity. Let $U(Q)$ denote the set of all the grid points within the detection range of Q. Then, the active probability of Q is

$$\omega(Q) = \max\{\omega(Q, p), \forall p \in U(Q)\}. \tag{17}$$

The selection of the initial probability is conservative in the sense that it takes the maximum value as its active probability to ensure that every point in its detection vicinity provides larger point coverage than required. The consequence is that the point coverage of a point may actually be much larger than the required one. Such conservativeness incurs additional energy consumption and therefore leads to less energy efficiency.

4.2.3. Refining Active Probabilities. To solve the problem introduced by the conservativeness of the initial selection, we propose a *coordinated probability refinement* procedure, which is a completely localized algorithm. Each sensor recalculates a new active probability based on the active probabilities of its detection neighbors. If the newly computed active probability is smaller, it tries to update its active probability, attempting to reduce its duty cycle. It is guaranteed that this refinement procedure terminates in finite number of rounds.

After determining the initial active probability, sensors exchange their active probabilities by local broadcast. Each sensor recalculates a feasible active probability based on the active probabilities of its detection neighbors. Similarly, a sensor firstly computes a new active probability for each grid point. Consider a point p. The new feasible active probability of Q for p is given by

$$\omega^{(k+1)}(Q, p) = \begin{cases} 1 - \dfrac{1 - \varsigma_0}{y}, & \text{if } 1 - \varsigma_0 < y \\ 0, & \text{otherwise,} \end{cases}$$
$$\text{where } y = \prod_{B \in S(p) - \{Q\}} (1 - \omega^{(k)}(B)), \tag{18}$$

where (k) denotes the number of generations of the associating active probability.

To compute the new active probability, Q also takes the maximum out of all the grid points within its detection vicinity,

$$\omega^{(k+1)}(Q) = \max\{\omega^{(k+1)}(Q, p), \forall\, p \in U(Q)\}. \quad (19)$$

If the new probability is smaller than the original one, it is preferable to update the probability to the new one for better energy efficiency. Otherwise, the sensor completes its refinement procedure.

If a sensor computes a smaller new probability, it cannot update its probability to the new one immediately due to the computation dependence. It is critical to avoid parallel updates. Thus, the sensor instead creates an update attempt, trying to reduce its active probability. It is required that before a sensor can actually update its active probability, it must broadcast its new probability to its detection neighbors and prevent them from updating simultaneously. An UPDATE message is used to enclose the ID and the new probability. Before an UPDATE is broadcast, the sensor undergoes a random backoff to minimize transmission collisions.

If the sensor successfully finishes the backoff process, not receiving any update from its detection neighbors, it broadcasts its UPDATE and commits the update. Next, it recomputes its new active probability. If the sensor receives an UPDATE from its detection neighbor before it finishes its backoff process, it suppresses its planned UPDATE broadcast and cancels its own update attempt. Next, it recomputes its active probability. Such a process repeats until all the sensors fail to further reduce their active probabilities.

In practice, one issue frequently arises that at the beginning the refinement procedure, many of the newly computed probabilities are close to zero. This is not desirable, because it does not facilitate balancing power consumption among the sensors. To address this issue, we pose a constraint on the maximum reduction (denoted by $th1$) by which the active probability of a sensor can be reduced each time. It is apparent that this threshold controls the tradeoff between convergence time and energy balance. A smaller threshold can produce better energy balance but need a longer time for the algorithm to converge.

We also notice that it is unwise to allow an update that actually causes a small reduction on its active probability since it not only requires communication overhead but also may prohibit other nodes from updating their probabilities. Therefore, we prefer updates that are more productive. To this end, we pose an additional constraint on the minimal reduction (denoted by $th2$) that a viable update should possess. For a node having computed a new probability, it can make an update attempt only if the resulting probability reduction is greater than the threshold. It is also obvious that this threshold controls the tradeoff between convergence time and granularity of energy balance. A larger threshold leads to a quicker convergence but produces a more coarse-grained balance of energy consumption.

4.2.4. Extension for Surveillance Differentiation. It is sometimes necessary for some area to be more carefully monitored, requiring detection differentiation for different areas. GAP supports two types of surveillance differentiation. The first type of differentiation lies in *event detectability*. For example, the detectability at a particular point q should be at least $v_0(q)$. It is not difficult to derive the required point coverage for q, denoted by $\varsigma_0(q)$. All sensors covering q should replace ς_0 with $\varsigma_0(q)$ in (16) and (18).

The second type of differentiation is in *detection degree*. Recall that previously an event is considered to be reliably detected as long as it is covered by one active sensor. It implies that the detection degree is one. In practice, however, the detection of a sensor on an event can be unreliable. To address this problem, we can require that an event must be detected by multiple sensors before it is considered to be successfully detected. This suggests a higher degree. This increases the robustness of event detection against unreliable sensing and sensor failures.

In the following, we take example that point q needs a higher detection degree of two. For distinguish, we define the resulting point coverage of q as *quadratic point coverage* (denoted by $\hat{\varsigma}(q)$). It is given by,

$$\hat{\varsigma}(q) = 1 - \prod_{Q \in S(q)} \omega(Q) - \prod_{Q \in S(q)} \left(\omega(Q) \prod_{B \in S(q) - \{Q\}} \omega(B) \right). \quad (20)$$

To obtain the initial active probability for each sensor coving q, we need to solve a high-dimensional equation. Nevertheless, the quadratic point coverage monotonously increases with increasing initial probability. Based on this, it is easy to develop a numerical procedure to find a desirable active probability that is close to the real minimum, which satisfies $\hat{\varsigma}(q) \geq \varsigma_0(q)$. It is worth noting that it is unnecessary to compute the exact minimum because the refinement procedure is only aimed to reduce active probability as much as possible. However, it is preferable to find one that is much closer to the minimum for the purpose of better energy efficiency.

For the refinement procedure, a sensor adjusts its active probability based on the active probabilities of its detection neighbors. The formula (18) should be reformulated as follows:

$$\omega^{(k+1)}(Q, q) = \begin{cases} 1 - \dfrac{1 - \varsigma_0(q) - a}{ab}, & \text{if } 1 - \varsigma_0(q) - a < ab \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{where } a = \prod_{B \in S(p) - \{Q\}} \left(1 - \omega^{(k)}(B)\right),$$

$$b = \sum_{B \in S(p) - \{Q\}} \frac{\omega^{(k)}(B)}{1 - \omega^{(k)}(B)}. \quad (21)$$

4.3. Design Issues

4.3.1. Grid Granularity. There is a concern about the granularity of grid points, characterized by grid size d. Notice

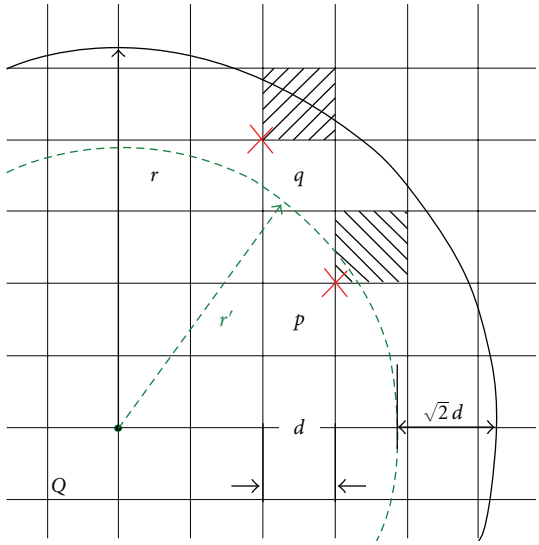


Figure 5: Grid point granularity. The dotted circle represents the nominal detection range, and the solid circle is the real detection range.

that GAP actually provides guaranteed detectability for every grid point. However, this does not necessarily imply that the detectability at any point in the field is also satisfied. To address this problem, we adopt a similar technique as in [5]. We propose a nominal detection range r' that is smaller than the real detection range. For each small grid, if a sensor covers any grid point of this grid with the nominal detection range, the sensor completely covers the whole grid with its real detection range. By this means, the system can guarantee required detectability of any point in the field if it ensures that the detectability at any grid point is greater than the required one when using the nominal detection range. It is not difficult to see that $r' \leq r - \sqrt{2}d$. As shown in Figure 5, grid point p is within the nominal detection range of Q, then the shadowed grid which p is attached to is completely covered by Q. In contrast, point q is out of the nominal detection range of Q; although it is within the real detection range of Q, the shadowed grid is not completely covered by Q.

It should be noted that such a solution comes at the expense of reduced energy efficiency. This is because each sensor loses some area that is actually within its detection vicinity. It is clear that the grid granularity controls the tradeoff between energy efficiency and computational complexity. A finer granularity can lead to better energy efficiency but causes a higher computational complexity. In our implementation, d is set to one-tenth of the detection range. Under this configuration, each sensor is expected to have $\lfloor 100\pi \rfloor$ grid points.

4.3.2. Network Dynamics. A sensor network is in nature very dynamic in the sense that existing sensors may become unavailable because of energy depletion or environmental damage, or new sensors may join the network for enhanced performance or extended lifetime. It is of great importance for the network to adapt to such dynamics.

To deal with new sensor additions, a new sensor broadcasts a PROBE message, which includes its location and ID, to inform its detection neighbors of its emergence. Upon receiving a PROBE, a sensor responds with an ECHO message that includes its ID and its location. With the received probabilities, the new sensor computes the necessary probability as specified in (16) to meet the point coverage of every point within its detection vicinity. Next, it broadcasts an UPDATE and triggers a refinement procedure, which gives its neighbors a chance to decrease their active probabilities.

To deal with sensor leave due to power depletion or environmental damage, there are two basic approaches. One is to let each sensor periodically broadcasts heartbeat beacons. By this means, a sensor is able to be aware of a neighbor's leave when it fails to receive the heartbeat beacons from that neighbor for a certain time. It can then recompute its probability to compensate the point coverage loss caused by that neighbor's leave. With periodic beacons, a WSN is responsive to sensor failure. However, periodic beacons should be used with caution since it causes much traffic overhead.

The other approach is to reschedule the whole network periodically. This approach can also deal with the dynamic addition of new sensors. Rescheduling also helps to achieve better energy balance since it gives additional chances for sensors with lower energy to decrease active probability. The key issue here is the selection of the rescheduling period. It should be adaptive to the degree of network dynamics. A more dynamic network should have a shorter rescheduling period.

4.3.3. Heterogeneous Sensors. The sensors may have different detection ranges due to various reasons. However, GAP is able to deal with such heterogeneity easily. Recall that each sensor determines the set of grid points according to its own detection range. In addition, when computing the active probability for a point, a sensor needs to know the detection vicinity of their neighbors. However, this can be easily done by exchanging such information during the phase of neighbor discovery.

4.4. Algorithm Analysis

4.4.1. Correctness

Theorem 4. *GAP is correct; that is, it is able to provide guaranteed detectability for any point in the sensing field.*

Proof. In Section 2, we have proved that the detectability of any event at a point can be ensued to be larger than the required detectability if the point coverage of each point in the field is not less than ς_0. In the selection of the initial active probability, each sensor is assigned the probability that is sufficiently larger than the required ς_0. In the refinement procedure, a sensor computes its necessary probability for each grid point in its detection range. It takes the maximum among all the grid points as its new active probability.

Parallel updates are prevented using the effective random backoff technique. An update is a local operation in the sense that it only involves the region within the detection vicinity of the updating sensor and does not affect other regions. By introducing the nominal detection range, GAP can successfully ensure that the detectability of every point is greater than the required detectability υ_0. □

4.4.2. Convergence.

Theorem 5. *GAP converges in a finite number of steps.*

Proof. The maximum probability of a sensor is one. Each successful update will reduce the active probability by at least $th2$. In GAP, no operation will cause the active probability of a sensor to increase. Thus, there is no fluctuation. Note that the minimum of the active probability is zero. This suggests that the number of updates that a sensor could have is at most $1/th2$. Thus, GAP converges in a finite number of steps. □

4.4.3. Computation Complexity. A tiny sensor processes limited computational capability. Thus, it is important that the computation complexity is affordable for such tiny sensors. Let us look at the number of steps needed for each sensor to compute the final active probability. Each sensor covers $s = \pi r^2/d^2$ grid points. Suppose a sensor has m detection neighbors. For each grid point, the sensor needs m steps to determine the set of covering sensors. In computing the initial active probability, the sensor spends constant time to compute the probability for a point. Finally, it takes s steps to compute its active probability. Thus, it needs ms steps in computing the initial probability. Thus, the total steps for computation is

$$\pi \frac{r^2}{d^2} \times m. \tag{22}$$

For instance, when $d = r/10$ and $m = 20$, it takes less than 10 thousand steps. Later, in each round of refinement, a sensor basically performs the same operations as in the initial computation. However, we emphasize that only those sensors that feasibly further reduce probability need to perform such operations.

A tiny sensor also has very small memory. For example, a typical Mica2 sensor [23] has 4 K Bytes RAM. Memory usage in GAP needs to be investigated. The memory usage is mainly for storing the probabilities computed for the grid points, which are s bytes. In addition, the sensor needs $4m$ bytes to store the related information of detection neighbors. For instance, when $d = r/10$ and $m = 20$, it takes less than 1 K bytes. By implementing GAP using TinyOS codes on a Mica2 node, we find that such computation and space cost are affordable for sensors.

4.4.4. Communication Cost. It is of importance to study the communication complexity as it reflects energy overhead introduced by GAP. We analyze the number of protocol messages. Both the neighbor discovery and the initial active probability exchange require each sensor to broadcast a message. Thus, each sensor needs two broadcast transmissions. Later, as mentioned, a sensor can have at most $x = 1/th2$ updates and therefore it can broadcast for at most x times. As a result, a sensor can has at most $2 + x$ broadcast transmissions. In implementation, we find that the $th2$ of 1/10 can provide a good tradeoff between convergence and communication cost.

Table 2: Simulation parameters.

Parameter	Value
R	30 m
r	10 m
L	300 m
n	4000
ρ_S	19 mW
ρ_P	20 mW
ρ_R	24 mW
$t(e)$	$2t_0$
υ_0	90%
ς_0	0.684
ξ	10 J
φ	0.1
$th1$	0.1
$th2$	0.01

5. Performance Evaluation

In this section we first present the evaluation methodology and then provide comparative evaluation results.

5.1. Methodology. To validate the design and to evaluate the performance of GAP, we conduct extensive simulation experiments. Simulations are conducted using a simulator developed with extra emphasis on event detection. The simulator is built on OMNet++ [24], a powerful discrete simulation system.

In simulation experiments, we study events with the fixed event life of $2t_0$. We fix the detection range and the communication range. To derive different densities, we vary the deployment rate n. The presented results are averaged over 20 independent experiments with different sensor deployments. The simulation configuration follows the setting shown in Table 2 if not stated elsewhere. In the table, ρ_S, ρ_P, and ρ_R denote the power consumption rates of the sensing device, the processor, and the transceiver, respectively. The transceiver of the transceiver φ is set to 0.1. Although the energy provided by two AA batteries can be several thousand Joules, the energy of each sensor node is initialized to 10 J to reduce lengthy simulations.

We design the following metrics to study the performance of the algorithm.

(i) *α-Lifetime of Surveillance.* It is defined as the amount of time until the instant when only α% of the sensing field can provide the guaranteed detectability.

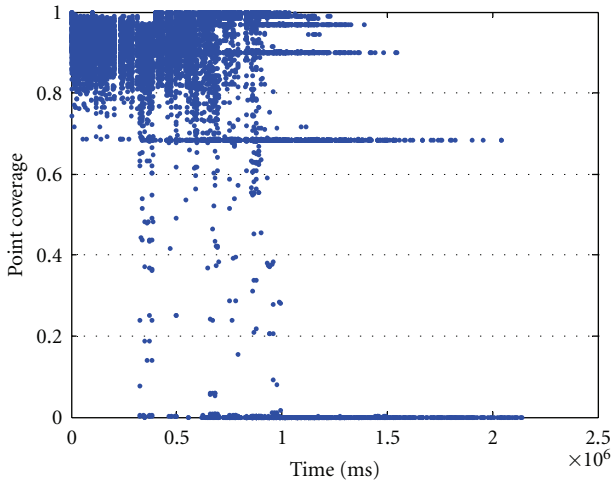


Figure 6: Point coverage over time.

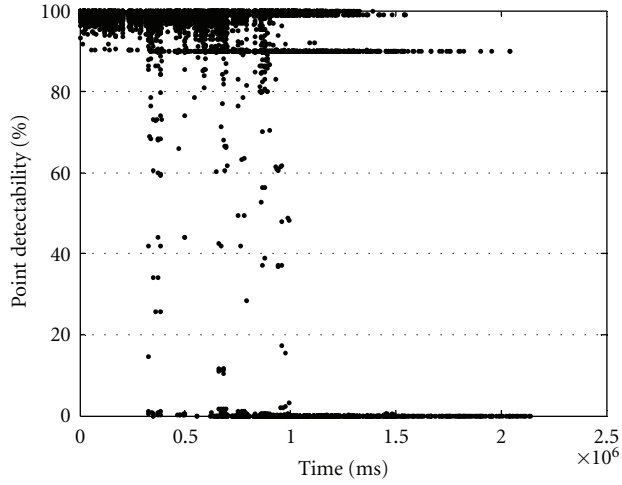


Figure 7: Point detectability over time.

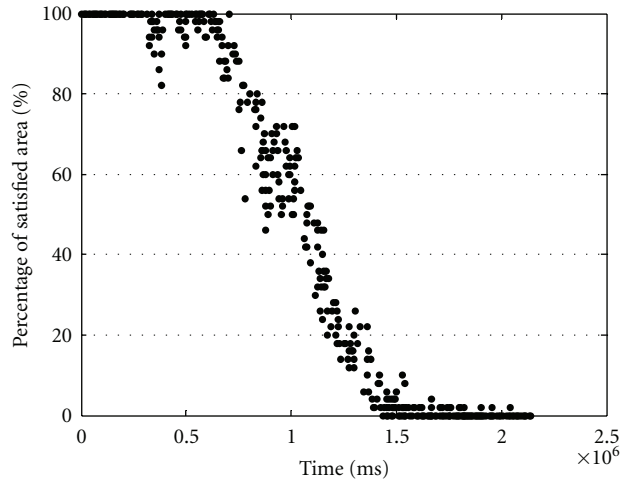


Figure 8: Percentage of satisfied area over time.

(ii) *α-Lifetime of Network.* It is defined as the amount of time until the instant when only α% of sensors are alive in the network.

(iii) *Convergence Time.* It is defined as the time from the beginning to the instant when the refinement procedure terminates.

(iv) *Number of Packets per Node.* We study the number of packets per node transmitted in the execution of the algorithm to study the communication cost.

We present a competitive study, comparing GAP with the following schemes:

(i) *NAV.* In this algorithm, every sensor has the identical active probability of ς_0.

(ii) *GNO.* It is the same algorithm as GAP except that GNO does not have the refinement procedure.

(iii) *BOUND.* It is the theoretical upper bound.

It is difficult to derive the tight bound of system lifetime. We give an optimistic upper bound of the lifetime. A point in the field is covered by λ sensors. Ideally, these sensors share the same active probability, which is $1 - (1 - \varsigma_0)^{1/\lambda}$. Thus, the actual power consumption rate of the sensing device is $(1 - (1 - \varsigma_0)^{1/\lambda})\rho_S$. The upper bound of the hard lifetime can be computed accordingly,

$$\Gamma_{\text{bound}} = \frac{\xi}{(\rho_P + \rho_R)\varphi + (\rho_S + \rho_P)(1 - \sqrt[\lambda]{1 - \varsigma_0})}. \quad (23)$$

Note, however, this upper bound is over optimistic because in reality there is no such uniform deployment where every point in the field is covered by an identical number of sensors.

5.2. Typical Run. We study a typical run in which the number of sensors is set to 4000 and the upper threshold of $th1$ is set to 0.1. The object is to investigate how the system successfully provides guaranteed detectability of any event. To this end, we generate fifty random events over the sensing field at each time instant. In Figure 6, we show point coverage over time. Each dot in the figure represents the point coverage of the location of an event. We can see that before the time of 3×10^5 ms every point coverage is beyond ς_0. After this time, the point coverages of some points drop below ς_0. This is because some spots in the field are covered only by limited sensors. After these sensors are depleted, the nearby sensors fail to provide the desired point coverage for these spots. It is very interesting that many dots are aligned on the line of $y = \varsigma_0$. This demonstrates that the GAP algorithm successfully supports the minimum necessary point coverage. Accordingly, the point detectability of corresponding points are shown in Figure 7. We can see that before the time of 3×10^5 ms, the detectability of every event is greater than v_0. Figure 8 shows the percentage of satisfied area (i.e., the point coverage of these areas are larger than ς_0) over time. Each point in the figure represents the percentage of points, out of the fifty, whose point coverage is

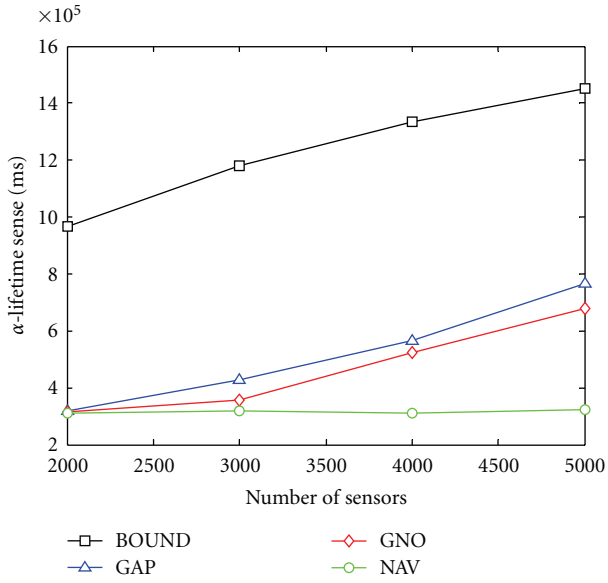


Figure 9: Comparison of 100-lifetime of surveillance.

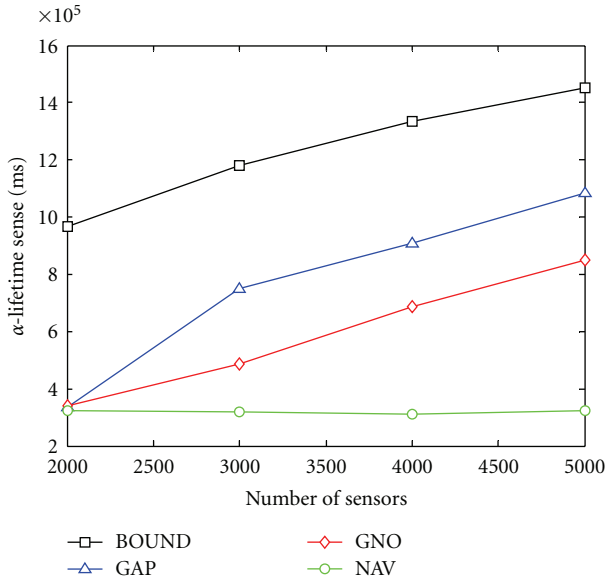


Figure 10: Comparison of 70-lifetime of surveillance.

over ς_0. We can clearly see that as time elapses, the percentage of satisfied area becomes smaller until it reaches 2×10^6 ms when all the sensors are depleted.

5.3. Lifetime Extension. We compare lifetime extensions achieved by different schemes under the different configurations of varying sensors. In Figure 9, we plot the 100-lifetime of surveillance for different schemes when the number of sensors increases. We can see that when the density of sensors is low, different schemes have similar performance in terms of 100-lifetime. This is because that most area is covered by a single sensor. When the sensor is depleted, the 100-lifetime is determined. As the number of sensors

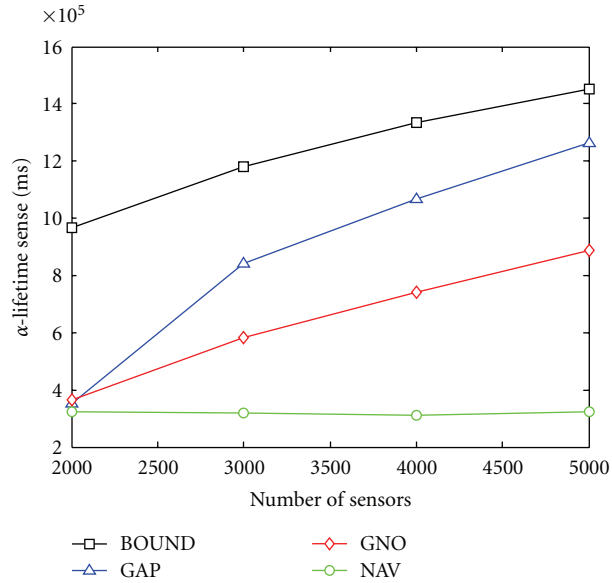


Figure 11: Comparison of 50-lifetime of surveillance.

becomes larger, the lifetime extension achieved by GAP becomes more significant. We can also find that the lifetime produced by GAP is larger than GNO, demonstrating the efficacy of the interactive refinement procedure. 100-lifetime is greatly limited by the specific deployment of the sensors. To further investigate the performance gain obtained by GAP, we show 70-lifetime and 50-lifeitme in Figures 10 and 11, respectively. From these figures, we can see the significance of the GAP algorithm. It is important to note that when the sensor density becomes higher, the lifetime extension is more significant. This suggests that GAP can scale up well with the increasing number of sensors.

We also compare lifetimes of network achieved by different schemes, as shown in Figures 12, 13, and 14. The lifetime extensions of network are reflecting the lifetime extensions of surveillance. The lifetime of NAV remains the same as the more sensors are deployed. This shows the limitation of NAV that it fails to adapt to the increasing sensor density.

6. Discussions

Unreliable Links. It has been well known that wireless transmissions are unreliable. In GAP, the broadcasting of an UPDATE is important. Suppose that sensor Q broadcasts its new probability and reduces its active probability accordingly. Sensor B, a detection neighbor of Q, fails to receive the packet from Q. In this case, there may occur a violation since B still keeps the previous probability of Q that is larger than the current real active probability of Q. Based on this out-of-date information, B may calculate a new probability that fails to ensure that the detectability of some point within its detection range.

However, we should point out that the detection range is usually much shorter than the communication range.

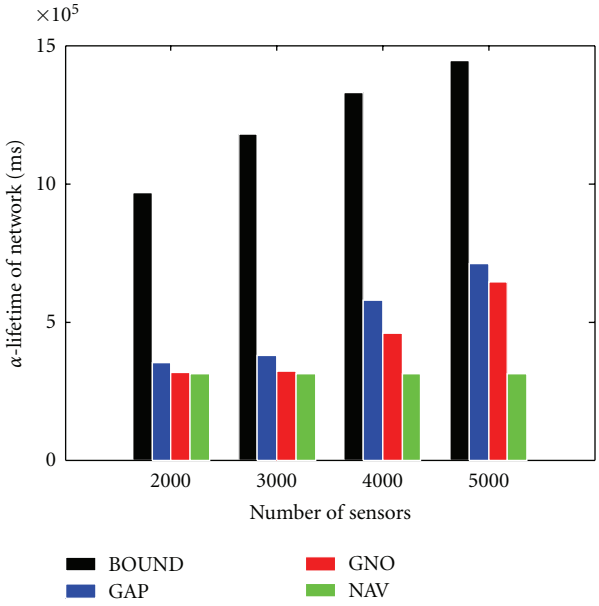


Figure 12: Comparison of 90-lifetime of network.

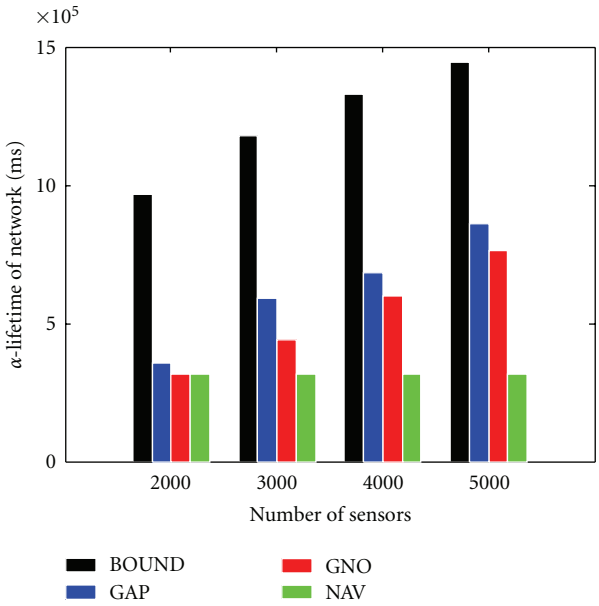


Figure 13: Comparison of 70-lifetime of network.

According to the data measured on eXtreme Scale Mote, the detection range of magnetic sensor detecting vehicles are 8 m. The communication range of a Mica2 Mote [23] is about 150 m in the outer door environment. It has been revealed that two sensor nodes that close to each other have much more reliable packet transmission. On the other hand, we can introduce an additional duplicate packet immediately following the previous one to confirm the update packet. This can further mitigate the problem that can be introduced by occasional failures of UPDATE reception, if the environment is harsh for wireless communication.

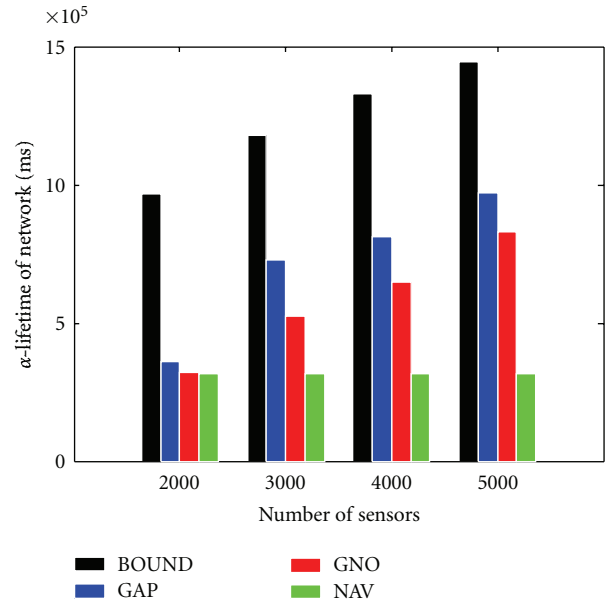


Figure 14: Comparison of 50-lifetime of network.

Inaccurate Locations. In the design of GAP, location information has been crucial. The accuracy of location information of sensor nodes certainly impacts the performance of the algorithm. If the estimated location is inaccurate, a sensor fails to precisely identify the set of grid points that are really within its detection range and the set of detection neighbors. The consequence is that the system may fail to provide guaranteed detectability.

Fortunately, we are able to address the problem introduced by inaccurate location if location errors are insignificant. On one hand, we have witnessed rapid advances in technologies for positioning sensor nodes accurately. Recently, study has reported that localization with accuracy of several centimeters has been possible [25]. On the other hand, we can use a more conservative nominal detection range to compensate the inaccuracy introduced by location errors.

7. Conclusion and Future Work

In this paper, we have presented the GAP algorithm that provides guaranteed detectability for any event occurring in the sensing field. GAP exposes a convenient interface for the user to specify the desired detectability. Employing the probabilistic approach, GAP is able to finely tune the active probability of each sensor so as to minimize the power consumption of the sensors. The algorithm does not rely on costly time synchronization and is fully distributed, therefore truly scalable to network scale and sensor density. It has demonstrated through simulation experiments that GAP significantly prolongs system lifetime while satisfying the specified detectability for any event.

The future work will proceed in several important directions. First, we plan to further study the impact of inaccurate location and unreliable wireless communications on

detection performance and the necessary design that should be enhanced. Second, we will implement the algorithm in a testbed to validate the design and to study its performance under realistic complex environments.

Acknowledgments

This research is supported by Shanghai Pu Jiang Talents Program (10PJ1405800), Shanghai Chen Guang Program (10CG11), NSFC (no. 61170238, 60903190, 61027009, 60970106, and 61170237), 973 Program (2005CB321901), MIIT of China (2009ZX03006-001-01 and 2009ZX03006-004), Doctoral Fund of Ministry of Education of China (20100073120021), 863 Program (2009AA012201 and 2011AA010500), HP IRP (CW267311), Science and Technology Comission of Shanghai Manicipality (08dz1501600), SJTU SMC Project (201120), and Program for Changjiang Scholars and Innovative Research Team in Universities of China (IRT1158, PCSIRT). In addition, it is partially supported by the Open Fund of the State Key Laboratory of Software Development Environment (Grant no. SKLSDE-2010KF-04), Beijing University of Aeronautics and Astronautics.

References

[1] S. M. Brennan, A. M. Mielke, and D. C. Torney, "Radioactive source detection by sensor networks," *IEEE Transactions on Nuclear Science*, vol. 52, no. 3, pp. 813–819, 2005.

[2] H. Ngan, Y. Zhu, L. M. Ni, and R. Xiao, "Stimulus-based adaptive sleeping for wireless sensor networks," in *Proceedings of the International Conference on Parallel Processing (ICPP '05)*, pp. 381–388, June 2005.

[3] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: a robust energy conserving protocol for long-lived sensor networks," in *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS '03)*, pp. 28–37, May 2003.

[4] D. Tian and N. D. Georganas, "A node scheduling scheme for energy conservation in large wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 3, no. 2, pp. 271–290, 2003.

[5] T. Yan, T. He, and J. A. Stankovic, "Differentiated surveillance for sensor networks," in *Proceedings of the ACM 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 51–62, November 2003.

[6] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pp. 35–45, June 2003.

[7] T. He, P. Vicaire, T. Yan et al., "Achieving long-term surveillance in VigilNet," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, April 2006.

[8] S. Xiong, J. Li, M. Li, J. Wang, and Y. Liu, "Multiple task scheduling for low-duty-cycled wireless sensor networks," in *Proceedings of the IEEE (INFOCOM '11)*, pp. 1323–1331, April 2011.

[9] Z. Li, M. Li, and Y. Liu, "Towards energy-fairness in asynchronous duty-cycling sensor networks," in *Proceedings of the the 31st Annual IEEE International Conference on Computer Communications (INFOCOM '12)*, 2012.

[10] V. Shnayder, M. Hempstead, B. R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 188–200, November 2004.

[11] Y. C. Tseng, C. S. Hsu, and T. Y. Hsieh, "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, pp. 200–209, June 2002.

[12] E. I. Shih, A. H. Shoeb, and J. V. Guttag, "Sensor selection for energy-efficient ambulatory medical monitoring," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, pp. 347–358, June 2009.

[13] M. Li, Y. Liu, and L. Chen, "Nonthreshold-based event detection for 3D environment monitoring in sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 12, pp. 1699–1711, 2008.

[14] P. Dutta, M. Grimmer, A. Arora, S. Bibykt, and D. Culler, "Design of a wireless sensor network platform for detecting rare, random, and ephemeral events," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 497–502, April 2005.

[15] Y. Zhu, Q. Chen, and L. M. Ni, "On providing guaranteed detectability for surveillance applications," in *Proceedings of the 36th International Conference on Parallel Processing (ICPP '07)*, September 2007.

[16] Q. Cao, T. Abdelzaher, T. He, and J. Stankovic, "Towards optimal sleep scheduling in sensor networks for rare-event detection," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 20–27, April 2005.

[17] T. He, S. Krishnamurthy, J. A. Stankovic et al., "Energy-efficient surveillance system using wireless sensor networks," *Proceedings of the ACM 2nd International Conference on Mobile Systems, Applications and Services (MobiSys '04)*, pp. 270–283, 2004.

[18] A. Keshavarzian, H. Lee, L. Venkatraman, K. Chitalapudi, D. Lal, and B. Srinivasan, "Wakeup scheduling in wireless sensor networks," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '06)*, pp. 322–333, May 2006.

[19] Y. Gu, T. Zhu, and T. He, "ESC: energy synchronized communication in sustainable sensor networks," in *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP '09)*, pp. 52–62, October 2009.

[20] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z. L. Zhang, "Leakage-aware energy synchronization for wireless sensor networks," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, pp. 319–332, June 2009.

[21] T. Zhu, Y. Gu, T. He, and Z. L. Zhang, "EShare: a capacitor-driven energy storage and sharing network for long-term operation," in *Proceedings of the 8th ACM International Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 239–252, November 2010.

[22] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts," in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, December 2002.

[23] XBow Company, http://www.xbow.com.
[24] OMNeT++, http://www.omnetpp.org/.
[25] A. Savvides, C. Han, and M. Srivastava, "Dynamic fine grained localization in Ad-Hoc sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom '01)*, July 2001.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 980241, 13 pages
doi:10.1155/2012/980241

Research Article

ARQ Protocols for Two-Way Wireless Relay Systems: Design and Performance Analysis

Zhenyuan Chen,[1] Qiushi Gong,[1] Chao Zhang,[2] and Guo Wei[1]

[1] *Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China*
[2] *School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China*

Correspondence should be addressed to Chao Zhang, chaozhang@mail.xjtu.edu.cn

Received 12 January 2012; Revised 25 March 2012; Accepted 6 April 2012

Academic Editor: Hongli Xu

Two-way relay (TWR) communication, a new cooperation paradigm that allows two terminals to share one relay node to communicate with each other in two phases, has played an increasingly valuable role in wireless networks to meet the stringent throughput requirement. In this paper, we focus on the designing of automatic repeat-request (ARQ) protocols for the two-way wireless relay systems. According to different feedback schedules, we propose three basic ARQ protocols to improve the throughput of two-way relay systems, namely, relay-only ARQ (Ro-ARQ), terminal only ARQ (To-ARQ) and relay-terminal ARQ (RT-ARQ). Through analyzing the outage throughput of these three ARQ protocols, it is verified that all three protocols can improve the system performance. In addition, simulation results reveal that the RT-ARQ protocol has the closest performance to the theoretical throughput upperbound among all given methods without severe deterioration on system complexity.

1. Introduction

Wireless communication has experienced tremendous progress in the past two decades. The development of relative technologies, for example, coding schemes, multiple-input, multiple-output (MIMO), and orthogonal frequency-division multiplexing (OFDM), and so forth, has contributed on accelerating the transmission rate sharply from a few kilo-bits per second (e.g., AMPS) to more than 300 Mb/s (e.g., 3GPP LTE) accompanied with the appearance of high-rate-requiring services [1]. On the other hand, however, it can also be predicted that the challenge of transmission data rate would be more serious in the near future on considering such rate-demanding applications and the limited radio resources. To cope with the insufficiency of rate caused by a variety of factors including fading, noise accumulation and interference, and so forth, the implementation of relay is introduced to assist the communication where the radio resources are not ideal, such as edge of cellular systems [2]. In this paper, two-way relay (TWR) channel, also known as physical-layer network coding (PNC) [3–6], is discussed for its improved spectral efficiency over the one-way relay or any other conventional relay strategies. The key idea of two-way relay is that two participating terminals can simultaneously transmit packets to the relay in the same phase, after which the relay processes the received signal and broadcasts it to each destination in the following phase. In other words, different from sequential data rate [7], both interacting terminals can exchange information via transmitting or receiving synchronously.

Recent works on two-way relay channels have gained great achievements on promoting its performance. These papers [3, 4] mainly demonstrated the application and designing of PNC. In [8], transmission protocols for TWR were proposed and verified of their contribution on the multiplexing as well as the diversity gain. Also, [9] designed a method of optimization on two-way relay transmission which raised the sumrate and utilize Karush Kuhn Tucker (KKT) condition to transform a nonconvex problem of power-minimization into a feasible one, reaching a tradeoff between multiplexing and diversity gain. To mitigate error

propagation, error check at relay was introduced in the work of [10], by setting a threshold at relay.

Departing from most previous works in TWR [1–10], an alternative method to improve system performance is applying the automatic repeat-request (ARQ) protocols at the data link layer to guarantee the system throughput performance [11], where cyclical redundancy check (CRC) is used for checking error packets, and retransmissions are requested if packets are received in error. The tasks of ARQ protocol designing for two-way relay channels were also conducted in previous works lately. In [12], a set of ARQ protocols were presented and analyzed but under the assumption that the bit-error rate (BER) between the relay and each individual terminal is directly assigned instead of taking the influence of transmitting power and rate into account. A unique set of ARQ protocols for TWR were also proposed and analyzed in [13] and it can be viewed as a special case of this work. Since there are two terminals and one relay in the two-way relay system, different feedback schedules can be designed to meet various transmission conditions for ARQ protocol [14]. For this reason, the ARQ protocols in our work are classified into 3 types according to where retransmissions are requested for an erroneous packet:

(i) relay-only ARQ (RO-ARQ), where retransmissions are requested at relay and the link reliability from terminals to the relay is guaranteed only,

(ii) terminal-only ARQ (TO-ARQ), where only the terminals execute repeat-request, and the end-to-end link between the terminals via relay will affect the performance, and

(iii) relay-terminal ARQ (RT-ARQ), which combines the RO-ARQ and TO-ARQ protocol together.

In [15], we just proposed above three protocols and described the details but did not analyze the performances and performed complete simulations. In this journal paper, throughput performances are analyzed. Finite-state Markov chain is applied to decompose the procedure of ARQ protocol into discrete states like [16]. Computer simulations are performed to verify the performance analysis. It can be obtained that the proposed protocols promote the throughput, and RT-ARQ protocol has the best performance.

The paper is organized as follows. A brief description of the system model of two-way relay channels is introduced in Section 2, followed by the detailed procedures of all three ARQ protocols in Section 3. The method of finite-state Markov chain analysis of the given protocols is in Section 4. After that, in Section 5, Monte-Carlo simulations of all three protocols are conducted. Finally the work is concluded in Section 6.

2. System Model and Assumptions

This work considers a wireless network where two terminals, T_1 and T_2, exchange their information through the assistance of a third node R, which acts as a relay and lies geographically between both terminals (Figure 1). Generally, there are two processing strategies at relay: amplify and forward (AF) and decode and forward (DF) [17]. The achievable throughput of the AF-based two-way relay systems has already been studied in [18]. Furthermore, the noise amplification can severely degrade the performance, especially in very low and middle signal-to-noise-ratio (SNR) environments. Thus, the DF strategy is the only consideration in this work. In addition, the DF scheme dealing with the data packet is more suitable for protocols in link layer. It is assumed that the direct link between T_1 and T_2 is not available, and all nodes work in the half-duplex mode [3, 4, 17, 18]. The transmission consists of two phases: the multiple access (MA) phase and broadcast (BC) phase. Two terminals simultaneously transmit their packets to the relay in the first phase (MA phase). Then the relay straightly decodes the received signal to perform network coding and broadcasts the encoded information in the next phase (BC phase). Each terminal is able to eliminate the interference (generated by its own packet) from the received signal and recover the information from the other terminal.

In the MA phase, the symbols of S_1 and S_2 are simultaneously transmitted to R from T_1 and T_2, respectively. Therefore, R receives the following:

$$y_R = \sqrt{P_{T_1}} h_{T_1R} S_1 + \sqrt{P_{T_2}} h_{T_2R} S_2 + n_R, \tag{1}$$

where h_{T_iR}, $i \in \{1,2\}$ is the channel coefficient between T_i and R assumed to be frequency flat and constant over the entire time slot and is characterized by Rayleigh fading, $h_{T_iR} \sim \mathrm{CN}(0,1)$. In this work, h_{T_iR} is presumed to be correctly estimated through the use of training sequences. In other words, perfect channel knowledge is available at both transceiver sides. P_{T_i} represents the average transmitting power of T_i while n_i (n_R) stands for the noise at $T_i(R)$ and is complex Gaussian random variable with $\mathrm{CN}(0, \sigma^2)$. The relay operates in the DF and adopts maximum likelihood principle to decode the received signal. To be specified, the relay will choose $\hat{S}_1$ and $\hat{S}_2$ from codebooks of each node as decoded symbols that satisfies the following:

$$\left(\hat{S}_1, \hat{S}_2\right) = \arg \min_{(\hat{S}_1, \hat{S}_2)} \left\{ \left\| y_R - \left(\sqrt{P_{T_1}} h_{T_1R} \hat{S}_1 + \sqrt{P_{T_2}} h_{T_2R} \hat{S}_2 \right) \right\|^2 \right\}. \tag{2}$$

Then the relay maps $(\hat{S}_1, \hat{S}_2)$ to S_R, $S_R = M(\hat{S}_1, \hat{S}_2)$ using the mapping principle $M(\cdot)$ like [4].

In the BC phase, the relay broadcasts S_R to T_1 and T_2. Hence, the signal received by T_i can be written as

$$y_i = \sqrt{P_R} h_{T_iR} S_R + n_i, \quad \text{for} \quad i = 1, 2, \tag{3}$$

where P_R symbolizes the average transmitting power of the relay. Additionally, at each packet, besides CRC, extra information about original owner of this packet is also included like [19], and the feedback messages from $T_i(R)$ are presumed to be received without error or delay at the $R(T_i)$.

3. Protocol Descriptions

In this paper, we aim at improving the reliable transmission in the TWR systems; thus, we propose three basic ARQ

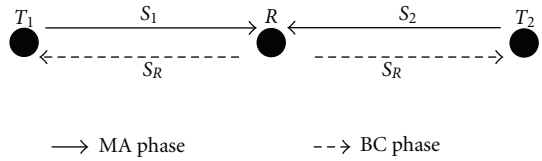


Figure 1: The two-way relay channel.

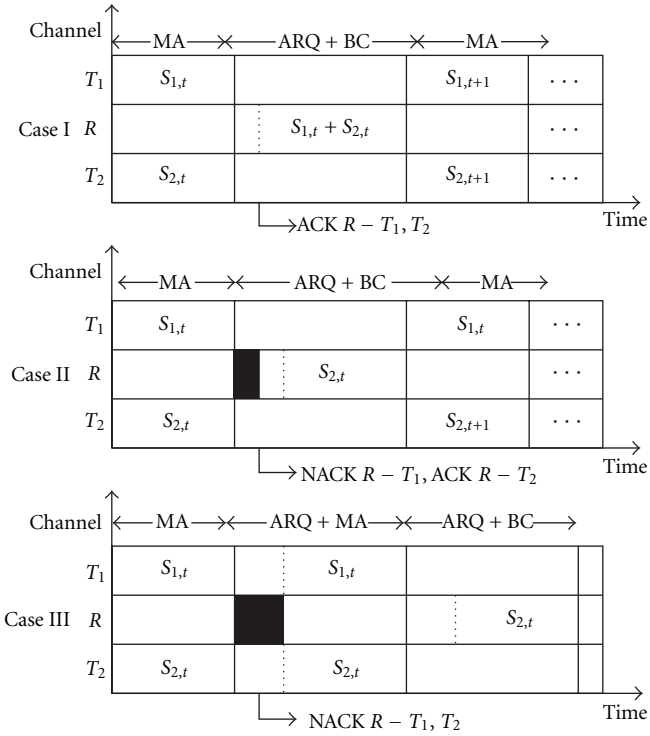


Figure 2: The RO-ARQ protocol.

protocols to fulfill this purpose: relay-only ARQ (RO-ARQ), terminal-only ARQ (TO-ARQ), and relay-terminal ARQ (RT-ARQ), which are named by where the retransmissions are requested and which link reliability is ensured. They are described in detail as follows. The analysis on their performance of throughput will be discussed in the next section.

3.1. RO-ARQ. Relay-only arq: only the relay feeds back the CRC, checking results of decoded packets $(\hat{S}_1, \hat{S}_2)$ in the MA phase, while the terminal does not feed back in the BC phase. In other words, the link reliability between T_k and R in the MA phase is guaranteed only. We classify three packet-error cases to describe the RO-ARQ protocol.

Case 1. No packets are in error at relay. The relay transmits two ACK messages, which inform T_1 and T_2 that their packets are intact in the MA phase and inserted in S_R packet header. Then T_1 and T_2 start a new round transmission in the next packet slot.

Case 2. One packet is in error at relay. If T_1's packet is in error only, the relay feeds back a NACK for T_1 and an ACK for T_2. Then retransmission will be performed by T_1 in the next packet slot while T_2 transmit its next packet in the same slot. Similarly, when only T_2's packet is in error, a reciprocity ARQ process is executed. In this paper, T_1's erroneous packet is taken as the example of Case 2 merely.

Case 3. Both the packets are in error at relay. The relay discards all the wrong packets and feeds back two NACK messages to inform the two terminals to retransmit copies of their packets. Retransmission will be started immediately on receiving the NACks. In other words, BC phase is skipped, and MA phase will be executed again. Figure 2 depicts the RO-ARQ protocol in detail.

3.2. TO-ARQ. Terminal-only ARQ: only the terminal feeds back the CRC-checking results after the BC phase. The relay just decodes and forwards in the MA phase and feedback duration; thus, the whole end-to-end link between T_1 and T_2 will have an effect on the throughput performance. Note that the retransmission requirement is made at the end of BC phase so that the procedure will not skip any phase comparing with that of RO-ARQ. The TO-ARQ protocol can also be classified into three individual packet-error cases as illustrated in Figure 3.

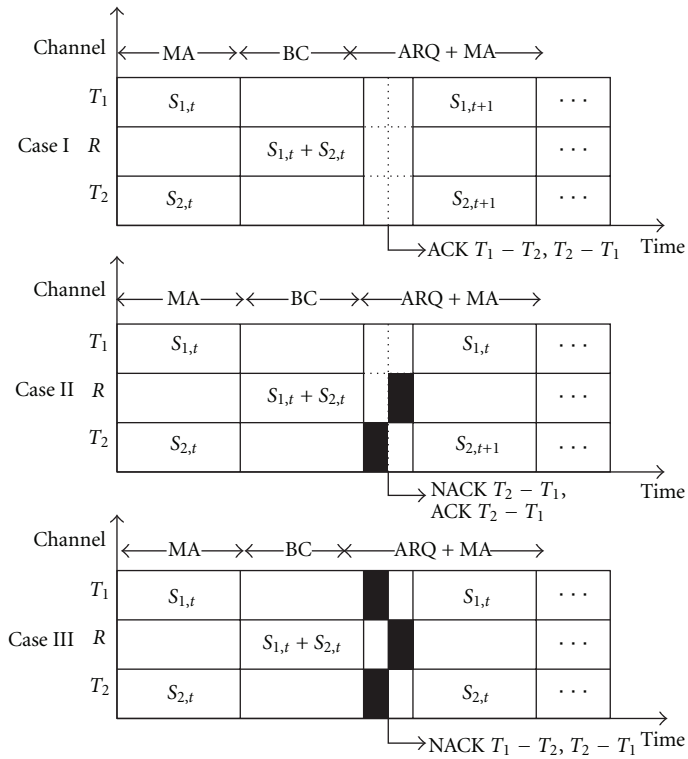


Figure 3: The TO-ARQ protocol.

Case 1. No packets are in error at terminal. Each terminal transmits an ACK message to the other one via the relay, informing that the packet was received correctly. Then the next packet slot is started.

Case 2. One packet is in error at terminal. If the packet received by T_2 is erroneous only, T_2 feeds back a NACK for T_1 and T_1 feeds back an ACK for T_2. Retransmission will be performed by T_1 in the next packet slot, during which T_2 will transmit its next packet and vice versa.

Case 3. Both the packets are in error at terminal. Each terminal will feed back a NACK message to inform the other one to retransmit the packet in next packet slot.

3.3. RT-ARQ. Relay-terminal ARQ: both the relay and terminals feed back the CRC-checking results, which combine the RO-ARQ and TO-ARQ protocol together. The relay will retransmit the packet only if packets are received at relay correctly in the MA phase yet corrupted at terminals during the BC phase. Similarly, when the relay detects error packets, NACKs will be sent to terminals and retransmission will be executed correspondingly. Note that whenever a packet fails to transmit correctly, only the related phase (i.e., MA when error at relay, BC when error at terminals) will be re-executed instead of the whole packet slot. Six packet error cases are classified to describe the RT-ARQ protocol as shown in Figure 4.

Case 1. No packets are in error at relay and terminals. The relay sends ACK messages to both terminals. The terminals send their ACK messages back in the feedback duration.

Case 2. No packets are in error at relay, while only one terminal's packet corrupted at terminal. The terminal who received the failed packet sends a NACK back in the feedback duration, and the BC phase will be executed again in which the relay retransmits the copy.

Case 3. No packets are in error at relay, while both the packets corrupted at terminal. Each terminal sends a NACK back, and the relay carries out the same operation as Case 2.

Case 4. Only one terminal's packet is in error at relay while no errors at terminal. The relay sends a NACK back for T_1 in the MA phase, and T_1 sends an ACK back in the feedback duration.

Case 5. Only one terminal's packet in error at relay, while the other's corrupted at the terminal. The relay sends a NACK back for T_1 in the MA phase, and T_1 sends a NACK back for T_2.

Case 6. Both the packets in error at relay. The relay feeds back two NACK messages to inform the terminals to retransmit their incorrect packet again in next packet slot.

4. Throughput Analysis

Data reliability in this work is more considered rather than the transmitting latency, forasmuch the relay and terminals will discard all failed copies of packets and their decoding are based only on the most recent copies, which have the highest

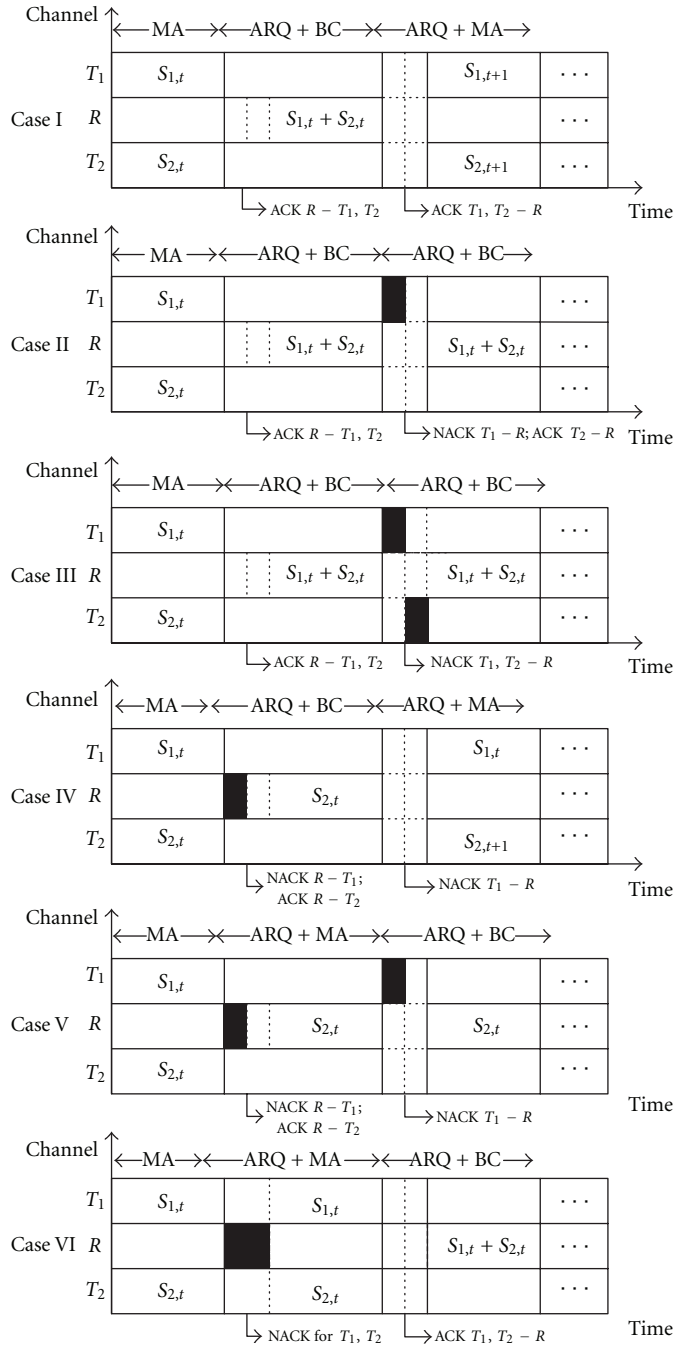


Figure 4: The RT-ARQ protocol.

probability of transmitting successfully [13]. By taking this issue into considerations, the system long-term throughput could be defined as:

$$\eta = \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} (R_1 I_1[m] + R_2 I_2[m]), \tag{4}$$

where R_i, for $i \in \{1, 2\}$, is the transmission rate (bps/Hz) of each terminal and $I_i[m]$ is an indicator function of a successful decoding event, in which a packet from node i is decoded by another terminal node in time slot m.

In this section, the procedures of all three types of ARQ protocol would be described and analyzed under the models of finite state Markov chains. Consequently, the proposed ARQ protocols satisfy appropriate assumptions of stationary and ergodicity. Thence, the long-term throughput can be rewritten as

$$\eta = R_1 \bar{I}_1 + R_2 \bar{I}_2, \tag{5}$$

where $\bar{I}_i = E[I_i[m]]$ denotes the expectation of $I_i[m]$ over fading.

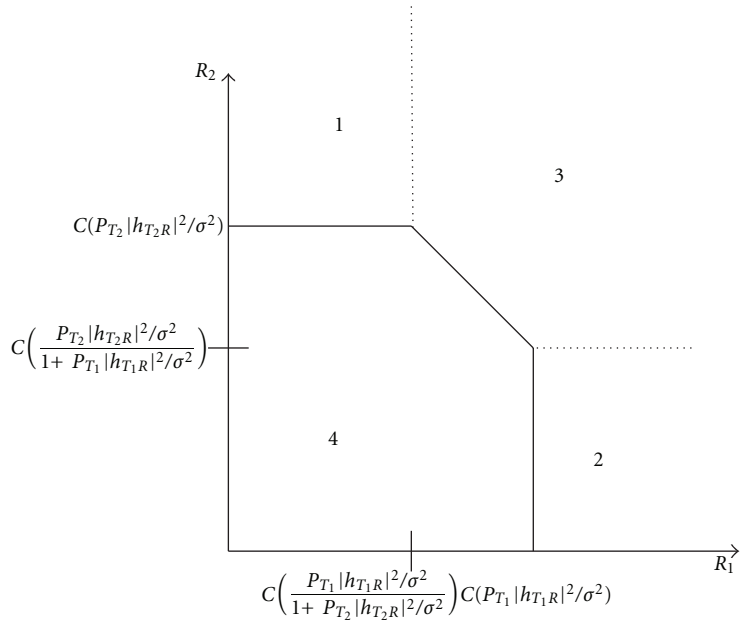


Figure 5: Achievable region conditioned on channel state for two-user MAC. Note that $C(x) = \log_2(1+x)$.

Before the analysis of proposed ARQ protocols is provided, several variables will be employed helping describe the outage probabilities of each model. In the MA phase, the outage probabilities could be depicted in Figure 5 [20].

Each region represents an individual event when two packets arrive at relay in the MA phase.

(i) Region 1: packet from T_1 arrives at relay successfully while packet from T_2 fails;

(ii) Region 2: packet from T_2 arrives at relay successfully while packet from T_1 fails;

(iii) Region 3: both packets fail;

(iv) Region 4: both packets arrive successfully.

Variables $\{P_1, P_2, P_3, P_4\}$ are defined as the probability of each corresponding region in the figure as follows:

$$\begin{aligned}
P_1 &= \frac{1}{2^{R_2}}\left[\exp\left(-\frac{2^{R_2}-1}{P_T}\right) - \exp\left(-\frac{2^{R_1+R_2}-1}{P_T}\right)\right],\\
P_2 &= \frac{1}{2^{R_1}}\left[\exp\left(-\frac{2^{R_1}-1}{P_T}\right) - \exp\left(-\frac{2^{R_1+R_2}-1}{P_T}\right)\right],\\
P_3 &= 1 - \frac{1}{2^{R_1}}\exp\left(-\frac{2^{R_1}-1}{P_T}\right) - \frac{1}{2^{R_2}}\exp\left(-\frac{2^{R_2}-1}{P_T}\right)\\
&\quad - \exp\left(-\frac{2^{R_1+R_2}-1}{P_T}\right)\\
&\quad \times\left[1 - \frac{1}{2^{R_1}} - \frac{1}{2^{R_2}} + \frac{(2^{R_1}-1)(2^{R_2}-1)}{P_T}\right],\\
P_4 &= 1 - P_1 - P_2 - P_3,
\end{aligned} \tag{6}$$

where P_T is the value of transmitting power of both terminals since in the current work they are assumed equal (i.e., $P_{T_1} = P_{T_2} = P_T$). Note that the value of σ is normalized in this work, the effect of SNR at each node is therefore directly reflected by their transmit power (i.e., $P_T/\sigma^2 = P_T$).

In the BC phase, the link between the relay and two terminals can be viewed as peer-to-peer links [13], and the outage probabilities on each link are defined as P_{out,RT_1} and P_{out,RT_2} and in the current work as follows:

$$P_{\text{out},RT_1} = P_{\text{out},RT_2} = 1 - \exp\left(-\frac{2^{R_R}-1}{P_R}\right), \tag{7}$$

where R_R and P_R symbolize the transmitting rate and power of the relay node, respectively. Similarly, SNR at relay is represented by its transmit power (i.e., $P_R/\sigma^2 = P_R$).

In order to represent the expressions in the rest of the paper less complicated, the complements of P_{out,RT_1} and P_{out,RT_2} are introduced as follows:

$$\begin{aligned}
P_{r1} &= 1 - P_{\text{out},RT_1},\\
P_{r2} &= 1 - P_{\text{out},RT_2}.
\end{aligned} \tag{8}$$

4.1. Upper Bound. The upper bound of the transmission can be obtained by assuming that two terminals can transmit without interfering each other. Thus, the maximum throughput of any terminal (e.g., T_1) can be calculated as

$$\widetilde{\eta}_1 = R_1\frac{(1 - P_{\text{out},T_1R})(1 - P_{\text{out},RT_2})}{(2 - P_{\text{out},T_1R} - P_{\text{out},RT_2})}, \tag{9}$$

in which P_{out,T_1R} (P_{out,T_2R}), similar to P_{out,RT_1} (P_{out,RT_2}), symbolizes the outage probability of the peer-to-peer link between $T_1(T_2)$ and the relay.

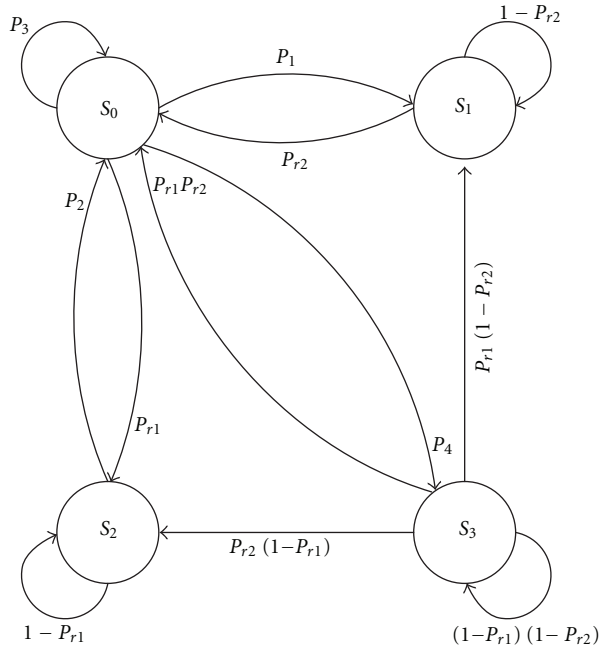


Figure 6: The state-transition diagram of RO-ARQ protocol.

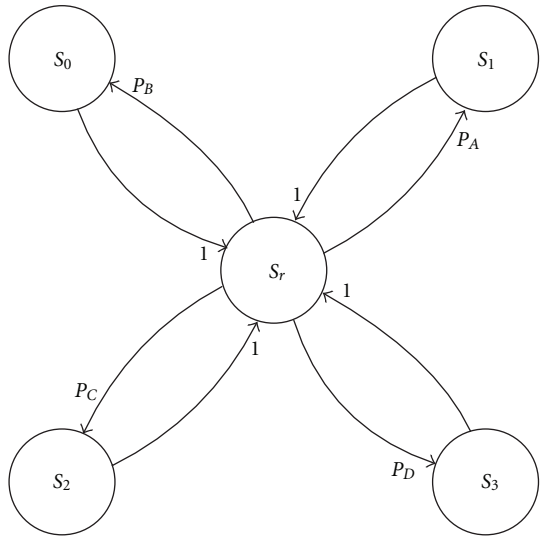


Figure 7: The state-transition diagram of TO-ARQ protocol.

Analogously, the maximum throughput $\tilde{\eta}_2$ can be given for T_2. Consequently, the upper bound for throughput can be obtained as $\eta \leq \eta_{\mathrm{UB}} = \tilde{\eta}_1 + \tilde{\eta}_2$.

4.2. RO-ARQ. Under the protocol of RO-ARQ, the repeat request is only made at the relay node; therefore the model can be studied as a Markov chain with states of relay's buffer.

(i) S_0: no packets are cached in the relay's buffer and the relay is expecting the next transmission;

(ii) S_1: packet from T_1 successfully arrives at relay while packet from T_2 fails, Corresponding to Case 2 of RO-ARQ in the previous section;

(iii) S_2: reciprocity of state S_1 substituting T_1 with T_2 and vice versa;

(iv) S_3: packets from both terminals arrive at relay with no error, corresponding to Case 1 of RO-ARQ in the previous section.

The states above can be depicted in Figure 6. As can be seen from the diagram, a successful transmission from T_i to T_j $(i, j \in \{1, 2\}, i \neq j)$ is determined when the system is at state S_3, S_2, or S_1, and packets are sent forward with probability P_{r1} or P_{r2}. Consequently, the indicator variable $\bar{I}_i$ can be written as

$$\bar{I}_i = \left(P_{si}^{\mathrm{RO}} + P_{s3}^{\mathrm{RO}}\right) P_{RT_j}, \quad i, j \in \{1, 2\},\ i \neq j, \tag{10}$$

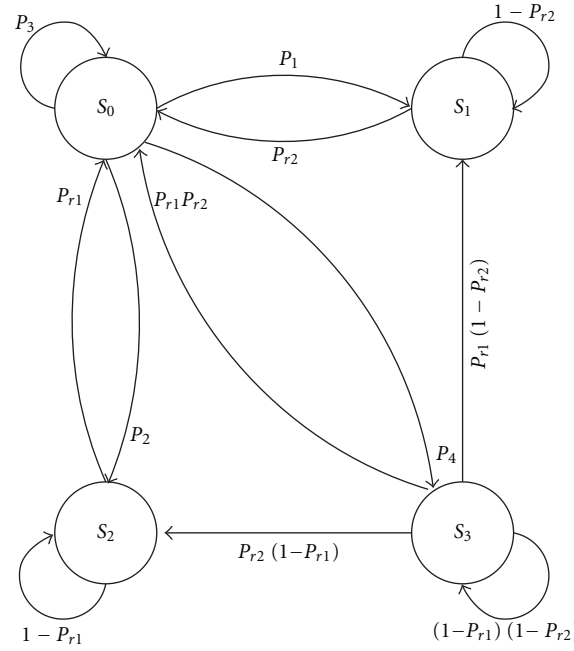


Figure 8: The state-transition diagram of RT-ARQ protocol.

where P_{RT_j} is defined as the probability of a successful packet transmission from R to T_j, and P_{si}^{RO} stands for the steady probability of relay being at state S_i within RO-ARQ protocol. By solving the state transition equations listed below:

$$\begin{aligned} P_{s0}^{\mathrm{RO}} P_1 &= P_{s1}^{\mathrm{RO}}, \\ P_{s1}^{\mathrm{RO}} P_2 &= P_{s2}^{\mathrm{RO}}, \\ P_{s2}^{\mathrm{RO}} P_4 &= P_{s3}^{\mathrm{RO}}, \\ \sum_{i=0}^{3} P_{si}^{\mathrm{RO}} &= 1, \end{aligned} \tag{11}$$

the steady state probability is:

$$\begin{aligned} P_{s0}^{\mathrm{RO}} &= (1 + P_1 + P_2 + P_4)^{-1}, \\ P_{s1}^{\mathrm{RO}} &= P_1(1 + P_1 + P_2 + P_4)^{-1}, \\ P_{s2}^{\mathrm{RO}} &= P_2(1 + P_1 + P_2 + P_4)^{-1}, \\ P_{s3}^{\mathrm{RO}} &= P_4(1 + P_1 + P_2 + P_4)^{-1}. \end{aligned} \tag{12}$$

The steady distribution of the Markov chain is acquired and thereby the throughput can be obtained as follows:

$$\eta_{\mathrm{RO}} = R_1\left[P_{r2}\left(P_{s3}^{\mathrm{RO}} + P_{s2}^{\mathrm{RO}}\right)\right] + R_2\left[P_{r1}\left(P_{s3}^{\mathrm{RO}} + P_{s1}^{\mathrm{RO}}\right)\right]. \tag{13}$$

4.3. TO-ARQ. In TO-ARQ model, the signals received at relay do not reveal whether the transmission is successful or not. Due to this reason, the analysis of TO-ARQ model adopts the combination of the relay's buffer together with terminals' rather than the relay's alone as the state variable. Under such circumstance, the state variable also has five possibilities:

(i) S_0: none of packets from both terminals transmitted correctly, and NACKs are sent to both terminals, corresponding to Case 3 of TO-ARQ in the previous section;

(ii) S_1: packet from T_1 successfully arrives at T_2 while packet from T_2 fails. Correspond to Case 2 of TO-ARQ in the previous section;

(iii) S_2: reciprocity of state S_1 substituting T_1 with T_2 and vice versa;

(iv) S_3: packets from both terminals arrive with no error, corresponding to Case 1 of TO-ARQ in the previous section;

(v) S_r: packets arrive at the relay and will be sent to both terminals in the next phase.

The states above can be depicted in Figure 7. Here the state-transition probabilities are coded for the convenience of representation as follows:

$$\begin{aligned} P_A &= P_1P_{r2} + P_4P_{r2}(1 - P_{r1}), \\ P_B &= P_3 + P_1(1 - P_{r2}) + P_2(1 - P_{r1}) + P_4(1 - P_{r1})(1 - P_{r2}), \\ P_C &= P_2P_{r1} + P_4P_{r1}(1 - P_{r2}), \\ P_D &= P_4P_{r2}P_{r1}, \end{aligned} \tag{14}$$

and easily the sum of all four probabilities is solved as follows: $P_A + P_B + P_C + P_D = 1$.

The state-transition equations of this Markov chain are the following:

$$\begin{aligned} P_{sr}^{\mathrm{TO}} P_B &= P_{s0}^{\mathrm{TO}}, \\ P_{sr}^{\mathrm{TO}} P_A &= P_{s1}^{\mathrm{TO}}, \\ P_{sr}^{\mathrm{TO}} P_C &= P_{s2}^{\mathrm{TO}}, \\ P_{sr}^{\mathrm{TO}} P_D &= P_{s3}^{\mathrm{TO}}, \\ P_{sr}^{\mathrm{TO}} + \sum_{i=0}^{3} P_{si}^{\mathrm{TO}} &= 1, \end{aligned} \tag{15}$$

where P_{si}^{TO}, $i \in \{0, 1, 2, 3, r\}$ is defined as the steady probability of each state S_i within TO-ARQ protocol. The steady-state probabilities to each state in the diagram can be solved as follows:

$$\begin{aligned} P_{sr}^{\mathrm{TO}} &= 0.5, \\ P_{s0}^{\mathrm{TO}} &= 0.5 P_B, \\ P_{s1}^{\mathrm{TO}} &= 0.5 P_A, \\ P_{s2}^{\mathrm{TO}} &= 0.5 P_C, \\ P_{s3}^{\mathrm{TO}} &= 0.5 P_D. \end{aligned} \tag{16}$$

According to the description of the protocol given above, the correct transmission of a packet from terminal i occurs only when the system is at state S_i or S_3. Therefore the throughput of TO-ARQ model is calculated as follows:

$$\eta_{\mathrm{TO}} = 0.5 R_1 \left(P_{s1}^{\mathrm{TO}} + P_{s3}^{\mathrm{TO}} \right) + 0.5 R_2 \left(P_{s2}^{\mathrm{TO}} + P_{s3}^{\mathrm{TO}} \right). \tag{17}$$

4.4. RT-ARQ. In RT-ARQ model, the relay shares the same functions as in RO-ARQ while it executes the retransmission requested by terminals. Hence, the state variable can be similar with that of RO-ARQ yet it is not the representation of the relay's buffer alone, but also, the operations the relay going to take in the next phase. Therefore, the diagram of the state transition needs modification as well to suit the current protocol, which is shown in Figure 8. The definitions of RT-ARQ's state S_0, S_1, S_2, and S_3 are given by

(i) S_0: no packets are stored in the relay's buffer, and the relay is expecting the next transmission;

(ii) S_1: the packet from T_1 is cached in the relay's buffer and will be transmitted to T_2 in the next phase. The transmission of packet from T_2 is ignored due to a failed transition $S_0 \rightarrow S_1$ or a accomplished one $S_3 \rightarrow S_1$;

(iii) S_2: reciprocity of state S_1 substituting T_1 with T_2 and vice versa;

(iv) S_3: packets from both terminals arrive at relay with no error. State may shift to S_0, S_1, or S_2 depending on whether the corresponding transmission of packet is successful or not. It is specified in Figure 8.

The state-transition equations of this Markov chain are the following:

$$\begin{aligned} P_{s1}^{\mathrm{RT}}(1 - P_{r2}) + P_{s0}^{\mathrm{RT}} P_1 + P_{s3}^{\mathrm{RT}} P_{r1}(1 - P_{r2}) &= P_{s1}^{\mathrm{RT}}, \\ P_{s2}^{\mathrm{RT}}(1 - P_{r1}) + P_{s0}^{\mathrm{RT}} P_2 + P_{s3}^{\mathrm{RT}} P_{r2}(1 - P_{r1}) &= P_{s2}^{\mathrm{RT}}, \\ P_{s3}^{\mathrm{RT}}(1 - P_{r1})(1 - P_{r2}) + P_{s0}^{\mathrm{RT}} P_4 &= P_{s3}^{\mathrm{RT}}, \\ \sum_{i=0}^{3} P_{si}^{\mathrm{RT}} &= 1, \end{aligned} \tag{18}$$

where P_{si}^{RT}, $i \in \{0, 1, 2, 3\}$ represents the steady probability of each state S_i within RT-ARQ protocol. The steady probabilities to each state in the diagram can be solved as follows:

$$\begin{aligned} P_{s0}^{\mathrm{RT}} = {} & \left(\frac{P_1}{P_{r2}} + \frac{P_{r1}}{P_{r2}} \frac{(1 - P_{r2}) P_4}{P_{r1} + P_{r2} - P_{r1} P_{r2}} + \frac{P_2}{P_{r1}} + \frac{P_{r2}}{P_{r1}} \right. \\ & \left. \times \frac{(1 - P_{r1}) P_4}{P_{r1} + P_{r2} - P_{r1} P_{r2}} + \frac{P_4}{P_{r1} + P_{r2} - P_{r1} P_{r2}} + 1 \right)^{-1}, \\ P_{s1}^{\mathrm{RT}} = {} & \left(\frac{P_1}{P_{r2}} + \frac{P_{r1}}{P_{r2}} \frac{(1 - P_{r2}) P_4}{P_{r1} + P_{r2} - P_{r1} P_{r2}} \right) P_{s0}^{\mathrm{RT}}, \\ P_{s2}^{\mathrm{RT}} = {} & \left(\frac{P_2}{P_{r1}} + \frac{P_{r2}}{P_{r1}} \frac{(1 - P_{r1}) P_4}{P_{r1} + P_{r2} - P_{r1} P_{r2}} \right) P_{s0}^{\mathrm{RT}}, \\ P_{s3}^{\mathrm{RT}} = {} & \frac{P_4}{P_{r1} + P_{r2} - P_{r1} P_{r2}} P_{s0}^{\mathrm{RT}}. \end{aligned} \tag{19}$$

A successful transmission from T_i to T_j $(i, j \in \{1, 2\}, i \neq j)$ is determined when the following state transitions take place:

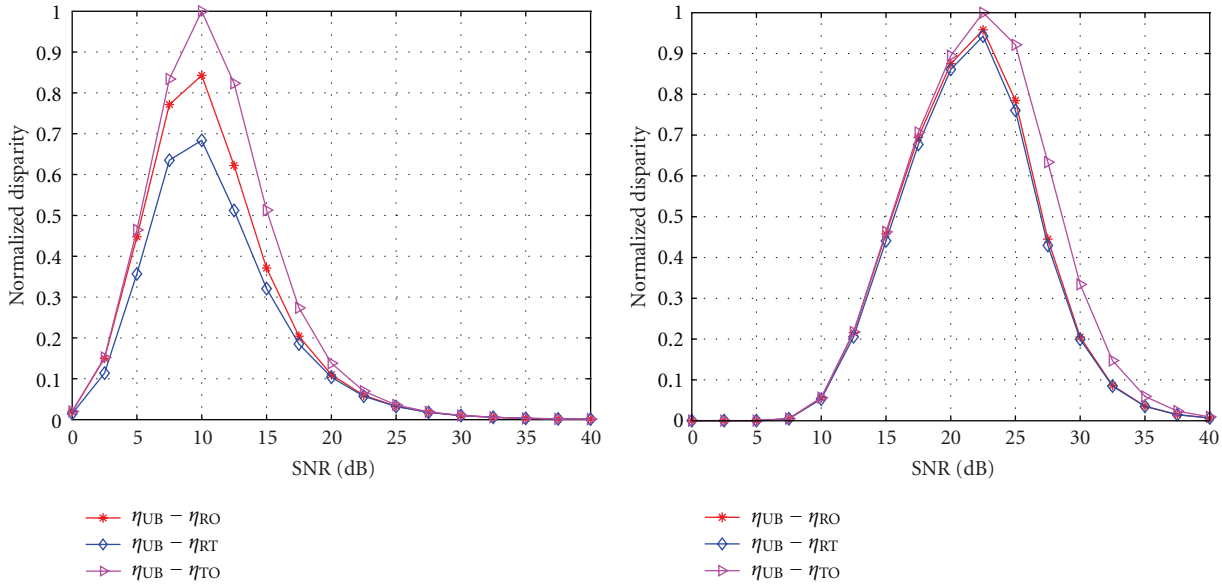
$$\begin{aligned} S_3 &\longrightarrow S_0, S_1, \quad \text{or} \quad S_2, \\ S_2 &\longrightarrow S_0, \\ S_1 &\longrightarrow S_0, \end{aligned} \tag{20}$$

thence the throughput of RT-ARQ model is given by:

$$\eta_{\mathrm{RT}} = R_1 \left[P_{r2} \left(P_{s3}^{\mathrm{RT}} + P_{s2}^{\mathrm{RT}} \right) \right] + R_2 \left[P_{r1} \left(P_{s3}^{\mathrm{RT}} + P_{s1}^{\mathrm{RT}} \right) \right]. \tag{21}$$

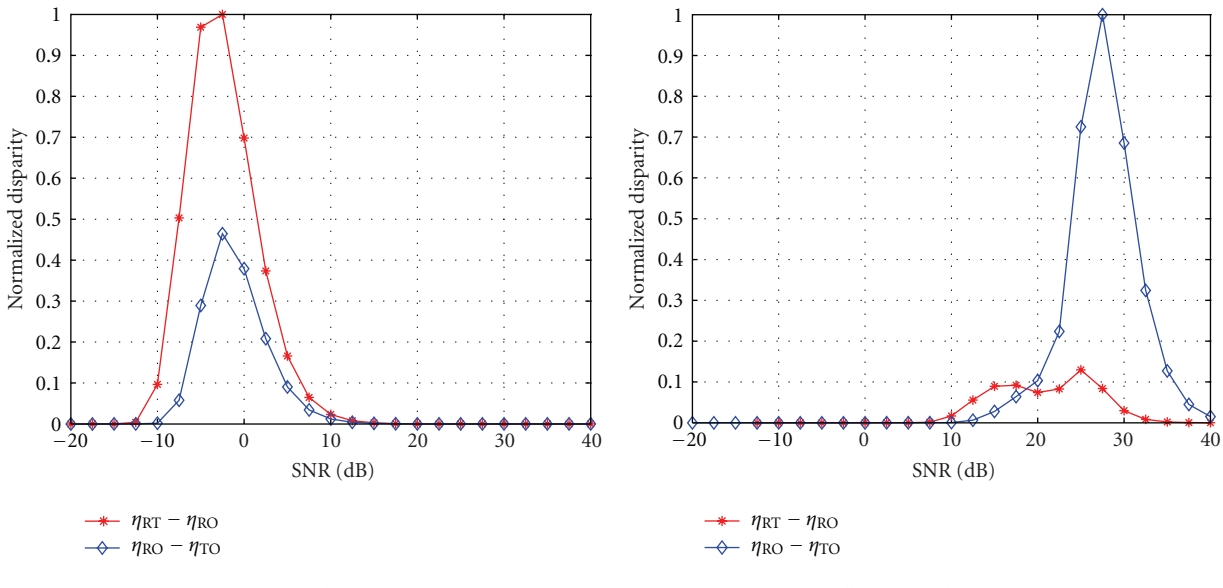
4.5. Throughput Comparisons of Different ARQ. By calculating the difference between each of $\{\eta_{\mathrm{RT}}, \eta_{\mathrm{RO}}, \eta_{\mathrm{TO}}, \eta_{\mathrm{UB}}\}$, their relationship can thus be obtained as $\eta_{\mathrm{UB}} \geq \eta_{\mathrm{RT}} \geq \eta_{\mathrm{RO}} \geq \eta_{\mathrm{TO}}$ which is depicted in Figure 9. The acceleration (deceleration) of transmission will shrink (broaden) all gaps, yet leaving the sequence of quantity unchanged.

Thereby the mutual gap between each protocol's throughput performances can be predicted from each of their differences with the upper bound. The gap between the RT-ARQ and TO-ARQ is omitted because the combination of RT-RO gap and RO-TO gap can indirectly reflect the RT-TO gap which is shown in Figure 10. When the rate



(a) $R = 2.5$ bps/Hz (b) $R = 5.0$ bps/Hz

FIGURE 9: Throughput difference between the upper bound and each of the proposed ARQ protocols under the transmitting rate of $R = 2.5$ and 5.0 bps/Hz, respectively.



(a) $R = 0.5$ bps/Hz (b) $R = 0.5$ bps/Hz

FIGURE 10: Throughput difference of RT-RO and RO-TO, under the transmitting rate of $R = 0.5$ and 5.0 bps/Hz, respectively.

is relatively low, the performance of RO-ARQ is quite close to that of the TO-ARQ's. And when it rises, on the contrary, RO-ARQ's throughput performance will approach RT-ARQ's. Therefore, it will be much saving to choose RO-ARQ over RT-ARQ under high transmitting rate since they perform similarly while reducing the number of execution of ARQ by half. However, RT-ARQ is more preferable when the rate is low for the throughput performance can be guaranteed.

5. Numerical Results

In this section, computer simulation results are presented to reveal the end-to-end throughput performance of proposed ARQ protocols. For the sake of comparison, the evaluation of the transmission's upper bound is also taken into the simulation. The simulation focused on symmetric case in which $R_1 = R_2 = R_R$ and $P_T = P_R$ and is executed on four different rates, namely, $R = 0.5, 2.5, 3.5, 5$ bps/Hz

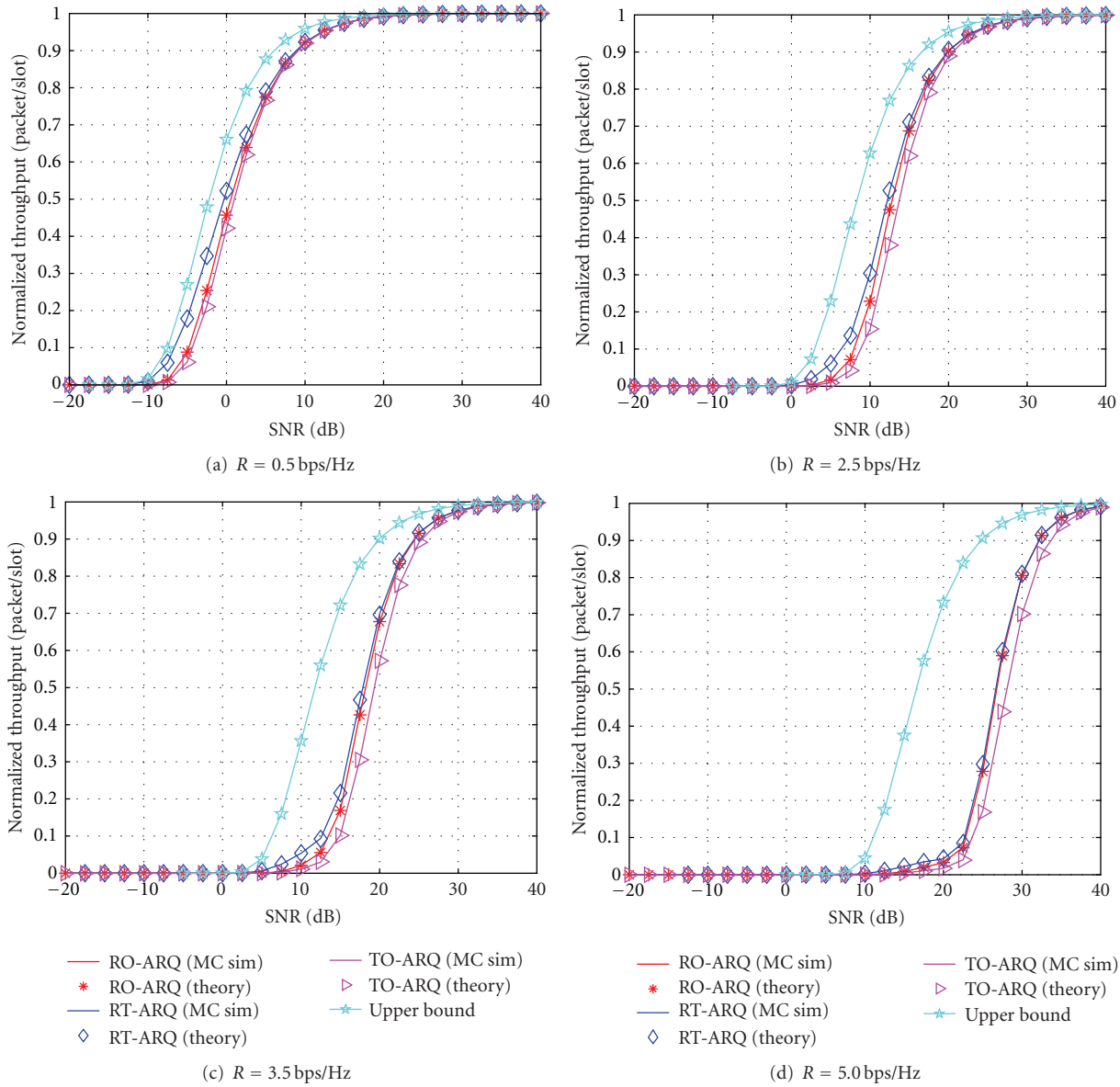
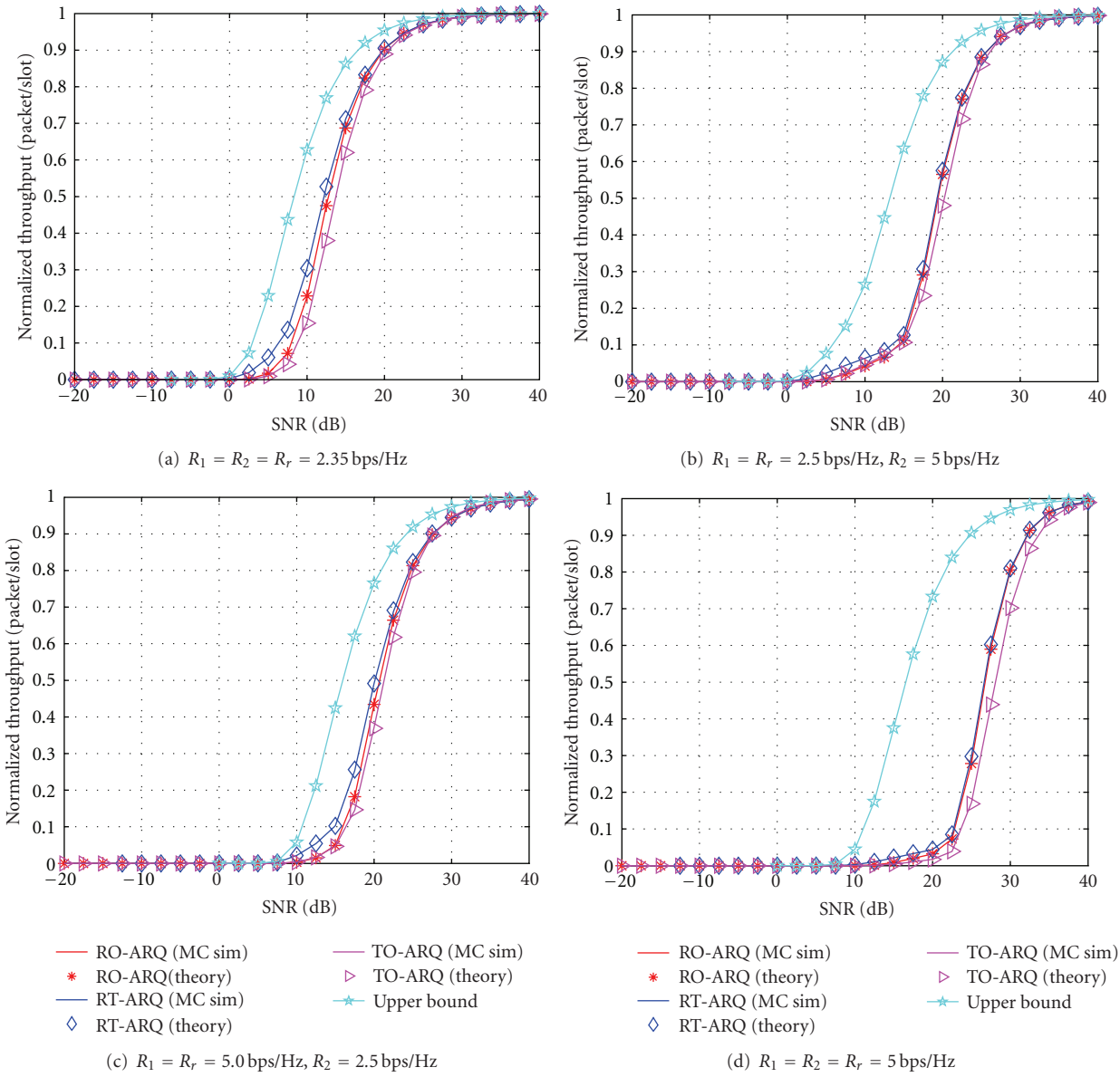


FIGURE 11: Normalized throughput η/R versus SNR for a symmetric TWR with transmission rate $R = 0.5, 2.5, 3.5, 5$ bps/Hz, respectively.

[13]. The range of SNR is from -20 dB to 40 dB [8, 17]; thus, the whole trend of curves can be observed while, on the other hand, both links ($T_1 \leftrightarrow R, T_2 \leftrightarrow R$) share the same SNR. The results are shown in Figure 11. Additionally, same scale of throughput can be convenient for observation and comparison; hence, all figures of computer simulations have been normalized. The Monte-Carlo simulation is implemented in this work to verify the consistency of the algorithm applied in the previous section with the actual scenarios, and, judging from the observation of Figure 11, both outcomes can be perfectly matched.

As can be seen, for any given transmission rate, RT-ARQ protocol has the best throughput performance among all the proposed schemes, and RO-ARQ has better throughput efficiency than TO-ARQ under any circumstances. This is due to the reason that RT-ARQ has the most flexible slot procedure. To be specified, whenever an erroneous transmission occurs, the retransmission requirement can be sent immediately in the next phase under RT-ARQ protocol: consequently, the MA or BC phase can be reexecuted and need not have to wait for any idle phase. In other words, the retransmission of RT-ARQ takes half the period of a slot on average. RO-ARQ protocol has the ability to reexecute the MA phase when packets arrive at relay unsuccessfully; yet, links of the second hop are not guaranteed forasmuch the successful transmission of a packet will require the retransmission to take more than one phase while less than a whole slot. As for TO-ARQ protocol, the retransmission takes a whole slot to carry out in the long run, and the MA phase will be seen as an idle phase when mistakes appear. For



(a) $R_1 = R_2 = R_r = 2.35$ bps/Hz

(b) $R_1 = R_r = 2.5$ bps/Hz, $R_2 = 5$ bps/Hz

(c) $R_1 = R_r = 5.0$ bps/Hz, $R_2 = 2.5$ bps/Hz

(d) $R_1 = R_2 = R_r = 5$ bps/Hz

Figure 12: Normalized throughput η/R versus SNR for a symmetric TWR with different transmission rate on each terminal.

example, if a packet arrives at relay with mistake, it will take RT-ARQ and RO-ARQ protocol one phase to accomplish the retransmission comparing that TO-ARQ will take a full slot.

When SNR is at low region, all protocols' curves including the upper bound are near zero and when SNR transcends a threshold value, throughput values rise dramatically and approach one. This is attributed to the fact that when SNR is abysmal, high-outage probability will stuck all the protocol at retransmitting states, causing decode-recode at relay unreliable. On the contrary, TWR system runs between the state of ready-to-send and the state of receiving successfully when SNR is very high. The rising range, observed from the figures, is approximately 20 dB, and the threshold floats with the transmission rate. However, comparing with the upper bound, the threshold of proposed protocols is rather more sensitive to the rate; thus, their curves move toward right side faster than the upper bound's curve as the rate increases.

Another set of curves are presented to demonstrate the performance of unequal transmission rate (i.e., $R_1 \neq R_2$). The trend of curves, as can be seen from Figure 12, follows that of equal-rate condition. Curves of three proposed protocols cluster and depart from the upper-bound as any of the rate increases. As for the influence of the relay, the ascension of the relay's transmitting rate also push all curves toward right judging from the observation of Figures 12(b) and 12(c). This can be concluded from the peer-to-peer outage probability (7) which is an increasing function of R_R and directly affects the evaluation of system's throughput. Yet RT-ARQ still outperforms RO-ARQ and TO-ARQ under all circumstances executed in the simulation.

6. Conclusion

In this paper, three ARQ protocols are investigated which designed for two-way relay systems with physical-layer network coding according to different feedback schedules at the relay and terminals. Work mainly focuses on the link reliability improvement in terms of end-to-end throughput of TWR system over slow fading wireless channels. Through performance evaluations, we confirmed that the proposed protocols can offer a smoother increase of the throughput curve, and it can significantly improve the end-to-end throughput performance in two-way relay systems. It can be observed that the RT-ARQ protocol has a better performance than the other protocols and can best approach the upper bound under low transmission rate among all proposed schemes.

Acknowledgments

The work of C. Zhang is supported by the Fundamental Research Funds for the Central Universities, by the National Natural Science Foundation of China (No. 61102082), by the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University (no. 2011D14) and National Hi-Tech Research Development Program, "863 Program," (no. 2011AA01A105). This work is supported in part by Natural Science Foundation of Education Department of Anhui, China (no. KJ2010A333) and National High Tech. Development Program of China (no. 2010ZX03003-002 and 2011ZX03004-002-01).

References

[1] S. Fu, K. Lu, T. Zhang, Y. Qian, and H. H. Chen, "Cooperative wireless networks based on physical layer network coding," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 86–95, 2010.

[2] S. W. Peters, A. Y. Panah, K. T. Truong, and R. W. Heath, "Relay architectures for 3GPP LTE-advanced," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, Article ID 618787, 14 pages, 2009.

[3] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: analog network coding," in *ACM SIGCOMM 2007: Conference on Computer Communications*, pp. 397–408, August 2007.

[4] S. Zhang, S. C. Liew, and P. P. Lam, "Hot topic: physical-layer network coding," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 358–365, September 2006.

[5] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Medard, "On practical network coding for wireless environments," in *Proceedings of the IEEE International Zurich Seminar on Digital Communications*, pp. 84–85, Zurich, Switzerland, 2006.

[6] C. H. Liu and F. Xue, "Network coding for two-way relaying: rate region, sum rate and opportunistic scheduling," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 1044–1049, May 2008.

[7] S. J. Kim, P. Mitran, and V. Tarokh, "Performance bounds for bidirectional coded cooperation protocols," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 5235–5241, 2008.

[8] Z. Ding, I. Krikidis, J. Thompson, and K. K. Leung, "Physical layer network coding and precoding for the two-way relay channel in cellular systems," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 696–712, 2011.

[9] R. Vaze and R. W. Heath, "On the capacity and diversity-multiplexing tradeoff of the two-way relay channel," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4219–4234, 2011.

[10] S. L. H. Nguyen, A. Ghrayeb, G. Al-Habian, and M. Hasna, "Mitigating error propagation in two-way relay channels with network coding," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3380–3390, 2010.

[11] C. Zhang, W. Wang, and G. Wei, "Design of ARQ protocols for two-user cooperative diversity systems in wireless networks," *Computer Communications*, vol. 32, no. 6, pp. 1111–1117, 2009.

[12] Q. T. Vien, L. N. Tran, and H. X. Nguyen, "Network coding-based ARQ retransmission strategies for two-way wireless relay networks," in *Proceedings of the 18th International Conference on Software, Telecommunications and Computer Networks (SoftCOM '10)*, pp. 180–184, September 2010.

[13] F. Iannello and O. Simeone, "Throughput analysis of type-I HARQ strategies in two-way relay channels," in *Proceedings of the 43rd Annual Conference on Information Sciences and Systems (CISS '09)*, pp. 539–544, March 2009.

[14] B. Zhao and M. C. Valenti, "Practical relay networks: a generalization of hybrid-ARQ," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 7–18, 2005.

[15] Z. Chen, Z. Chao, Z. Jun, and W. Guo, "ARQ protocols for two-way relay systems," in *Proceedings of the 6th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '10)*, pp. 1–4, September 2010.

[16] Q. Chen and M. C. Gursoy, "Energy efficiency and goodput analysis in two-way wireless relay networks," in *Proceedings of the 20th International Conference on Computer Communications and Networks (ICCCN '11)*, Maui, Hawaii, USA, 2011.

[17] P. Popovski and H. Yomo, "Physical network coding in two-way wireless relay channels," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 707–712, Glasgow, Scotland, June 2007.

[18] P. Popovski and H. Yomo, "Bi-directional amplification of throughput in a wireless multi-hop network," in *Proceedings of the 63rd IEEE Vehicular Technology Conference (VTC '06)*, pp. 588–593, Melbourne, Australia, May 2006.

[19] G. Yu, Z. Zhang, and P. Qiu, "Efficient ARQ protocols for exploiting cooperative relaying in wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2765–2773, 2007.

[20] R. Narasimhan, "Individual outage rate regions for fading multiple access channels," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '07)*, pp. 1571–1575, June 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 781682, 8 pages
doi:10.1155/2012/781682

Research Article

Distributed Routing and Spectrum Allocation Algorithm with Cooperation in Cognitive Wireless Mesh Networks

Zhigang Chen,[1] Zhufang Kuang,[1,2] Yiqing Yang,[1] Xiaoheng Deng,[1] and Ming Zhao[1]

[1] *School of Information Science and Engineering, Central South University, Changsha 410083, China*
[2] *School of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha 410004, China*

Correspondence should be addressed to Zhufang Kuang, zfkuangcn@csu.edu.cn

Received 7 January 2012; Revised 9 March 2012; Accepted 1 April 2012

Academic Editor: Shukui Zhang

Routing and spectrum allocation is an important challenge in cognitive wireless mesh networks. A distributed routing and spectrum allocation algorithm with cooperation (DRSAC-W) in cognitive wireless mesh networks is proposed in this paper. In order to show the decrease of the average end-to-end delay with cooperation in DRSAC-W, a distributed routing and spectrum allocation algorithm without cooperation (DRSAC-WO) is proposed in this paper. Minimizing the average end-to-end delay is the objective of DRSAC-W and DRSAC-WO. Simulation results show that the proposed algorithm DRSAC-W with cooperation can alleviate the high delay due to the heterogeneity of available channels of different nodes and achieve low average end-to-end delay.

1. Introduction

The scarcity of spectrum resource is often thought to be a bottleneck in wireless mobile communications. Cognitive radio (CR) is intelligent revolutionary spectrum (channel) sharing technology and the most important new wireless technology today. The core function of CR is that it can sense the vacancy spectrum resources and share these unused spectrum resources [1]. Secondary users (SU) can use the authorized spectrum which primary users (PU) did not use [2, 3].

A cognitive wireless mesh network (CWMN) is a wireless mesh network which integrates CR technology [4, 5]. A CR-Mesh node (such as a CR-Mesh gateway, a CR-Mesh router, or a CR-Mesh client), which integrates CR technology, can sense the spectrum which PU are not using and access the vacancy spectrum resource.

Wireless mesh networks (WMN) are a type of next generation broadband wireless access networks. There are many challenge problems in wireless mesh networks. Recently, there are some research results about routing and channel allocation [6–10]. However, research results of routing and channel allocation in WMN cannot be applied to CWMN directly, because the problem of routing and channel allocation in a CWMN has the following characteristics. (1) The routing protocol of WMN uses static channel, while the routing protocol of a CWMN must utilize dynamic channels. (2) The CR-Mesh node uses the allocated spectrum which the PU did not use; hence, the CR-Mesh node must ensure that it does not interfere with the communication of the PU. (3) The channels available to a CR-Mesh node are a subset of all available channels, and this subset changes over time in a CWMN. (4) There are heterogeneity available channel sets among different CR-Mesh nodes in a CWMN. (5) There are differences among the different channels, due to the activity of PU.

At present, the research about CWMNs is at an early stage. There are many open challenges [11] in CWMN. Although for the routing and spectrum allocation problems, there are already some research results [12–19].

An improved layered AODV route protocol in cognitive wireless mesh networks was proposed by Tingrui et al. [12]. An AODV-COG route protocol based on AODV protocol was proposed by Sun et al. The objective of AODV-COG is to increase the throughput of a CWMN [13]. An economic framework for adaptation and control of the network resources with the final goal of the network profit maximization was proposed by Amini and Dziong [14]. A multisource video on-demand application over a multiinterface cognitive wireless mesh networks was studied by Yong Ding with the objective of maximizing the number of sessions of the

network. A distributed multipath routing and spectrum allocation algorithm (DRCA) and a centralized multi-path routing and spectrum allocation algorithm (CRCA) were proposed by Ding and Xiao [15]. Lee et al. aim at solving the problem of coexistence of CWMN and other wireless networks, in order to share spectrum among multiple wireless networks. A route and spectrum allocation algorithm with the objective of minimizing the used spectrum was proposed [16].

With the optimization of average throughput and average delay, a distributed routing and channel allocation was proposed by Zhang et al. [17]. A multi-path routing and channel allocation strategy was proposed by Gu et al., with the goal of optimizing average throughput and average delay [18]. A dynamic layered-graph routing model and routing policy for CWMN were proposed by Li et al. [19].

The problem of routing and spectrum allocation with node cooperation is studied in this paper. We aim to minimize the end-to-end average delay.

This paper offers the following innovations when compared to existing research. (1) The effect among multiple wireless requests is taken into account, in order to minimize the average end-to-end delay. (2) The different wireless channels have different transmission characteristics, with delay being one of the most important of these characteristics. (3) DRSAC-WO, a distributed routing and spectrum allocation algorithm without cooperation, and DRSAC-W, a distributed routing and spectrum allocation algorithm with cooperation, are proposed in this paper.

The remainder of the paper is organized as follows. We discuss the network model and problem description in Section 2. In Section 3, we describe the proposed DRSAC-WO and DRSAC-W algorithms. Simulations comparing the performance of the proposed algorithms are presented in Section 4. Section 5 concludes the paper and outlines our future work.

2. Network Model and Problem Description

2.1. Network Model. We adopt a simple undirected graph $G = (V, E)$ model of the CWMN, which consists of CR-Mesh router and CR-Mesh gateways. V represents the set of CR-Mesh routers and CR-Mesh gateways. GW(GW $\subset V$) represents the set of CR-Mesh gateways. E represents the set of wireless links. Each node $v_i \in V$ has an available channel set K_i which has been sensed. Each node $v_i \in V$ has I_i cognitive radio interfaces (CRIs). T_R and I_R represent the communications distance and interference distance, respectively, and $I_R = 2 \times T_R$. The physics distance between node v_i and node v_j is represented by $d(v_i, v_j)$. Two CR-Mesh nodes which can communicate with each other must satisfy the following conditions. (1) There are common available channels, $K_i \cap K_j \neq \Phi$. (2) There are unoccupied CRIs for each node. (3) The nodes must satisfy the restriction of distance, $d(v_i, v_j) \leq T_R$. (4) The nodes must satisfy the restriction of interference.

There is interference between wireless links (u_1, v_1) and (u_2, v_2) which must satisfy the following condition. (1) $d(u_1, u_2) \leq I_R$ or $d(u_1, v_2) \leq I_R$ or $d(v_1, u_2) \leq I_R$ or $d(v_1, v_2) \leq I_R$, and (2) the same channel must have been allocated to two wireless links, $x(u_1, v_1) = x(u_2, v_2)$.

Table 1: Symbol implication.

Symbol	Implication
V	Sets of nodes $\lvert V\rvert = n$
E	Set of edges $\lvert E\rvert = m$
Δ	Set of wireless requests
K	Set of available channels
T_R	Communications distance
I_R	Interference distance
I_i	Available number of cognitive radio interfaces of node i
K_i	Available channel set of node i
D^k	Delay of channel k
$x(u, v)$	Allocation channel of wireless link (u, v)

$H(u, g_i)$ represents the hop count from CR-Mesh route node u to the CR-Mesh gateway node $g_i (g_i \in \text{GW})$.

$X = \{x(u, v)\}_{n*n}$, $x(u, v) = k$ that represents the wireless link (u, v) is allocated channel k. $x(u, v) = 0$ that represents the wireless link (u, v) is not allocated any channel. Every wireless link either is allocated only one channel or is not allocated a channel.

D^k represents the delay of the channel k $(k \in K, k \geq 1)$, in units of *ms*. Different channels have different delays, that is, different channels i and j lead to $D^i \neq D^j$. In order to describe the proposed algorithm, we assume that there is channel 0. The delay of channel 0 is $D^0 = \infty$. The meaning of other symbols are summarized in the Table 1.

2.2. Problem Description. We study the problem under the condition of heterogeneous available channels, and the route from source node to destination node is constructed distributedly. We aim to minimize the average end-to-end delay.

$\Delta = \{\delta_i = (s_i, d_i)\}$ represent the set of wireless requests, s_i and d_i represents the source node and destination node of wireless request $\delta_i \cdot \text{Path}(s_i, d_i)$ represents the path from source node s_i to destination node $d_i \cdot \text{Delay}(s_i, d_i)$ represents the average end-to-end delay of $\text{Path}(s_i, d_i)$, as computed with the following:

$$\text{Delay}(s_i, d_i) = \sum_{(u,v) \in \text{Path}(s_i, d_i)} D^k x(u, v) = k. \tag{1}$$

AvgDelay (Δ) represents the average end-to-end delay. Minimizing the average end-to-end delay is the goal and is formulated as follows:

$$\begin{aligned} &\text{Min} \quad \text{AvgDelay}\,(\Delta), \\ &\text{AvgDelay}\,(\Delta) = \frac{1}{|\Delta|} \sum_{\delta_i \in \Delta} \text{Delay}(s_i, d_i). \end{aligned} \tag{2}$$

A simple topology is considered. This topology is shown in Figure 1. There are 2 CR-Mesh gateways, and 10 CR-Mesh router nodes. CR-MR4 $\{1, 2, 3, 4, 5\}/3$ represents that the node CR-MR4 has the available channel set $\{1, 2, 3, 4, 5\}$,

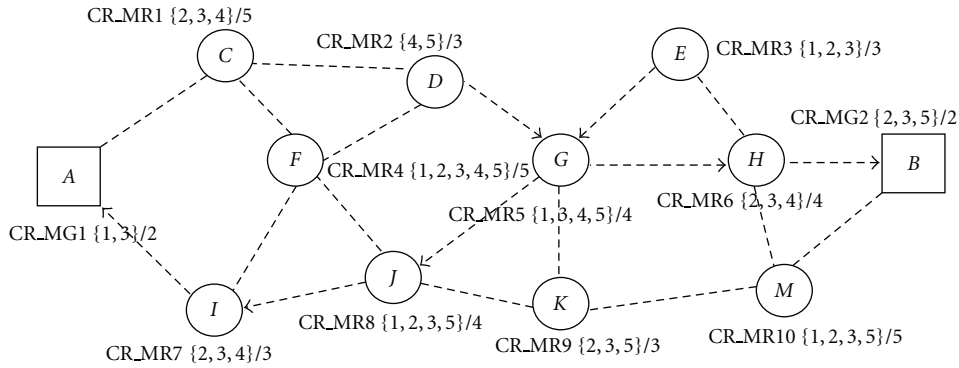


Figure 1: Cognitive wireless mesh network topology.

with five CRIs, $K_2 = 5$, and $I_2 = 5$. There are 5 available channels in wireless network, $K = \{1, 2, 3, 4, 5\}$, the delay of each of these is $D = \{3, 5, 6, 9, 2\}$.

$\delta_1 = (E, A)$ and $\delta_2 = (D, B)$ are two wireless requests in the network environment. Table 2 shows the constructed paths and spectrum allocations without cooperation. $\text{Path}(E, A) = E \xrightarrow{1} G \xrightarrow{5} J \xrightarrow{2} I \xrightarrow{3} A$ represents the constructed path of wireless request δ_1. It means that the allocated channel from node E to node G is channel 1, the allocated channel from G to node J is channel 5, the allocated channel from J to node I is channel 2, and the allocated channel from I to node A is channel 3. We can compute the $\text{Delay}(E, A) = 16$ and $\text{Delay}(D, B) = 20$ using (1). The following computes the average end-to-end delay:

$$\text{AvgDelay}(\Delta) = \frac{(\text{Delay}(E, A) + \text{Delay}(D, B))}{2} = 18. \quad (3)$$

Table 3 shows the constructed paths and allocated spectrum with cooperation. The delays are $\text{Delay}(E, A) = 20$ and $\text{Delay}(D, B) = 13$, and the average end-to-end delay is

$$\text{AvgDelay}(\Delta) = \frac{(\text{Delay}(E, A) + \text{Delay}(D, B))}{2} = 16.5. \quad (4)$$

The wireless request δ_1 arrives before wireless request δ_2. Without cooperation, the fundamental of spectrum allocated is channel with the lowest delay. When the wireless request δ_2 arrives, the wireless link $G \rightarrow J$ has been allocated channel 5. The wireless link $D \rightarrow G$ only can be allocated channel 4.

With cooperation, the channel allocated to the wireless link $G \rightarrow J$ is changed to channel 3, and the channel of the wireless link $G \rightarrow J$ is changed to channel 5. Although the $\text{Delay}(E, A)$ increases, the $\text{Delay}(D, B)$ decreases with cooperation. Additionally, the decrease in $\text{Delay}(D, B)$ is more than the increase in $\text{Delay}(E, A)$, thus, the overall average end-to-end delay decreases.

The claim of this paper is that making these types of choices will minimize the average end-to-end delays for all requests in the network.

Table 2: Path and delay without cooperation.

	$\text{Path}(s_i, d_i)$	$\text{Delay}(s_i, d_i)$
δ_1	$E \xrightarrow{1} G \xrightarrow{5} J \xrightarrow{2} I \xrightarrow{3} A$	16
δ_2	$D \xrightarrow{4} G \xrightarrow{3} H \xrightarrow{2} B$	20

Table 3: Path and delay with cooperation.

	$\text{Path}(s_i, d_i)$	$\text{Delay}(s_i, d_i)$
δ_1	$E \xrightarrow{1} G \xrightarrow{3} J \xrightarrow{2} I \xrightarrow{3} A$	20
δ_2	$D \xrightarrow{5} G \xrightarrow{3} H \xrightarrow{2} B$	13

3. Distributed Routing and Spectrum Allocation Algorithm

DRSAC-WO, a distributed routing and spectrum allocation algorithm without cooperation, and DRSAC-W, a distributed routing and spectrum allocation algorithm with cooperation, are proposed in this paper. In order to show the decrease in average end-to-end delay when there is node cooperation, we compare the two algorithms. The InitCRNode algorithm is common to both DRSAC-WO and DRSAC-W algorithms.

3.1. InitCRNode Algorithm. InitCRNode algorithm initializes all CR-Mesh nodes of the CWMN. The initialization constructs the neighbor node list, available channels of each neighbor node, and the hop count to CR-Mesh gateway node.

We must do some parts of this computation with a centralized algorithm rather than a distributed algorithm However, the choice of path to the gateway is based upon local information. $L(u)$ represents the information at node u and the neighbor information of node u. $L(u) \cdot \text{Set}$ represents the set of neighbor nodes of CR-Mesh router node u. Other related information is listed in Table 4.

The following formulas show how to compute $L(u) \cdot ax(u)$ and $L(u) \cdot AC(v)$:

$$L(u) \cdot ax(u) = K_u - L(u) \cdot x(u), \quad (5)$$

$$L(u) \cdot AC(v) = L(u) \cdot C(v) - L(u) \cdot UC(v). \quad (6)$$

Input: $ICM(v)$
Output: $ICM(u)$, $L(u)$
1. $L(u) \cdot \text{Set} \leftarrow \Phi \quad T_s \leftarrow 0$
2. if u is GW node {
3. $ICM(u) \cdot H(u,u) \leftarrow 0$
4. $ICM(u) \cdot Ch \leftarrow K_u$
5. Broadcast $ICM(u)$
6. Exit }
7. else if u is MR node {
8. $H(u,g_i) \leftarrow \infty \quad \forall g_i \in GW$;
9. While ($GetCurrTime() \le T_s$ or $L(u) \cdot Set = \Phi$){
10. if (u *receives* $ICM(v)$){
11. $L(u) \cdot \text{Set} \leftarrow L(u) \cdot \text{Set} \bigcup \{v\}$
12. $L(u) \cdot C(v) \leftarrow ICM(v) \cdot C$
13. $L(u) \cdot UC(v) \leftarrow \Phi$
14. $L(u) \cdot x(u) \leftarrow \Phi$
15. $L(u) \cdot x(u,v) \leftarrow 0$
16. if $|L(u) \cdot \text{Set}| = 1${
17. Init Timer T_s; }}// end if
18. }// end while
19. $L(u) \cdot H(u,g_i) \leftarrow$
$\text{Min}\{\text{Min}_{v \in L(u) \cdot \text{Set}}\{H(v,g_i)\} + 1,\ H(u,g_i)\} \quad \forall g_i \in GW$;
20. $ICM(u) \cdot H(u,g_i) \leftarrow L(u) \cdot H(u,g_i)$
21. $ICM(u) \cdot C \leftarrow K_u$;
22. Boardcast $ICM(u)$; }// end else if

Algorithm 1: InitCRNode algorithm.

Table 4: Information $L(u)$ of node u.

ID	Name	Description
1	$x(u)$	Set of used channel of node u
2	K_u	Set of available channel of node u
3	$ax(u)$	Set of allocable channel of node u
4	$H(u,g_i)$	Hop from node u to gateway node g_i
5	$x(u,v)$	Allocated channel for wireless link (u,v)
6	$C(v)$	Set of available channel of neighbor node v
7	$UC(v)$	Set of used channel of neighbor node v
8	$AC(v)$	Set of allocable channel of neighbor node v

$ICM(u)$ represents the initialization control information of node u.

$ICM(u) \cdot C = K_u$ represents the available channel set of node u.

$ICM(u) \cdot H(u,g_i) = H(u,g_i)$ represents the minimum hop count from node u to gateway node g_i. See Algorithm 1.

3.2. DRSAC-WO Algorithm. DRSAC-WO is a distributed routing and spectrum allocation algorithm without cooperation.

$UCM(u)$ represents the update control information of node u. $UCM(u)$ is sent when the allocated channel of node u changed.

$UCM(u) \cdot UC$ represents the set of channels used by node u. The choice of the next hop is that, the node which has the lowest delay channel in common with node u is chosen as the next hop node from the neighbor node set. If more than one neighbor has the same lowest delay common channel, then the node with the lowest hop count is chosen as the next hop node. The DRSAC-WO algorithm is shown below.

3.3. DRSAC-W Algorithm. DRSAC-W is a distributed routing and spectrum allocation algorithm with cooperation. The difference between the DRSAC-W and DRSAC-WO is (1) adding a cooperation request strategy to Algorithm 2 between line 16 and line 17 to DRSAC-W and (2) adding the cooperation response strategy for the neighbor node of node u to DRSAC-W.

$RCM(u)$ represents the information contained in the cooperation request from node u.

$RCM(u) \cdot x(u,v)$ represents the allocated channel of wireless link (u,v). The fundament of cooperation in DRSAC-W algorithm is that, node u sending the request cooperation control information to find the lower delay wireless link for wireless link (u,v). It must ensure that the sum of delays is lower than the sum of the earlier delays. Minimizing the average end-to-end delay is the goal.

$REM(v)$ represents the response information which is sent from node v to node u.

$REM(v) \cdot x(u,v)$ represents the allocated channel for wireless link (u,v).

Algorithm 3 shows the cooperation request strategy of node u, while Algorithm 4 shows the cooperation response strategy of node v which is the neighbor of node u.

Input: $\delta_i = (s_i, d_i)$, u
Output: $x(u, v)$
1. $j \leftarrow 0 \quad k \leftarrow 0$
2. if (u *receives* $UCM(x)$) {
3. $L(u) \cdot UC(x) \leftarrow UCM(x) \cdot UC$}
4. if $L(u) \cdot x(u) \geq I_u$ exit
5. Compute $L(u) \cdot ax(u)$ according to (5)
6. for each $y \in L(u) \cdot$ Set{
7. if $L(u) \cdot UC(y) \geq I_y$ continue
8. Compute $L(u) \cdot AC(y)$ according to (6)
9. $K(u, y) = L(u) \cdot ax(u) \bigcap L(u) \cdot AC(y)$
9. $j = \arg \text{Min}\{D^m\} \quad m \in K(u, y)$
10 if $(D^k > D^j)$ {
11. $k \leftarrow j \quad v \leftarrow y$
12. } else if $(D^k = D^j)$ {
13. if $L(v) \cdot H(v, d_i) > L(y) \cdot H(y, d_i)$ {
14. $k \leftarrow j \quad v \leftarrow y$}}
15. }//end for
16. $x(u, v) \leftarrow k$
17. $L(u) \cdot x(u) \leftarrow L(u) \cdot x(u) \bigcup \{x(u, v)\}$
18. $UCM(u) \cdot UC \leftarrow L(u) \cdot x(u)$
19. Boardcast $UCM(u)$

Algorithm 2: DRSAC-WO algorithm.

Input: $\delta_i = (s_i, d_i, \tau_i)$, u, $x(u, v) = k$
Output: $x(u, v) = k'$
1. Send $RCM(u)$
2. Init Timer T_w
3. while $(GetCurrTime() \leq T_w)$ {
4. if (u *receives* $REM(v)$ from z) {
5. Cancel timer T_w
6. Boardcast $FBM(u)$
7. $k' \leftarrow REM(v) \cdot x(u, v)$
8. $L(u) \cdot x(u) \leftarrow L(u) \cdot x(u) - \{x(u, v)\}$
9. $x(u, v) \leftarrow k'$
10. }// end while

Algorithm 3: Cooperation request strategy of node u.

The following formula shows the sum of delays for all edges in the network:

$$\Im = \sum_{x(u,v)=k} D^k \quad (u, v) \in E. \tag{7}$$

Before adjusting the channel, $x(u, v) = k1$, $x(v, w) = k2$, after adjusting the channel, $x(u, v) = k2$, $x(v, w) = k4$. α represents the delay difference of channel $k4$ and $k1$. The larger the value of α, the lower the average end-to-end delay.

4. Simulation and Results

In order to validate the efficiency of the algorithms proposed in this paper, we implemented the DRSAC-W, DRSAC-WO, and DRCA [15] algorithms using NS-2 [20].

The network topology that was simulated corresponds to the wireless access network of a university. There are some

Input: (u, v), $RCM(u)$
Output: $x(u, v)$
1. if (v *receives* $RCM(u)$ from u) {
2. Compute $L(v) \cdot ax(v)$ according to (5)
3. Compute $L(v) \cdot AC(u)$ according to (6)
4. $k1 \leftarrow RCM(u) \cdot x(u, v)$
5. $K(v, u) \leftarrow L(v) \cdot ax(v) \bigcap L(v) \cdot AC(u)$
6. for each $k2$
$k2 \in K(v, u) and (D^{k2} < D^{k1}))$ {
7. For each $y \in L(v) \cdot Set$ {
8. if $x(v, y) = k2$
9. Compute $L(v) \cdot AC(y)$ according to (6)
10. $K(v, y) \leftarrow L(v) \cdot ax(v) \bigcap L(v) \cdot AC(y)$
11. $k3 = \text{argMin}\{D^m\} \quad m \in K(v, y)$
$st \cdot D^{k3} < D^{k1}$.
12. $\beta \leftarrow D^{k1} - D^{k3}$
13. if $\alpha < \beta$ {
14. $\alpha \leftarrow \beta \quad k4 \leftarrow k3 \quad w \leftarrow y$}
15. } }// end for
16. $x(v, w) \leftarrow k4$
17. $REM(v) \cdot x(u, v) \leftarrow k2$
18. Send $REM(v)$
19. $L(v) \cdot x(v) \leftrightarrow L(v) \cdot x(v) \bigcup \{k4\} - \{k1\}$
20. $UCM(v) \cdot UC \leftarrow L(v) \cdot x(v)$

Algorithm 4: Cooperation response strategy of node v which is the neighbor node of node u.

available channels in 2000 m × 2000 m area. The PU uses the channel stochastically with $T_R = 50$ m and $I_R = 100$ m.

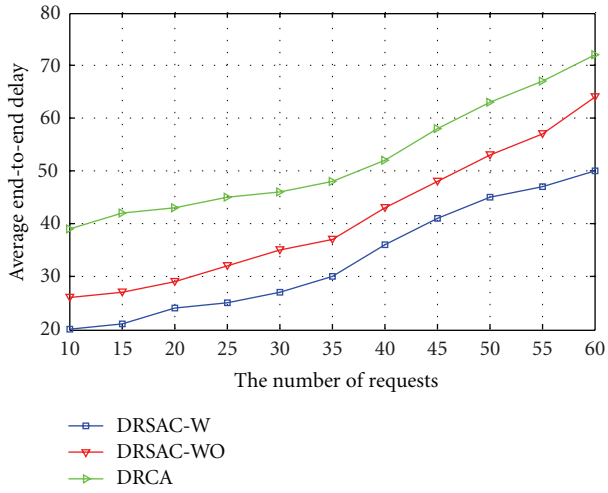
There are two network topologies with different numbers of nodes: $n = 25$ and $n = 50$. Two nodes are chosen randomly as the gateway nodes.The available number of channels for $n = 25$ and $n = 50$ are $|K| = 6$ and $|K| = 9$. The duration in seconds of each wireless request is randomly selected from the interval $[1, 10]$. The rate of wireless requests is 2 Mb/s. The delay in ms of each channel is a random value in the range $[1, 10]$. The simulated time is 200 s.

The simulation results that we report are the average of 500 simulation runs. The performance parameters that we report are the average end-to-end delay and average throughput.

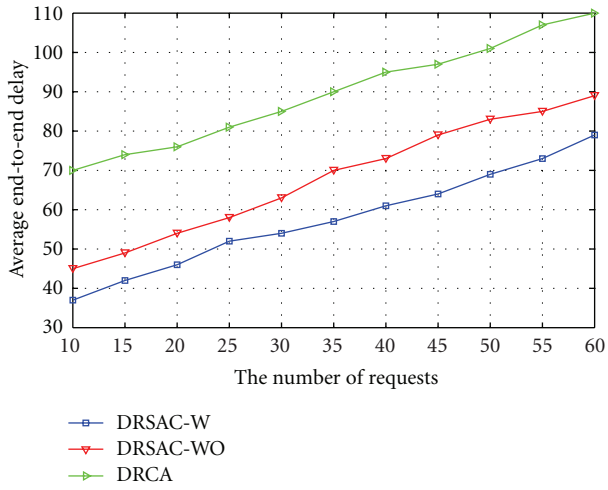
The simulation considers the following two aspects (1) Analyzing the performance of DRSAC-WO, DRSAC-W and DRCA with different numbers of requests. (2) Analyzing the performance of DRSAC-WO, DRSAC-W and DRCA with different numbers of available channels.

4.1. The Performance Comparison with Different Numbers of Requests. We analyse the performance of algorithms with different numbers of wireless requests. Figures 2 and 3 show the simulation results.

We can see from Figure 2, as the number of requests increases the average end-to-end delay increases for all three algorithms. This is because the available network resources do not change despite the increased number of wireless requests. Therefore, the average end-to-end delays increase.

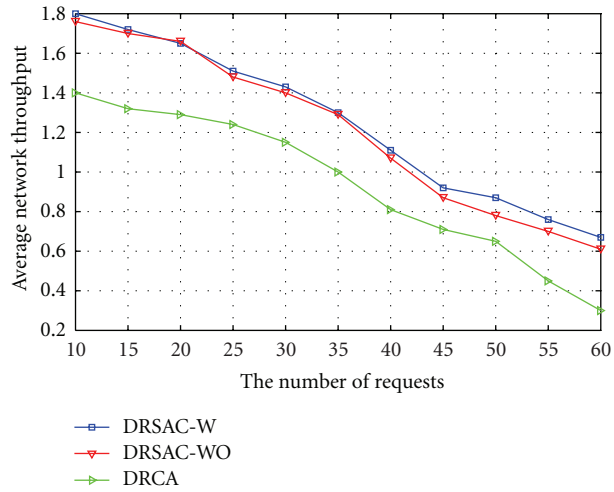


(a) $n = 25$

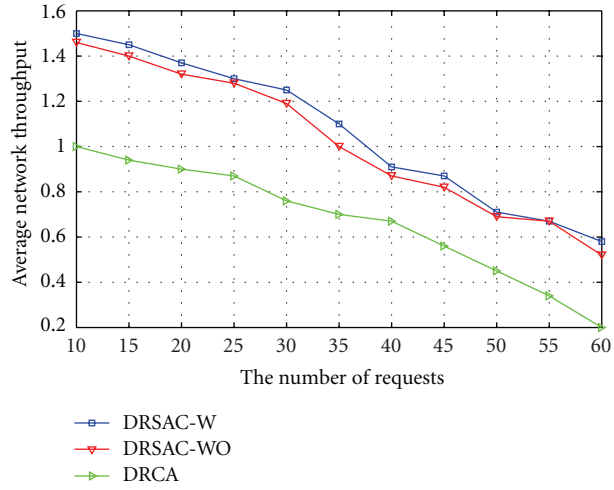


(b) $n = 50$

Figure 2: Average end-to-end delay with different numbers of requests.



(a) $n = 25$



(b) $n = 50$

Figure 3: Average throughput with different numbers of requests.

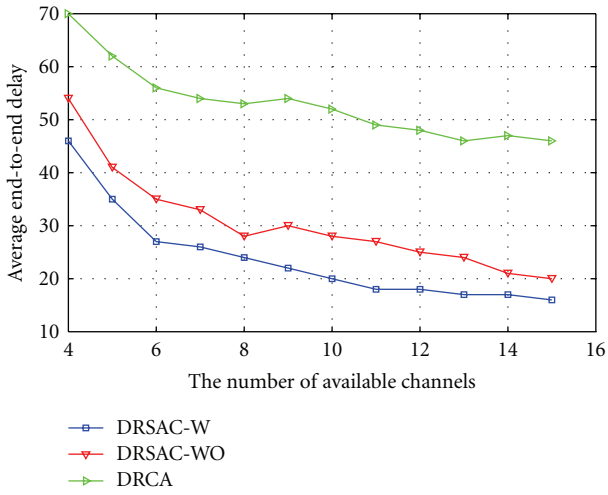
The average end-to-end delay of DRSAC-W and DRSAC-WO algorithm are less than for the DRCA algorithm. This is because DRSAC-W and DRSAC-WO algorithms choose the node, which has the lowest delay common channel as the next hop. Unlike our goal of minimizing the average end-to-end delay, minimizing the sum of bandwidths of each session is the goal of DRCA algorithm. Furthermore, the average end-to-end delay of DRSAC-W is less than that of the DRSAC-WO. This is because that the DRSAC-W algorithm reduces the average end-to-end delay due to node cooperation.

We can see from Figure 3, as the number of requests increases, the average throughput of all three algorithms deceases. This is because the available network resource does not change despite the number of wireless requests increasing. Additionally, the average throughput of DRSAC-W and DRSAC-WO algorithms is greater than is DRCA algorithm. The average throughput of DRSAC-W and DRSAC-WO iss the same. This is because that the difference between DRSAC-W and DRSAC-WO is that DRSAC-W algorithm adopt the node cooperation in order to decease end-to-end average delay.

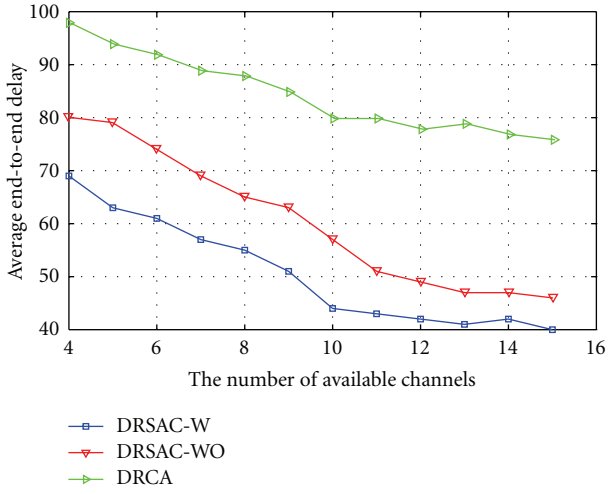
4.2. The Performance Comparison with Different Numbers of Available Channels. We analyse the performance of the three algorithms with different numbers of available channels via simulation. Figures 4 and 5 are the result of averaging the result of 500 simulations, when the number of wireless requests in each 200 second simulation run was 30.

We can see from Figure 4, as the number of available channels increases, the average end-to-end delay of all three algorithms decreases. This is because that the number of wireless requests did not change while the number of available channels increased. The average end-to-end delay of DRSAC-W and DRSAC-WO algorithms was less than for the DRCA algorithm.

We can see from Figure 5, as the number of available channels increase, the average throughput of all three

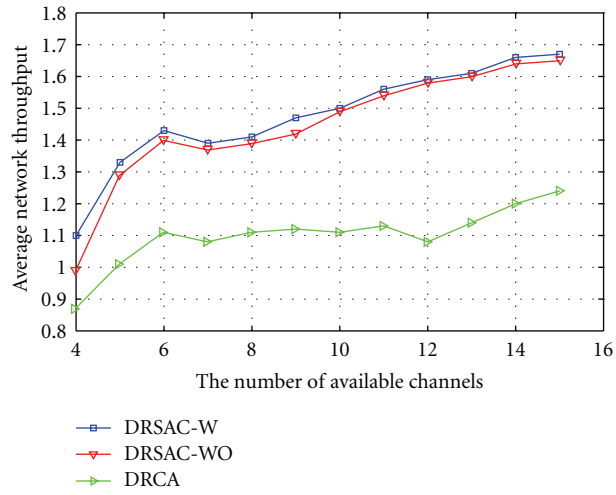


(a) $n = 25$

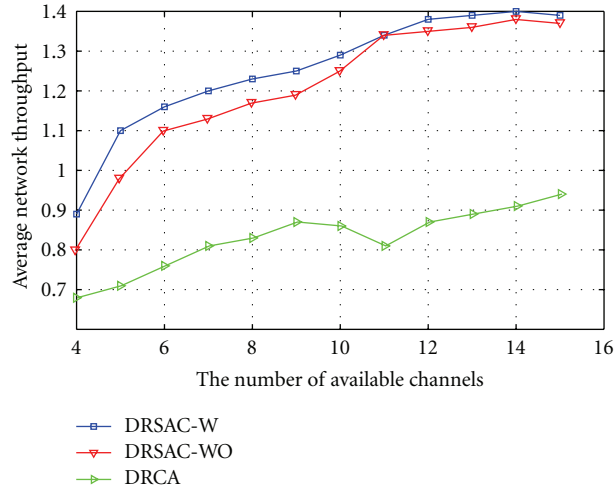


(b) $n = 50$

Figure 4: Average end-to-end delay with different numbers of available channels.



(a) $n = 25$



(b) $n = 50$

Figure 5: Average throughput with different numbers of available channels.

algorithms increases. Although, the average throughput of the DRSAC-W and DRSAC-WO algorithm is greater than for the DRCA algorithm. There is no difference between the DRSAC-W and DSRAC-WO algorithms.

5. Conclusion

The problem of routing and spectrum allocation with the goal of minimizing end-to-end average delay is researched in this paper. A distributed routing and spectrum allocation algorithm without cooperation and a distributed routing and spectrum allocation algorithm with cooperation are proposed in this paper. Simulation results show that DRSAC-W and DRSAC-WO algorithms can achieve low average end-to-end delay and high average throughput. The average end-to-end delay of DRSAC-W is less than DRSAC-WO, showing that the average end-to-end delay deceases with node cooperation. The problem of load balanced of routing and spectrum allocation will be addressed in our future work.

Acknowledgments

The authors would like to thank the reviewers for their detailed comments that have helped to improve the quality of the paper. This work is supported by National Natural Science Foundation of China under Grants no. 61073186, 61073104, 61070169, and 61170021; Natural Science Foundation of Jiangsu Province under Grant no. BK2011376; Specialized Research Foundation for the Doctoral Program of Higher Education of China no. 20103201110018.

References

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[2] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[3] J. Su, J. Wang, and W. Wu, "A truthful bilateral multiunit auction for heterogeneous cognitive radio networks," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 350476, 11 pages, 2011.

[4] G. F. Wu, Z. M. Ji, J. Zhang et al., "Cognitive wireless mesh networks," *Journal of Information Engineering University*, vol. 11, no. 8, pp. 429–433, 2010.

[5] N. Bouabdallah, B. Ishibashi, and R. Boutaba, "Performance of cognitive radio-based wireless mesh networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 122–135, 2011.

[6] R. Bruno and M. Nurchis, "Survey on diversity-based routing in wireless mesh networks: challenges and solutions," *Computer Communications*, vol. 33, no. 3, pp. 269–282, 2010.

[7] W. Si, S. Selvakennedy, and A. Y. Zomaya, "An overview of Channel Assignment methods for multi-radio multi-channel wireless mesh networks," *Journal of Parallel and Distributed Computing*, vol. 70, no. 5, pp. 505–524, 2010.

[8] Z. Tian, G. Leus, and V. Lottici, "Joint dynamic resource allocation and waveform adaptation for cognitive networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 2, pp. 443–454, 2011.

[9] L. Y. Huang, C. Liu, and S. P. Wang, "Improved spectrum sharing model in cognitive radios based on game theory," *Journal on Communications*, vol. 31, no. 2, pp. 136–140, 2010.

[10] Z. Y. Chai and F. Liu, "Spectrum allocation of cognitive wireless network based on immune clone selection optimization," *Journal on Communications*, vol. 31, no. 11, pp. 92–100, 2010.

[11] A. Al-Dulaimi, H. Al-Raweshidy, J. Cosmas, and J. Loo, "Cognitive mesh networks: Cognitive radio over fiber for microcells applications," *IEEE Vehicular Technology Magazine*, vol. 5, no. 3, pp. 54–60, 2010.

[12] P. Tingrui, Z. Zhi, Z. Wenli, and Z. Zhaoxia, "A cognitive improved hierarchical AODV routing protocol for cognitive wireless mesh network," *Information Technology Journal*, vol. 10, no. 2, pp. 376–384, 2011.

[13] X. B. Sun, Y. R. Zhang, and C. L. Zhao, "A new routing protocol in cognitive wireless mesh networks," in *Proceedings of the International Conference on Advanced Intelligence and Awareness Internet (AIAI '10)*, pp. 123–126, October 2010.

[14] R. M. Amini and Z. Dziong, "A framework for routing and channel allocation in cognitive wireless mesh networks," in *Proceedings of the 7th International Symposium on Wireless Communication Systems (ISWCS '10)*, pp. 1017–1021, September 2010.

[15] Y. Ding and L. Xiao, "Routing and spectrum allocation for video on-demand streaming in cognitive wireless mesh networks," in *Proceedings of the IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS '10)*, pp. 242–251, November 2010.

[16] D. H. Lee, W. S. Jeon, and D. G. Jeong, "Joint channel assignment and routing in cognitive radio-based wireless mesh networks," in *Proceedings of the IEEE 71st Vehicular Technology Conference (VTC '10)*, May 2010.

[17] G. A. Zhang, J. Y. Gu, and Z. H. Bao, "Distributed joint routing and channel allocation algorithm in cognitive wireless mesh networks," in *Proceedings of the 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT '10)*, pp. 432–437, October 2010.

[18] J. Y. Gu, G. A. Zhang, and Z. H. Bao, "Joint multi-path routing and channel assignment strategy for cognitive wireless mesh networks," *Computer Science*, vol. 38, no. 5, pp. 45–48, 2011.

[19] Y. Li, Y. N. Dong, and H. T. Zhao, "Dynamic layered-graph routing model and routing policy in cognitive radio mesh networks," *Journal of Electronics and Information Technology*, vol. 31, no. 8, pp. 1975–1979, 2009.

[20] K. Fall and K. Varadhan, NS manua l[EB/OL], 2011, http://www.isi.edu/nsnam/ns/.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 659235, 13 pages
doi:10.1155/2012/659235

Research Article

NUNS: A Nonuniform Network Split Method for Data-Centric Storage Sensor Networks

Ki-Young Lee,[1] Hong-Koo Kang,[2] In-Su Shin,[3] Jeong-Joon Kim,[3] and Ki-Joon Han[3]

[1] *Department of Medical IT and Marketing, Eulji University, Seongnam 461-713, Republic of Korea*
[2] *Team of Security Research & Development, Korea Internet & Security Agency, Seoul 138-803, Republic of Korea*
[3] *Division of Computer Science & Engineering, Konkuk University, Seoul 143-701, Republic of Korea*

Correspondence should be addressed to Jeong-Joon Kim, jjkim9@db.konkuk.ac.kr

Received 20 October 2011; Accepted 27 February 2012

Academic Editor: Hongli Xu

If data have the same value frequently in a data-centric storage sensor network, then the load is concentrated on a specific sensor node and the node consumes energy rapidly. In addition, if the sensor network is expanded, the routing distance to the target sensor node becomes longer in data storing and query processing, and this increases the communication cost of the sensor network. This paper proposes a nonuniform network split(NUNS) method that distributes the load among sensor nodes in data-centric storage sensor networks and efficiently reduces the communication cost of expanding sensor networks. NUNS splits a sensor network into partitions of nonuniform sizes in a way of minimizing the difference in the number of sensor nodes and in the size of partitions, and it stores data occurring in each partition in the sensor nodes of the partition. In addition, NUNS splits each partition into zones of nonuniform sizes as many as the number of sensor nodes in the partition in a way of minimizing the difference in the size of the split zones and assigns each zone to the processing area of each sensor node. Finally, we performed various performance evaluations and proved the superiority of NUNS to existing methods.

1. Introduction

With the recent development of wireless communication and microsensors via processing and storage functions, the application fields of sensor networks are being expanded to environment monitoring, location-based services, telematics, home networking, and so forth [1–5]. A sensor network is composed of hundreds of sensor nodes, and a large one contains even tens of thousands of sensor nodes. Each sensor node has one or more sensors that can measure surrounding environments. For example, such sensor nodes measure data such as temperature, moisture, and illumination in a scalar form [6–8].

Sensor networks are classified into three types according to the method of storing measured data [9, 10]. In the external storage sensor network, measured data are stored in an external storage (or the central system) of the sensor network. In the local storage sensor network, the sensor node that measures data in the sensor network stores the data within itself. And in the data-centric storage sensor network, data measured by the sensor node are stored in the corresponding sensor node according to the measured data value. Among them, the data-centric storage sensor network is proposed to solve the problem in the external storage sensor network, which is the load concentration on the sensor node nearest to the external storage, and to solve the problem in the local storage sensor network, which is the involvement of unnecessary sensor nodes in query processing. Therefore, active research is currently being made on the data-centric storage sensor network [7, 10–12].

In the data-centric storage sensor network, the sensor node in which data are stored is determined by the value of the measured data, and thus if the data frequently have the same value, then the load is concentrated on a specific sensor node and the sensor node consumes energy rapidly [13–16]. In addition, if the sensor network is expanded, the routing distance to the target sensor node becomes longer in data storing and query processing, and this increases the communication cost of the sensor network. Therefore, in the data-centric storage sensor network, it is important to enhance the energy efficiency of the sensor network by distributing the load among sensor nodes and reducing

the communication cost of expanding the sensor network [17].

Representative studies on data-centric storage sensor networks such as GHT [10], DIFS [14], and DIM [7] split a sensor network into zones of uniform size and assign the split zones to sensor nodes for data storage and query processing. DIM has been proved superior to GHT and DIFS in terms of data storage and query processing performance, since orphan zones that contain no sensor nodes can occur and data corresponding to the orphan zones are stored in a neighbor sensor node. Therefore, this results in the concentration of load on a specific sensor node [15, 17–19]. Recent techniques including ZS [19], KDDCS [17], and ZP/ZPR [18] have been proposed to solve the hot-spot problem of DIM. Here, a hot-spot means the sensor node with the highest energy consumption. However, recent techniques do not consider the randomness of the hash function of the data-centric storage sensor network and incur additional overhead to maintain their data structures and routing methods. In addition, with the expansion of the sensor network, the communication cost for data storage and query processing also increases.

To solve such problems, this paper proposes a nonuniform network split (NUNS) method that can distribute the load among sensor nodes in the data-centric storage sensor network and reduce the communication cost of data storage and query processing resulted from expanding the data-centric storage sensor network. For this, NUNS performs sensor network split in the form of kd-tree through two steps. First, NUNS splits the sensor network into partitions of nonuniform sizes in a way of minimizing the difference in the number of sensor nodes and in the size of the split partitions, and data occurring in each partition are stored and managed by sensor nodes in the partition. Therefore, since data measured in the sensor network are distributed to partitions, and the distance between the sensor node measuring data and the sensor node storing the data becomes shorter, NUNS can distribute the load among sensor nodes and consequently reduce the communication cost of data storage and query processing resulted from expanding the sensor network considerably. Second, NUNS splits each partition into zones of nonuniform sizes by as many as the number of sensor nodes in the partition in a way of minimizing the difference in the sizes of the split zones until only one sensor node is left in each zone. Through this process, NUNS can reduce the load concentration on a specific sensor node and the cost of unnecessary routing resulting from the existence of orphan zones in DIM.

This paper is organized as follows. Section 2 analyzes previous studies on the data-centric storage sensor network. Section 3 describes NUNS proposed in this paper and its algorithms. Section 4 conducts experiments for comparing NUNS with previous researches and presents the results. Lastly, Section 5 draws the conclusions.

2. Related Works

This section analyzes previous studies on the data-centric storage sensor network. Geographic hash table (GHT) [10] is an index that generates a geographical location-based on the value of measured data and stores the data in the sensor node nearest to the generated geographical location in the data-centric storage sensor network. GHT uses the "Put" operation for data storage and "Get" operation for query processing, and it uses GPSR [14] as a routing method for finding the sensor node with the required data. For example, if a sensor node calls Put (event, data), then a geographical location is generated as a result of event hashing, and data is stored in the sensor node nearest to the generated geographical location. On the other hand, if a sensor node calls Get (event) for query processing, then the query is transferred to the sensor node nearest to the geographical location generated as a result of event hashing, and the result of the query is returned.

In GHT, if d is given as the split level for reducing the data storage cost of sensor nodes upon the expansion of the sensor network, then the structured replication is used in which the whole sensor network is split into 4^d $(d \geq 0)$ regions of uniform size [10–12]. In the structured replication, the highest representative sensor node called the root point is designated, and the representative sensor node is designated for each region of the split level. If a query occurs, then it is transferred from the root point to the representative sensor nodes of lower levels. However, because the structured replication forms a hierarchical structure that has a representative sensor node for each split region, the query load is concentrated on the root point, and thus the energy of the root point is consumed rapidly.

Distributed Index for Features in Sensor networks (DIFS) [14] is an extended GHT that reduces the access load of sensor nodes and supports range queries. In order to solve the problem of load concentration on the root point in the structured replication of GHT, DIFS uses a variation of quad-tree in which a child node can have multiple parent nodes. In addition, compared to the structured replication that entirely accesses all index nodes in querying, DIFS reduces the number of accesses to index nodes and supports range queries using the range of data values stored in index nodes and the size of split regions. Like GHT, DIFS also uses GPSR as its routing method.

DIFS can reduce load concentration on high level sensor nodes in the structured replication of GHT and support range queries. However, compared to the structured replication, DIFS has a larger number of index nodes and consumes more energy throughout the entire sensor network. In addition, although DIFS can support range queries, it cannot support range queries for multidimensional data since the index is designed for one-dimensional data. What is more, if data of the same value occur frequently, then the load is concentrated on the corresponding node, and the expansion of the sensor network results in an increase of the communication cost. DIMENSIONS [13] also can be thought of as using the same set of primitives as GHT, but it allows the drill down search for objects within a sensor network while DIFS allows range queries on a single key in addition to other operations.

Distributed index for multidimensional data (DIM) [7] is an index that maps data domains to the region domains of

the sensor network by using kd-tree and stores data in a sensor node geographically close to the corresponding region. DIM splits the sensor network into regions of uniform size alternately between axis X and axis Y until each split region has only one sensor node. In DIM, the split region is called the *zone*. If a sensor node measures data, then it hashes the data to generate a zone code of bit string type that indicates a zone and stores the data in the sensor node of the zone corresponding to the generated zone code. If the corresponding zone is an orphan zone that does not have a sensor node, then the data are stored in the sensor node of the backup zone that is a neighbor zone for storing data to be stored in the orphan zone. DIM also uses GPSR as its routing method.

In addition, DIM can store data of multiple attributes and process a query with multiple attributes in the data-centric storage sensor network. However, DIM also has the problems of load concentration resulting from the occurrence of data with the same value and high communication cost resulted from expanding the sensor network. Furthermore, as data to be stored in an orphan zone are stored in the sensor node of the backup zone, the load on the sensor node of the backup zone increases. Therefore, DIM has the hot-spot problem in which the load is concentrated on a specific sensor network (called hot-spot) and the node consumes energy rapidly.

Several techniques including KDDCS [17], ZS [19], and ZP/ZPR [18] have been proposed to solve the hot-spot problem of DIM in the data-centric storage sensor network. KDDCS, based on kd-tree like DIM, splits the sensor network into zones of nonuniform sizes to contain a sensor node, unlike DIM, and rebalances kd-tree to distribute the load of sensor nodes. However, KDDCS needs to move the data of sensor nodes to their neighbors and visit a higher node to move lower nodes in kd-tree using its routing technique (i.e., it should send more data than DIM). ZS locally detects the hot-spots and tries to evenly distribute the load among the sensor nodes. Also, ZP distributes the load of hot-spots to several sensor nodes, and ZPR replicates the data of hot-spots to neighbor nodes. However, KDDCS, ZS, and ZP/ZPR do not consider the randomness of the hash function for the data-centric storage sensor network [15].

3. NUNS (Non-Uniformed Network Split)

This section explains NUNS proposed in this paper and its algorithms in detail.

3.1. Partition Generation. In order to efficiently distribute the load among sensor nodes and to reduce the communication cost resulted from expanding the sensor network, NUNS splits a sensor network into partitions of nonuniform sizes, and data measured in each partition are stored in the sensor nodes of the partition. Assuming that R (X-Y plane) is a bounding rectangle that contains all sensor nodes within the sensor network, R is split into rectangular *partitions* of nonuniform sizes alternately between the X axis and Y axis to minimize the differences in the number of sensor nodes and in the size of the split partitions.

If the number of sensor nodes in rectangle R is an even number, then the two split partitions have the same number of sensor nodes. However, if it is an odd number, then the two partitions cannot have the same number of sensor nodes. In such a case, rectangle R is split so that the larger partition has one more sensor node. Accordingly, the number of sensor nodes is equal or different at most by one between the two split partitions. This split process is repeated as many times as required alternately between the X and Y axis, and if it is repeated i ($i \geq 0$) times, then the number of partitions generated becomes 2^i. Particularly in NUNS, if the number of splits i increases (i.e., the number of partitions increases), then the storage load is decentralized since data measured in each partition are stored in the sensor nodes of the partition. Also, the query load is decentralized among the sensor nodes, but the entire communication cost for query processing is not improved sufficiently in the sensor network. For example, assume that D is the number of data measured, Q is the number of queries, and flooding cost is $O(\sqrt{n})$ with n sensor nodes, then the total communication cost of data storage and query processing is approximately $O(D\sqrt{n/2^i} + Q2^i\sqrt{n})$. Therefore, the optimal number of splits i should be determined in consideration of the frequency of data storage and the frequency of query processing in the sensor network.

In NUNS, R is split to be the minimum difference in the size between the two split partitions. Assuming that there are n sensor nodes, $\{S_1, S_2, \ldots, S_n\}$, in R. If the split axis is axis $X(Y)$ and n is an even number, then the split position of R is determined between $x(y)$-coordinate of the $(k/2)$th sensor node and $x(y)$-coordinate of the $((k/2)+1)$th sensor node among the n sensor nodes ordered by their $x(y)$-coordinates. If n is an odd number, then the split position of R is determined between $x(y)$-coordinate of the $((k+1)/2)$th sensor node and $x(y)$-coordinate of the $((k+1)/2+1)$th sensor node among the n sensor nodes ordered by their $x(y)$-coordinates.

And to minimize the difference in size between two split partitions, let MP mean the middle position on split axis $X(Y)$ of R. Assuming that n is an even number, if MP exists between $x(y)$-coordinate of the $(k/2)$th sensor node and $x(y)$-coordinate of the $((k/2)+1)$th sensor node among the n sensor nodes ordered by their $x(y)$-coordinates of split axis $X(Y)$ of R, MP is determined as the optimal split position. If MP is less than $x(y)$-coordinate of the $(k/2)$th sensor node, then this coordinate is determined as the optimal split position. On the other hand, if MP is more than $x(y)$-coordinate of the $((k/2)+1)$th sensor node, then this coordinate is determined as the optimal split position. Of course, if n is an odd number, then the optimal split position is determined similarly. In NUNS, each partition has a unique partition code that identifies its partition. A partition code is a bit-string composed of partition separators. Table 1 shows a partition separator.

In Table 1, the partition separator is a bit indicating whether a split partition is the left (lower) one or the right (upper) one. Bit 0 indicates that the partition is the left (lower) one and bit 1 the right (upper) one with $X(Y)$ axis as the split axis. Figure 1 shows an example of partition generation.

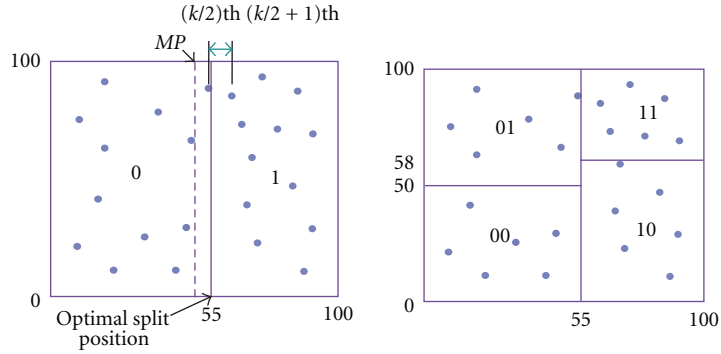


Figure 1: An example of partition generation.

Table 1: Partition separator.

Left or lower (1 bit)	Right or upper (1 bit)
0	1

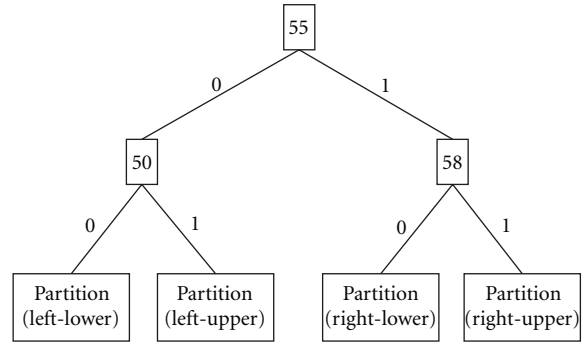


Figure 2: An example of a partition tree.

In Figure 1, the solid-line rectangles are partitions. As in Figure 1(a), the whole sensor network is first split into two nonuniformly partitions at position 55 as the optimal split position on the X axis since MP 50 is less than x-coordinate 55 of the $(k/2)$th sensor node. Therefore, the split partitions have minimum difference in the number of sensor nodes and in the size of the split partitions, and the partition codes of the two split partitions are 0 and 1, respectively.

Similarly, as in Figure 1(b), the left and right partitions in Figure 1(a) are split nonuniformly at position 50 and 58 on the Y axis to minimize the difference in the number of sensor nodes and in the size of the split partitions. As a consequence, the partition codes of the four split partitions, left-lower, left-upper, right-lower, and right-upper ones, are 00, 01, 10, and 11, respectively. Figure 2 shows the partition tree in the form of kd-tree for the four partition codes in Figure 1(b).

As in Figure 2, each node in the partition tree has the split position and two pointers for subnodes. To store data and process queries, the target sensor nodes are obtained by traversing from the root node down to the leaf nodes in the partition tree. In the partition tree traversal, if the partition is the left (lower) one, then the partition separator bit is set to 0 and it goes down to the left lower node, and if the partition is the right (upper) one, then the partition separator bit is set to 1 and it goes down to the right lower node. By using the partition tree, the diagonal coordinates of the partition can be obtained. For example, as the partition code of the left-lower partition obtained from the partition tree in Figure 2 is 00, the diagonal coordinates of the partition are (0, 0) and (55, 50).

In NUNS, we assume that the entire partition split process is performed by a sensor node, called the partition split node (PSN), which is a gateway node connecting the external base station and having higher power than others in the sensor network. The split process of PSN is as follows. *PSN* requests the locations of sensor nodes by flooding requests to all sensor nodes and performs the partition split process according to the partition generation algorithm by using responds from all sensor nodes in the sensor network. Finally, PSN floods the partition tree as a result of the partition split process to all sensor nodes in the sensor network. Since a flooding cost is $O(\sqrt{n})$ for point-to-point routing, where n is the number of sensor nodes, the time complexity of the algorithm is $O(\sqrt{n})$ times the diameter of the sensor network in general. Thus, the time complexity for constructing the partition tree is $O(d\sqrt{n})$, where d is the diameter of the sensor network. Algorithm 1 shows the partition generation algorithm in NUNS.

In the partition generation algorithm in Algorithm 1, input parameter *sensornetwork* is the partition to be split, *factor* is the number of partition splits, and *axis* is the split axis. Line 1 initializes variable *loc* for storing the optimal split position of the split axis and structure *pArray* for storing information on the split partitions. Line 2 calculates the optimal split position of the split axis that minimizes the difference in the number of sensor nodes and in the size of the split partitions and stores it in variable *loc*. Line 3 splits *sensornetwork* nonuniformly using the split axis and the optimal split position, and it stores information on the split partitions in *pArray*. Line 4 generates the partition tree for mapping specific data to a partition while the partition split process is performed, and Lines 5~7 switch the current split axis into the other for splitting *sensornetwork* alternately

```
GeneratePartition(sensornetwork, factor, axis)
Begin
  (1) i ← 0; loc ← null; f ← factor; initPartition (pArray[2]);
  (2) loc ← FindSplitPosition(sensornetwork, axis);
  (3) SplitPartition(sensornetwork, pArray, axis, loc);
  (4) UpdatePartitionTree(pArray, axis, loc);
  (5) if(axis) then
  (6)     axis ← 0;
      else
  (7)   axis ← 1;
      end if
  (8) for i from 0 to1 do
  (9)   if(factor > 1) then
  (10)    GeneratePartition(pArray[i].part, f-1, axis i);
        else
  (11)   GenerateZone(pArray[i].part, 0);
      end if
    end for
End
```

Algorithm 1: Partition generation algorithm.

between the X axis and Y axis. Lines 8∼11 check the number of partition splits and if the number is larger than 1, then the partition split process is performed repeatedly; if not, then *GenerateZone*() function is called to generate zones for the partition.

3.2. Zone Generation. Several sensor nodes can exist in a partition. In order to assign a processing region to each sensor node, NUNS splits each partition into zones of nonuniform sizes by as many as the number of sensor nodes in the partition. Assuming that an arbitrary initial partition is P (X-Y plane), partition P is split into rectangular *zones* of nonuniform size alternately between the X axis and Y axis to minimize the difference in the number of sensor nodes and the size of the split zones until only one sensor node is left in each zone.

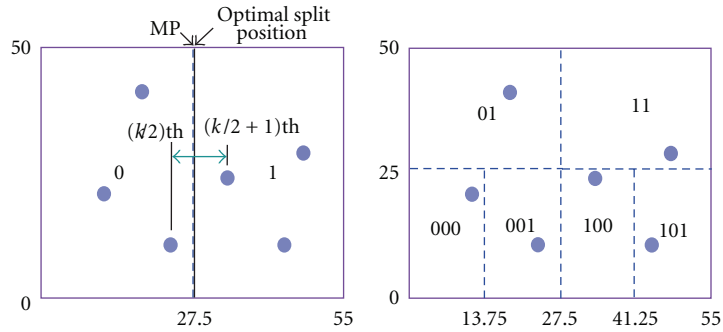
In the zone split process, if the number of sensor nodes in partition P is an even number, then the two split zones have the same number of sensor nodes, but if it is an odd number, then the two zones cannot have the same number of sensor nodes. In such a case, the larger zone has one more sensor node. This process is repeated by alternating between the X axis and Y axis until each of the zones of initial partition P contains one sensor node. Therefore, the number of zones in partition P is equal to the number of sensor nodes in partition P. In this way, zone generation is similar to partition generation, but different from partition generation; zones are generated as many as the number of sensor nodes in partition P and each zone contains one sensor node.

In NUNS, just like a partition has a partition code, a zone has a unique zone code that identifies the zone. Similar to a partition code, a zone code is also a bit string composed of zone separators. Figure 3 shows an example of zone generation from the left-lower partition in Figure 1(b).

In Figure 3, the dotted-line rectangles are zones. As in Figure 3(a), the partition is first split into two non- uniform zones at position 27.5 as the optimal split position on the X axis since MP 27.5 exists between x-coordinate of the $(k/2)$th sensor node and x-coordinate of the $(k/2+1)$th sensor node, so that the zones have minimum difference in the number of sensor nodes and the size of the split zones. Here, the zone codes of the two split zones are 0 and 1, respectively. In Figure 3(b), all zones have been generated in the partition via the zone split process, and the original partition has a number of zones equal to the number of sensor nodes in the partition. Figure 4 shows the zone tree in the form of kd-tree for the six zones in Figure 3(b).

As in Figure 4, each node in the zone tree has the split position and two pointers for subnodes. In data storage or query processing, the zone tree is used for obtaining target sensor nodes by traversing from the root node down to the leaf nodes. In the zone tree traversal, if the zone is the left (lower) one, then the zone separator bit is set to 0 and goes down to the left lower node, and if the zone is the right (upper) one, then the zone separator bit is set to 1 and goes down to the right lower node. As the diagonal coordinates of a partition are obtained from the partition tree, the diagonal coordinates of a zone can be obtained from the zone tree.

In NUNS, the entire zone split is performed by a sensor node, called the zone split node (ZSN), in each partition. NUNS selects a sensor node as ZSN, whose location is nearest to centroid of each partition. The split process of ZSN is as follows. ZSN requests the locations of sensor nodes by flooding requests to all sensor nodes in the partition and performs the zone split process according to the zone generation algorithm by using responds from all sensor nodes in its partition. Finally, ZSN floods the zone tree as a result of the zone split process to all sensor nodes in its partition. Since each partition has as many zones as the number of sensor nodes that it contains and 2^i partitions exist in the sensor network, where i is the level of the partition tree, the time complexity for constructing the zone trees



(a) Generation of two zones (b) Generation of all zones

Figure 3: An example of zone generation.

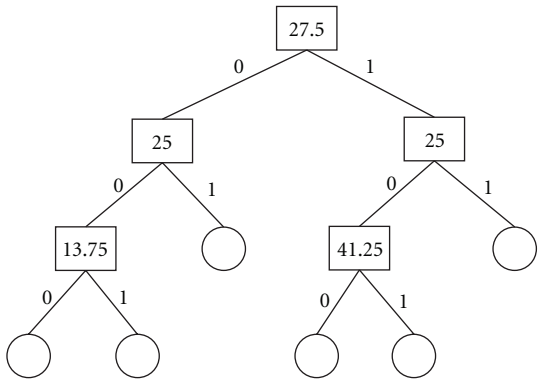


Figure 4: An example of a zone tree.

is $O(d/2^i\sqrt{n})$, where n and d are the number of sensor nodes and the diameter of the sensor network, respectively. Algorithm 2 shows the zone generation algorithm in NUNS.

In the zone generation algorithm in Algorithm 2, input parameter *partition* is the partition to be split and *axis* is the split axis. Line 1 initializes variable *loc* for storing the optimal split position of the split axis and structure *zArray* for storing information on split zones. Line 2 calculates the optimal split position of the split axis that minimizes the difference in the number of sensor nodes and in the size of split zones and stores it in variable *loc*. Line 3 splits *partition* nonuniformly using the split axis and the optimal split position, and it stores information on the split zones in structure *zArray*. Line 4 generates the zone tree for the split zones, and Lines 5~7 switch the current split axis into the other for splitting *partition* alternately between the X axis and Y axis. If each of the two zones obtained from splitting *partition* has two or more sensor nodes, then Lines 8~10 repeat the zone split process. Otherwise, it stops splitting the zones.

Compared to DIM that performs a uniform split, it seems that partition and zone generation in NUNS brings about overhead to store information on a nonuniform split. However, because each sensor node has information on the zones within the partition to which it belongs, NUNS can reduce the storage overhead for the index management and does not have orphan zones. Furthermore, since the nonuniform split of the sensor network is performed periodically in NUNS, the sensor network can operate normally even if the sensor nodes are added or deleted. And compared to KDDCS that performs storage load balancing of sensor nodes, though load balancing in NUNS is not sufficient, it can reduce the storage overhead of hot-spots and, unlike KDDCS, does not need additional routing overhead for storing data, processing queries, and maintaining kd-tree.

3.3. Data Storage. In NUNS, data measured in a partition are stored and managed in the sensor nodes within the partition. If a sensor node measures data, then the sensor node hashes the data value to find a zone that contains a sensor node for storing the data using the zone tree for the partition, and it then stores the data in the sensor node of the zone. Here, the corresponding zone is determined by traversing the zone tree from the root node down to leaf nodes and comparing the split axis and the optimal split position of the split axis stored in the nodes with the measured data value.

For example, let us assume that data have two attributes, each of which can have a value between 0 and 1. Then, because the ranges of the X axis and Y axis of a partition are mapped to the ranges of data attribute values in NUNS, the values mapped the X axis and Y axis of the partition range between 0 and 1. In Figure 5, the numbers on the two lines in parallel with the coordinate axes show the values of the attributes normalized between 0 and 1.

In our case, if $\langle 0.3, 0.8\rangle$ is the data measured by sensor node **A** as shown in Figure 5, then zone tree traversing for the data is performed as follows. First, from the root node, because the value 0.3 of the first attribute in the measured data is smaller than the mapping value 0.5 corresponding to the optimal split position 27.5 of the split X axis, then it goes down to the left child node. Then, because the value 0.8 of the second attribute in the measured data is larger than the mapping value 0.5 corresponding to the optimal split position 25 of the split Y axis, it goes down to the right child node. At this time, because the accessed lower node is a leaf node, its zone code is determined. The zone code is 01, and the data is stored in the sensor node within the zone indicated by the zone code. Figure 5 shows an example of data stored in this way.

```
GenerateZone(partition, axis)
Begin
  (1) i ← 0; loc ← null; initZone(zArray[2]);
  (2) loc ← FindSplitpoint(partition, axis);
  (3) SplitZone(partition, zArray, axis, loc);
  (4) UpdateZoneTree(zArray, axis, loc);
  (5) if(axis) then
  (6)     axis ← 0;
      else
  (7)   axis ← 1;
      end if
  (8) for i from 0 to1 do
  (9)   if(GetSensorNumber(zArray[i]) > 1) then
  (10)          Generate Zone (zArray [i].zone, axis);
      end if
    end for
End
```

Algorithm 2: Zone generation algorithm.

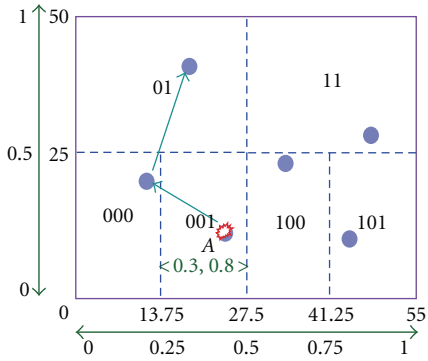


Figure 5: An example of data storage.

As shown in Figure 5, data measured in a partition is stored in the sensor nodes within the partition, and this can prevent the concentration of data storing load on a specific sensor node when data of the same value occur frequently in the sensor network. In addition, this can reduce the communication cost for storing data because the distance between the sensor node measuring data and the sensor node storing the data is shortened. Moreover, as every zone has one sensor node in NUNS, there is no orphan zone that can exist in DIM, and this can reduce the cost of unnecessary routing. Algorithm 3 shows the data storing algorithm in NUNS.

In the data storing algorithm shown in Algorithm 3, input parameter *loc* is the location of the current sensor node that generates data, and *data* is the data to be stored. Here, *data* can contain one or more attribute data. Line 2 hashes *data* and determines the centroid of the target zone using the zone tree. Lines 3∼4 check whether or not the target zone contains the current sensor node. If it does, since the current sensor node is the target sensor node for storing the data, then *data* is stored and the data storing algorithm terminates. If not, then Line 5 gets the number of neighbor sensor nodes in order to find a sensor node for transferring the data to the target zone, and Lines 6∼9 find the location of the sensor node nearest to the target zone among the neighbor sensor nodes and stores the location in variable *dl*. If it is not found, then Lines 10∼11 switch the routing path to the right and get the location of the sensor node there, and they store the location in variable *dl*. Line 12 calls *StoreData*() function with the location of the sensor node and data to be stored.

3.4. Query Processing. In NUNS, data measured in a partition is stored and managed in the sensor nodes within the partition. Therefore, if a query occurs in a sensor node, then the query should be transferred to all sensor nodes, which will process the query, in all the partitions of the sensor network. In order to transfer the query to all partitions, the sensor node in which the query occurred should know the locations of all partitions. The locations of all partitions can be determined by traversing the partition tree. If a query reaches any sensor node of a partition by using the partition tree, the query should be transferred to the target sensor node in the partition, which will process the query. Figure 6 shows an example of a query that is transferred to the four partitions of Figure 1(b). Numbers on the two lines in parallel with the coordinate axes show the values of the attributes normalized between 0 and 1.

As in Figure 6(a), sensor node *A* finds the centroids of all partitions using the partition tree and transfers query ⟨0.3–0.7, 0.8–0.9⟩ to them. The query from sensor node *A* is transferred first to sensor nodes *B*, *C*, and *D* in the partitions. In addition, as shown in Figure 6(b), each of sensor nodes *A*, *B*, *C*, and *D* hashes query ⟨0.3–0.7, 0.8–0.9⟩ to generate a centroid of the target zone, finds the target sensor node in the its zone, and transfers the query to the target sensor node. In the hashing process, the query is decomposed into subqueries according to the data range of each zone, and its target zone is determined for each of the decomposed subqueries. In addition, the decomposed subquery is transferred to the sensor node in the target zone

```
StoreData(loc, data)
Begin
  (1) n, i ← 0; ctz, tl ← null; dl ← loc;
  (2) ctz ← HashData(dara);
  (3) if(CheckInternalSensor(dl)) then
  (4)    AddData(data);
      else
  (5)   n ← CheckNeighbor(dl);
  (6)   for i from 1 to n do
  (7)     tl ← GetNeighborLocation(i);
  (8)     if(IsNearest(ctz, tl, dl) then
  (9)       dl ← tl;
          end if
  (10)    if(dl == loc) then
  (11)      dl ← RightRoutingQuery(loc, ctz);
          end if
      end for
  (12) StoreData(dl, data);
      end if
End
```

Algorithm 3: Data storing algorithm.

that is determined by traversing the zone tree from the root node down to the leaf nodes and by comparing the split axis and the optimal split position of the split axis stored in the nodes and the attribute values used in the query.

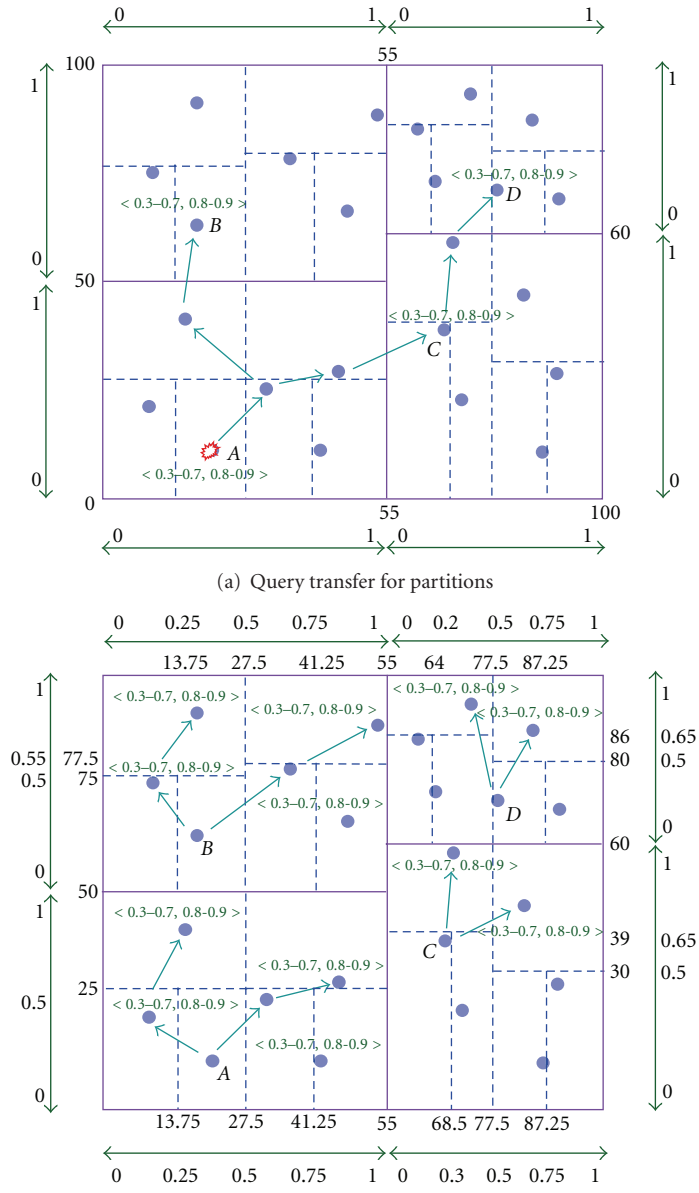
For example, the zone codes for query ⟨0.3–0.7, 0.8–0.9⟩ occurring in sensor node A in the left lower one among the four partitions of Figure 6(b) are determined as follows. First, from the root node in the zone tree in Figure 4, because the value of the first attribute is ⟨0.3–0.7⟩, the query is decomposed into subqueries ⟨0.3–0.5, 0.8–0.9⟩ and ⟨0.5–0.7, 0.8–0.9⟩ based on the mapping value 0.5 corresponding to the optimal split position 27.5 of the split X axis, and they are transferred to the left and right lower nodes, respectively. In the subquery ⟨0.3–0.5, 0.8–0.9⟩ transferred to the left lower node, since the value of the second attribute ⟨0.8–0.9⟩ is larger than the mapping value 0.5 corresponding to the optimal split position 25 of the split Y axis, it goes down to the right lower node. Because the right lower node is a leaf node, its zone code becomes 01. Next, in the subquery ⟨0.5–0.7,0.8–0.9⟩ transferred to the right lower node, since the value of the second attribute ⟨0.8–0.9⟩ is larger than the mapping value 0.5 corresponding to the optimal split position 25 of the split Y axis, it goes down again to the right lower node. Because the right lower node is a leaf node, its zone code is 11. Consequently, the query is decomposed into two subqueries with zone codes 01 and 11, respectively.

In NUNS, since the measured data of a sensor network are distributedly stored among the partitions, the communication cost of sensor nodes to store the data can be reduced. However, because the queries should be transferred to all the partitions and their results should be returned to the queried sensor node, the communication cost of the sensor network can increase due to query processing. Algorithm 4 shows the algorithm for transferring a query to all partitions and returning the results.

In the query transfer algorithm for partitions shown in Algorithm 4, input parameter *loc* is the location of the current sensor node that transfers a query, *query* is the query, and *ptroot* is the root node of the partition tree. Line 1 initializes all variables to be used in the algorithm. Lines 2∼13 transfer the query to partitions by as many as the number of partitions in the sensor network. Line 2 generates centroids of partitions to which the query will be transferred using *ptroot*. Line 3 checks whether or not the partition contains the current sensor node. If it does not, then Line 4 gets the number of neighbor sensor nodes in order to find a sensor node for transferring the query. Lines 5∼8 find the location of the sensor node nearest to the partition, to which the query will be transferred, among the neighbor sensor nodes and store the location in variable *dl*. If it is not found, then Lines 9∼10 switch the routing path to the right and get the location of the sensor node there, and they store the location in variable *dl*. Line 11 calls *TransferZQuery*() function with the location of the sensor node stored in variable *dl* and query to be transferred to find the target sensor node in the corresponding partition.

Algorithm 5 shows the algorithm for transferring a query to the target sensor nodes in zones and processing the query.

In the query transfer algorithm for zones shown in Algorithm 5, input parameter *loc* is the location of the current sensor node that transfers a query, *query* is the query, and *ztroot* is the root node of the zone tree. Line 1 initializes all variables to be used in the algorithm. In case the range of the attribute values of the query covers multiple zones, Line 2 decomposes the query according to the zones of the corresponding partition and generates the centroids of zones to which the query will be transferred using *ztroot*. Line 3 checks whether or not the zone contains the current sensor node, and if it does, then the query is processed in the current sensor node. If it does not, then Line 5 gets the number of



(a) Query transfer for partitions

(b) Query transfer for zones

FIGURE 6: An example query transfer.

neighbor sensor nodes in order to find a sensor node for transferring the query. Lines 8~9 find the location of the sensor node nearest to the target zone among the neighbor sensor nodes and store the location in variable *dl*. If it is not found, then Lines 10~11 switch the routing path to the right and get the location of the sensor node there, and they store the location in variable *dl*. Line 12 calls *TransferZQuery*() function with the location of the sensor node stored in variable *dl* and the query to be transferred.

4. Performance Evaluation

This section compares performance in terms of data storage and query processing among NUNS proposed in this paper, DIM which is superior to GHT and DIFS, and KDDCS which efficiently performs storage load balancing of sensor nodes. In the performance evaluation, we used a computer system with Intel Core2 CPU 2.13 GHz, 2 GB RAM, and Windows XP Professional. In addition, we simulated the sensor networks of sizes ranging from 200 to 1,000 sensor nodes, each having an initial energy of 10,000 units, a radio range of 40 m, and a storage capacity of 20 units. Especially, the locations of the sensor nodes were arbitrarily deployed into the sensor network boundary.

4.1. Data Storage Cost. In this section, we compared NUNS with DIM and KDDCS in terms of communication cost for storing measured data in a sensor network. In the

```
TransferPQuery(loc, query, ptroot)
Begin
  (1) n, i ← 0; ctp, tl ← null; dl ← loc;
      do
  (2)   ctp ← HashQuery(ptroot);
  (3)   while (!CheckInternalPartition(dl)) do
  (4)     n ← CheckNeighbor(dl);
  (5)     for i from 0 to n do
  (6)       tl ← GetNeighborLocation(i);
  (7)       if(IsNearest(ctp, tl, dl)) then
  (8)         dl ← tl;
        end if
      end for
  (9)     if(dl == loc) then
  (10)      dl ← RightRoutingQuery(loc, ctp);
      end if
      end while
  (11) TransferZQuery(dl, query, ztroot);
  (12) RespondResult(query);
  (13) end while(ctp)
End
```

Algorithm 4: Query transfer algorithm for partitions.

```
TransferZQuery(loc, query, ztroot)
Begin
  (1) n, i ← 0; ctz, tl ← null; dl ← loc;
      do
  (2)   ctz ← HashQuery (ztroot);
  (3)   if(CheckInternalPartition(dl)) then
  (4)     RespondResult(query);
      else
  (5)     n ← CheckNeighbor(dl);
  (6)     for i from 0 to n do
  (7)       tl ← GetNeighborLocation(i);
  (8)       if(IsNearest(ctz, tl, dl)) then
  (9)         dl ← tl;
    end if
    end for
  (10)    if(dl == loc) then
  (11)    dl ← RightRoutingQuery(loc, ctz);
    end if
  (12) TransferZQuery(dl, query, ztroot);
      end if
  (13) end while (ctz)
End
```

Algorithm 5: Query transfer algorithm for zones.

experiment, each sensor node generated three data with two attributes randomly while increasing the size of the sensor network by increasing the number of sensor nodes from 200 up to 1,000. Especially, in order to impose a storage hot-spot on the sensor network, for each network size, we generate a series of hot-spots where a percentage of 10% to 80% of the data fell into a percentage of 5% to 10% of the range of each attribute value. Figures 7 and 8 show the data storage cost in the sensor network and at the hot-spot, respectively. NUNS L0, NUNS L1, NUNS L2, and NUNS L3 mean that the split is performed 0, 1, 2, and 3 times, respectively (i.e., they have 1, 2, 4, and 8 partitions, resp.).

As shown in Figure 7, as the size of the sensor network becomes larger, the communication cost for data storage of NUNS with many partitions is more efficient than that of DIM and KDDCS. This is because by storing the data value measured in a partition in a sensor node within the partition in NUNS, the distance between the sensor node

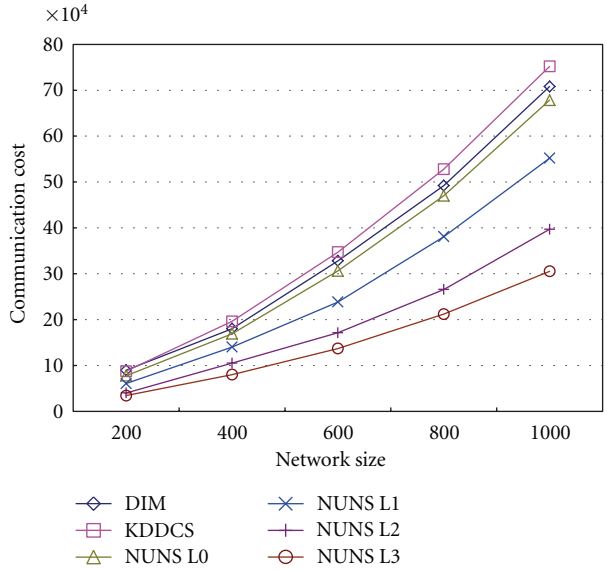


Figure 7: Data storage cost in sensor network.

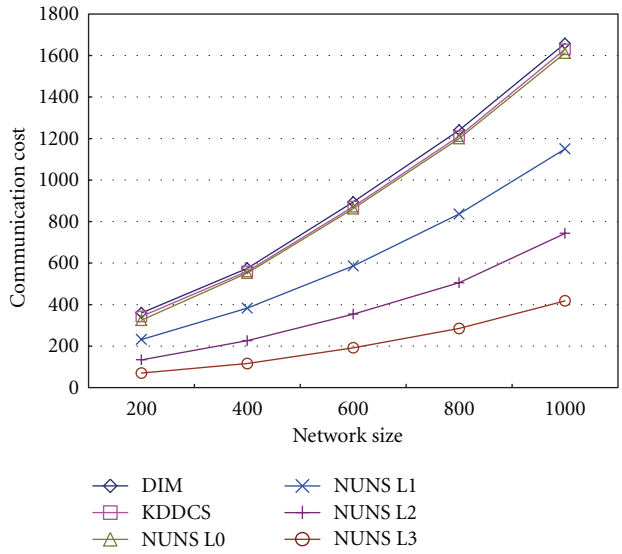


Figure 8: Data storage cost at hot-spot.

measuring data and the sensor node storing the data is shortened, and consequently the communication cost for storing data is reduced. In addition, the cost of unnecessary routing can be avoided in NUNS as a partition is split nonuniformly into zones and there is no orphan zone, unlike DIM. Especially, the communication cost of KDDCS can increase since additional overhead for rebalancing kd-tree is needed in KDDCS.

Similar to the data storage cost of the sensor network in Figure 7, as the size of the sensor network becomes larger, the communication cost of data storage at the hot-spot in NUNS with many partitions also appeared more efficient than that in DIM and KDDCS, as is shown in Figure 8. This is because by storing the data value measured in the sensor network distributedly among the partitions in NUNS, the

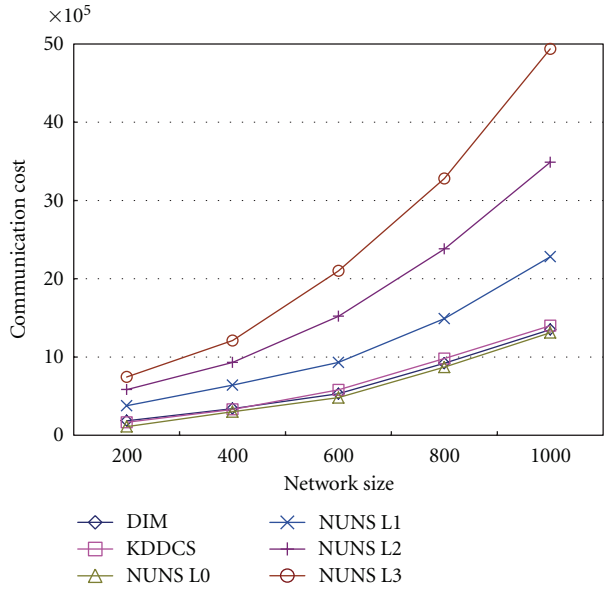


Figure 9: Query processing cost in sensor network.

storage load in the hot-spot can be reduced, and by splitting each partition into zones by minimizing the difference in the size of the split zones in NUNS, load concentration on a specific zone with a large processing region can be prevented. Especially, KDDCS can consume additional energy to move data of the hot-spot to neighbors for load balancing, which shorten the lifetime of the hot-spot.

4.2. Query Processing Cost. We compared NUNS with DIM and KDDCS in terms of communication cost for query processing in the sensor network. In the experiment, each sensor node generated two range queries with two attributes at random while increasing the size of the sensor network with the number of sensor nodes from 200 up to 1,000. For each network size, the queries were executed within 10% of the maximum range of the attribute values. Figures 9 and 10 show the query processing cost in the sensor network and at the hot-spot, respectively.

As shown in Figure 9, with the increase of sensor network size, NUNS with 1 partition was a little more efficient in terms of communication cost for query processing than DIM and KDDCS. However, NUNS with 2, 4, or 8 partitions showed a higher communication cost for query processing than DIM and KDDCS. This is because the communication cost for query processing particularly increases when the number of partitions becomes larger, since a query should be transferred to the target sensor nodes of all the partitions and its result should be returned to the sensor node in which the query occurred.

As shown in Figure 10, the communication cost for query processing at the hot-spot in NUNS with 1 partition appeared less efficient than that of KDDCS and more efficient than DIM, but the cost in NUNS with 2, 4, or 8 partitions was more efficient than that of DIM and KDDCS. This is because NUNS can distribute the storage load of the hot-spot

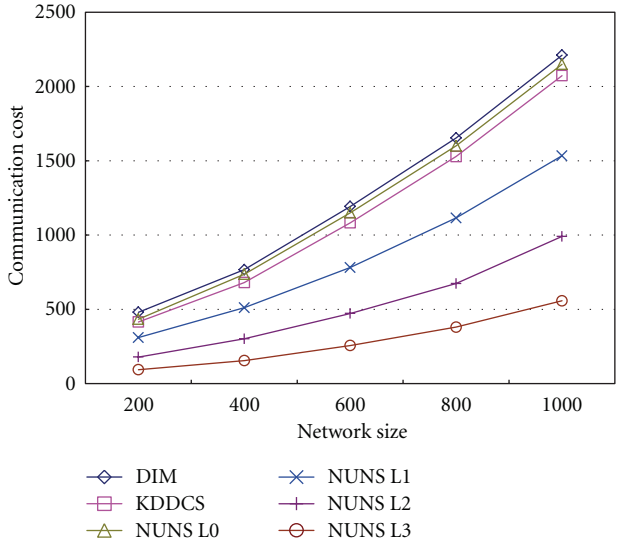


Figure 10: Query processing cost at hot-spot.

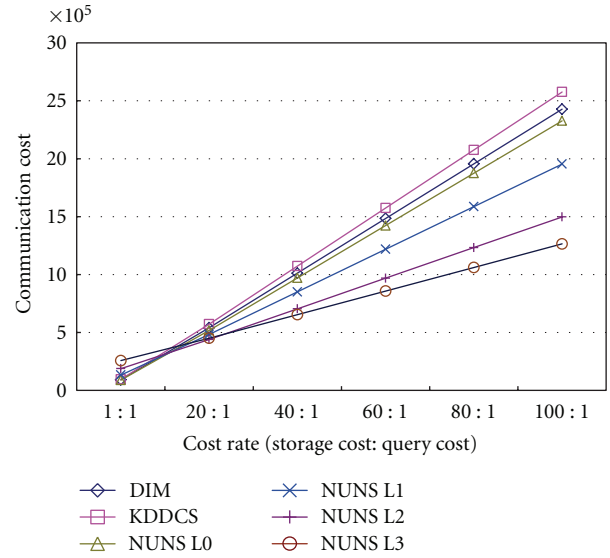


Figure 11: Communication cost in sensor network.

among partitions in the sensor network and consequently reduce the transfer cost of query results at the hot-spot. Therefore, the optimal number of partitions in NUNS should be determined in consideration of the frequency of data storage and the frequency of query processing in the data-centric storage sensor network.

4.3. Communication Cost according to Cost Ratio. This section experimented on the communication cost of the sensor network and that of the hot-spot according to the ratio of the frequency of data storage to the frequency of query processing. That is, we compared NUNS with DIM and KDDCS in terms of communication cost while changing the ratio from 1 : 1 up to 100 : 1. In the experiment, the communication range of the sensor nodes was set to 40 m and 1,000 sensor nodes were used. In addition, 100~10,000 data with two attributes and 100 range queries with two attributes were generated at random. For each network size, the queries were executed within 10% of the maximum range of the attribute values. Figures 11 and 12 show the communication cost according to the ratio in the sensor network and at the hot-spot, respectively.

As shown in Figure 11, in comparison with DIM and KDDCS, NUNS with 1 partition was most efficient in terms of communication cost of the sensor network when the ratio of the frequency of data storage to the frequency of query processing was 1 : 1~20 : 1, and NUNS with 8 partitions was most efficient when the ratio was 40 : 1~100 : 1. In addition, the communication cost efficiency of NUNS with the large number of partitions is expected to be higher than DIM and KDDCS as the frequency of data storage becomes larger than the frequency of query processing. This is because with the increase in the number of partitions in NUNS, the cost of data storage decreases but the cost of query processing increases, and consequently the decrease in the cost of data storage becomes relatively larger than the increase in the cost

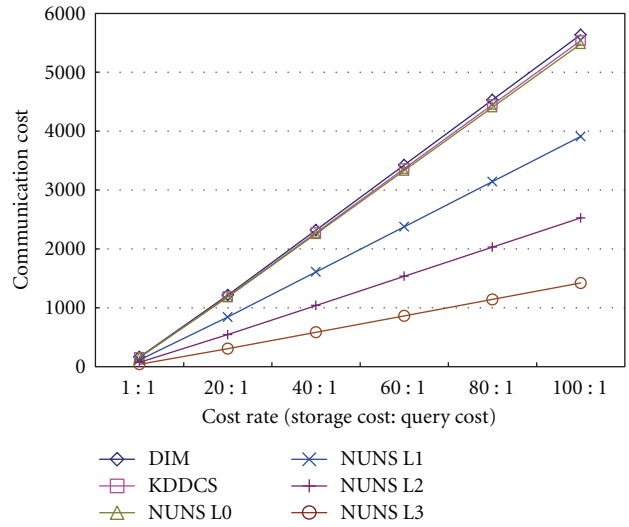


Figure 12: Communication cost at hot-spot.

of query processing, as the frequency of data storage becomes higher than the frequency of query processing.

Similar to the communication cost of the sensor network in Figure 11, in comparison with DIM and KDDCS, the communication cost at the hot-spot as shown in Figure 12 also revealed the efficiency of NUNS with a large number of partitions as the frequency of data storage becomes higher than the frequency of query processing.

The results of the experiments above show that in all cases, NUNS with one partition is more efficient for data storage and query processing than DIM and KDDCS. In addition, with the increase in the number of partitions in NUNS, the cost of query processing increases, but the cost of data storage decreases. Accordingly, we expect to enhance the energy efficiency of the sensor network by using NUNS in the data-centric storage sensor network where data values are stored frequently.

5. Conclusions

In the data-centric storage sensor network, the sensor nodes for data storage are determined by the value of measured data, and thus if data have the same value frequently, then the load is concentrated on a specific sensor node and the sensor node consumes energy rapidly. In addition, if the sensor network is expanded through the addition of new sensor nodes, then the distance between the senor node measuring data and the sensor node storing the data grows longer, and this increases the communication cost in data storage and query processing. Therefore, it is important to enhance the energy efficiency of the sensor network by distributing the load among the sensor nodes and by reducing the communication cost resulted from expanding the sensor network.

To solve these problems, this paper proposed a nonuniform network split method, called NUNS, for the data-centric storage sensor network. In order to distribute the load among sensor nodes and reduce the communication cost for data storage and query processing resulted from expanding the sensor network, NUNS splits a sensor network into partitions of nonuniform sizes and stores data that occurs in each partition and is managed by sensor nodes within the partition. In addition, for preventing load concentration on a specific sensor node and reducing the cost of unnecessary routing, NUNS splits each partition into zones of nonuniform sizes by as many as the number of sensor nodes in the partition. Lastly, this paper proved through experiments that NUNS is more energy efficient than DIM and KDDCS in the data-centric storage sensor network where data are stored more frequently.

Acknowledgment

This research was supported by a Grant (07KLSGC05) from Cutting-edge Urban Development—Korean Land Spatialization Research Project funded by Ministry of Land, Transport and Maritime Affairs of Korean Government.

References

[1] S. C. Draper and G. W. Wornell, "Side information aware coding strategies for sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 966–976, 2004.

[2] Z. He, B. S. Lee, and X. S. Wang, "Aggregation in sensor networks with a user-provided quality of service goal," *Information Sciences*, vol. 178, no. 9, pp. 2128–2149, 2008.

[3] 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*, pp. 243–254, Boston, Mass, USA, August 2000.

[4] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks," in *Proceedings of the 5th Symposium on Operating System Design and Implementation*, pp. 131–146, 2002.

[5] Q. Ren and Q. Liang, "Energy and quality aware query processing in wireless sensor database systems," *Information Sciences*, vol. 177, no. 10, pp. 2188–2205, 2007.

[6] R. Doss, G. Li, V. Mak, S. Yu, and M. Chowdhury, "Improving the QoS for information discovery in autonomic wireless sensor networks," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 334–349, 2009.

[7] X. Li, Y. J. Kim, R. Govindan, and W. Hong, "Multidimensional range queries in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 63–75, Los Angeles, Calif, USA, November 2003.

[8] 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, Atlanta, Ga, USA, September 2002.

[9] X. Liu, Q. Huang, and Y. Zhang, "Balancing push and pull for efficient information discovery in large-scale sensor networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 3, pp. 241–251, 2007.

[10] S. Ratnasamy, B. Karp, L. Yin et al., "GHT: a geographic hash table for data-centric storage," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 78–87, Atlanta, Ga, USA, September 2002.

[11] S. Ratnasamy, D. Estrin, R. Govindan, B. Karp, and S. Shenker, "Data-centric storage in sensornets," in *Proceedings of the 1st ACM SIGCOMM Workshop on Hot Topics in Networks*, pp. 137–142, 2003.

[12] S. Ratnasamy, B. Karp, S. Shenker et al., "Data-centric storage in sensornets with GHT, a geographic Hash Table," *Mobile Networks and Applications*, vol. 8, no. 4, pp. 427–442, 2003.

[13] D. Ganesan, D. Estrin, and J. Heidemann, "Dimensions: why do we need a new data handling architecture for sensor networks?" in *Proceedings of the 1st ACM Workshop on Hot Topics in Networks*, pp. 143–148, 2002.

[14] B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker, "DIFS: a distributed index for features in sensor networks," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 163–173, Anchorage, Alaska, USA, May 2003.

[15] Y. Lai, Y. Wang, and H. Chen, "Energy-efficient robust data-centric storage in wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 2735–2738, Shanghai, China, September 2007.

[16] M. Sharifzadeh and C. Shahabi, "Supporting spatial aggregation in sensor network databases," in *Proceedings of the 12th ACM International Symposium on Advances in Geographic Information Systems*, pp. 166–175, Washington, DC, USA, November 2004.

[17] M. Aly, K. Pruhs, and P. K. Chrysanthis, "KDDCS: a load-balanced in-network data-centric storage scheme for sensor networks," in *Proceedings of the 15th ACM Conference on Information and Knowledge Management*, pp. 317–326, Arlington, Va, USA, November 2006.

[18] M. Aly, P. K. Chrysanthis, and K. Pruhs, "Decomposing data-centric storage query hot-spots in sensor networks," in *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous '06)*, pp. 1–9, San Jose, Calif, USA, July 2006.

[19] M. Aly, N. Morsillo, P. K. Chrysanthis, and K. Pruhs, "Zone sharing: a hot-spots decomposition scheme for data-centric storage in sensor networks," in *Proceedings of the 2nd International Workshop on Data Management for Sensor Networks*, pp. 21–26, August 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 274576, 11 pages
doi:10.1155/2012/274576

Research Article

An Efficient Clustering Algorithm in Wireless Sensor Networks Using Cooperative Communication

Shukui Zhang,[1,2] Jianxi Fan,[1] Juncheng Jia,[1] and Jin Wang[1]

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *State Key Lab. for Novel Software Technology, Nanjing University, Nanjin 210093, China*

Correspondence should be addressed to Shukui Zhang, zhangsk2000@163.com

Received 7 January 2012; Accepted 27 February 2012

Academic Editor: Hongli Xu

Processing the gathered information efficiently is a key functionality for wireless sensor networks. In generally, the sensor networks often use in-network data aggregation and clustering to optimize network communication. The set of aggregating nodes forms a dominating set of the network graph. Finding the weakly connected dominating set (WCDS) is a promising approach for clustering the WSN. However, finding a minimum WCDS is NP-hard problem for most graphs, and a host of approximation algorithm has been proposed. The aim of the paper is to construct a minimum WCDS as a clustering scheme for WSN. Our clustering schemes construction algorithm includes two phases. First of all, we construct a maximal data aggregation tree (DAT) of the network. The second phase of the algorithm is to choose the nodes (called connectors) to make the WCDS connected. The correctness and performance of our algorithms are confirmed through theoretical analysis and comprehensive simulations.

1. Introduction

A wireless sensor network (WSN) is a multihop wireless communication network. In WSN, each node assumes the role of a router and relays the packets toward the final destinations if a source cannot directly send the packets to a final destination due to the limitation of the radio transmission range. In addition, the energy efficiency is one of the major constraints in WSN. The network topology may also change unpredictably due to node failure, running out of power, or adding new nodes into the network. Most topology changes are localized within a small area of the network. Therefore, it is desirable to abstract the network structure as local changes which need not be seen by the entire network. This is done by using logical substructures called clusters. It is believed that clustering can dramatically improve a network's broadband utilization and delivery ratio, extend network lifetime, and reduce packet retransmission [1]. A natural method for forming clusters is based on the idea of graph domination [2]. The most basic clustering methods that have been studied in ad hoc networks and WSN are based on the dominating sets (DSs). Moreover, among various existing clustering schemes, dominating set-based clustering [3, 4] is a promising approach.

The main advantage of dominating set-based clustering is that it simplifies the clustering process to the one in a smaller subnetwork generated from the connected dominating set (CDS). The efficiency of this approach depends largely on the process of finding and maintaining a CDS and the size of the corresponding sub-network. In addition, the CDS formation algorithm should be localized (i.e., based on local information) for low overhead and fast convergence. The research that works on selecting a minimum CDS has never been interrupted because of its dramatic contributions to wireless networks. Unfortunately, finding a minimum CDS is NP complete for most graphs, even if global information is available and no constraint [5].

In addition, in wireless channels, packets are usually dropped when the channel goes into deep fade and thus an outage occurs. In particular, the outage happens when instantaneous channel capacity falls below the amount of information carried in the packet [6]. Recently, the cooperative communication technique was exploited to study energy management issues for ad hoc and sensor networks [7, 8].

Such as in [7], a network model using cooperative communication is developed to deal with broadcasting in ad hoc networks and WSN. Transmitting independent copies of a packet generates diversity and combats the effects of fading. The selected relay r_k cooperates with one another if the direct transmission fails to the final destinations D. Each relay will decide whether it can successfully decode the M sources' information based on its local channel information. The criterion used for successful decoding is that its local channel information can satisfy the condition

$$\sum_{m\in S} R_m \le \log\left(1+\rho\sum_{m\in S}\left|h_{mr_k}\right|^2\right), \quad \forall S\subseteq\{1,2,\ldots,M\}, \tag{1}$$

where ρ is the SNR [9]. h_{mr_k} is the coefficient of the channels between mth source and kth relay. The above expression characterized the capacity region multiple access channels [10]. Assume that the Q relays R can satisfy the criterion and hence be able to decode the M sources' information correctly. By ordering the relay-destination channels, we denote the Q-qualified relays as $R_1, R_2, \ldots, R_Q$, where $|h_{R_nD}|^2 \ge |h_{R_{n+1}D}|^2$. The study has shown that cooperative communication can potentially combine the following advantages: (1) the power saving provided by multihopping, (2) the spatial diversity provided by the antennas of separate mobile nodes, and (3) node cooperation can also lead to increased data rates [11, 12].

Motivated by cooperative communication in ad hoc networks and WSN, Alzoubi et al. proposed an algorithm for weakly connected dominating set (WCDS) based on a spanning tree [4]. In this scheme, a maximal independent set (MIS) is elected such that each node in the MIS can be connected to the spanning tree via an extra node. Chen and Liestman [13, 14] proposed a zonal algorithm, in which the graph is divided into regions, a WCDS is constructed for each region, and adjustments are made along the borders of the regions to produce a WCDS for the whole graph. Their algorithm for the partitioning phase is partly based on a minimum spanning tree (MST) algorithm of Gallager et al. [15]. Han and Jia [16] also proposed an area-based distributed algorithm for WCDS construction in ad hoc networks with constant approximation ratio, linear time, and message complexity. While it has a lower message complexity than the zonal algorithm proposed by Chen and Liestman, it outperforms the mentioned algorithm. Basagni et al. [17] presented a performance comparison of the protocols proposed for clustering and backbone formation in large scale ad hoc network. Wu [3] presented two distributed algorithms for finding a WCDS in ad hoc networks. The first algorithm was implemented by first electing a leader among the nodes, which was going to be the root of a spanning tree. The spanning tree is then traversed, and the dominator nodes are selected. But the distributed leader election is extremely expensive in practice and exhibits a very low degree of parallelism. The second algorithm first constructs a maximum independent set (MIS) by an iterative labeling strategy and then modifies the MIS by selecting one intermediate node between each pair of dominators separated by exactly three hops.

At present, the study of WCDS is not more. As mentioned different above, we consider the WCDS as a better method for clustering [4] an ad hoc network and WSN. In this paper, based on the characteristics of communication under the cooperative communication, we extend the dominative capability of nodes in the corresponding network, and we turn the clustering scheme construction problem of a cooperative network into the WCDS problem in the graph model of cooperative communication. A novel algorithm (called DAT-WCDS) to find WCDS for clustering in ad hoc networks and WSN is proposed. And their good performance is confirmed by simulations.

2. Preliminaries and Definitions

2.1. A Network Environment. In this paper, the aim of the proposed algorithm is to form a clustering scheme for the WSN by finding a connected dominating set problem. We consider a monitor area A with N wireless sensors, represented by the set $S = (s_1, s_2, \ldots, s_N)$ randomly deployed. Each sensor node is equipped to learn its location coordinates such as its location information (x_i, y_i) [18]. It is not the purpose of this paper to define mechanisms to find this location. Without loss of generality, let us assume that nodes in the set S belong to two dimensional planes as illustrated in Figure 1.

At first, the goal of the proposed algorithm is to construct the data aggregation tree (DAT) in this N nodes network, where DAT is consisted of N_t nodes called tree node, which is used to receive and aggregate data, the other $(N - N_t)$ nodes are referred to as non-tree (NT) nodes. Each NT node senses its environmental parameter and reports it to its nearest tree node. The DAT is well spread over the entire WSN so that N_t tree nodes are uniformly distributed on the network. In this way, it ensures that the attribute readings sent by NT nodes to the corresponding tree node incur a smaller hop count. For simplicity, we use R_{event} (denoted by the dashed rectangle in Figure 1) to represent an event, and the event region is denoted by the area P_{event}, where $R_{\text{event}} \subseteq R$. Normally all the events are assumed to have already been sensed in the network by DAT. R' is defined as the portion of R not occupied by any event, that is, $R' = R - R_{\text{event}}$.

2.2. Connected Dominating Set. For simplicity, we assume a simple and yet general enough model that is widely used in the community. Wireless sensor networks are modeled as unit disk graphs $G = (V, E)$. Where, the vertices in V represent the communication nodes. Let $V' \subseteq V$ be a subset of vertices in $G = (V, E)$. In the following, we use $G[V']$ to denote the subgraph induced by V'. For a subgraph G' of G, we use $V(G')$ and $E(G')$ to refer to the vertices and edges of G'; respectively, we denote by $\Gamma(v)$ the closed neighborhood of a vertex $v \in V$, that is,

$$\Gamma(v) = \{u \in V \mid (u, v) \in E\} \cup \{v\}. \tag{2}$$

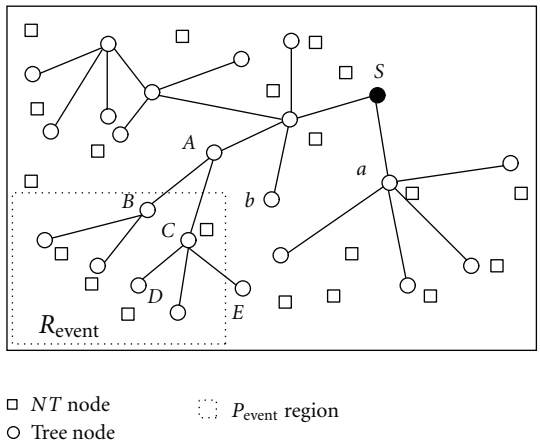


Figure 1: Network environment.

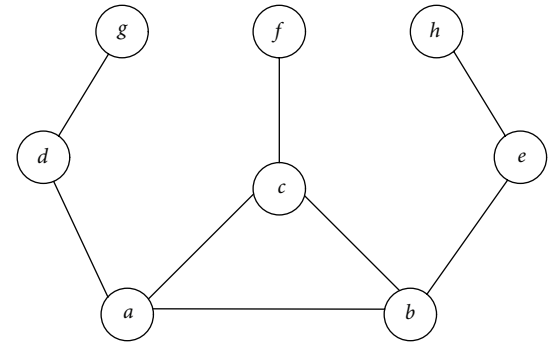


Figure 2: $\{a, b\}$ is EDS.

Analogously, for $V' \subseteq V$, $\Gamma(V) = \cup_{w \in V'} \Gamma(w)$ define the neighborhood of V'. In this context, we set $\Gamma(\Phi) = \Phi$, for $k \in N$, and we call $\Gamma_k(v) = \Gamma(\Gamma_{k-1}(v))$ the recursively defined $r$kth neighborhood of $v \in V$, $\Gamma_0(v) = \{v\}$.

A normal transmission range r, using the Euclidean distance $d(u, v)$, denoting the number of hops on a shortest path in G between vertices u and v, where $d(u, v)$ is also viewed as the transmission cost between u and v. This means that two vertices are connected by an edge if and only if u's disk covers v and v's disk covers u. Let $p(u, v) = \{u, w_1, w_2, \ldots, w_k, v\}$ be a shortest path between node u and v.

In graph theory, a dominating set (DS) of a graph $G = (V, E)$ is a subset $S \subseteq V$, such that every vertex $v \in V$ is either in S or adjacent to a vertex of S. A minimum DS (MDS) is a DS with the minimum cardinality $\gamma(G)$. A subset $I \subseteq V$ is called independent if for every two vertices $u, v \in I$, there does not exist an edge $(u, v) \in E$. An independent set is called maximal if it cannot be extended by the addition of any other vertices from the graph. There is an important relationship between maximal independent sets and dominating sets in a graph; an independent set is also a dominating set if and only if it is a maximal independent set [4].

A CDS S of a given graph G is a dominating set whose induced subgraph, denoted $G[S]$, is connected, and a minimum CDS (MCDS) is a CDS with the minimum cardinality. A dominating set S is a weakly connected dominating set (WCDS) of a graph G, if the graph $G[S] = (\Gamma(S), E \cap (\Gamma(S) \times S))$ is a connected subgraph of G. In other words, the weakly induced sub graph $G[S]$ contains the vertex of S, their neighbors, and all edges with at least one endpoint in S.

Finding the minimum WCDS of the network graph is one of the most investigated methods for cluster formation in which a dominator node assumes the role of a cluster head, and its one-hop neighbors and 2-hop neighbors are assumed to be cluster members. The structure of the network graph can be simplified using WCDS and made more succinct for transmitting in ad hoc networks and WSN [15, 16].

In this paper, we focus on clustering mechanisms to elect a minimum and sufficient number of links to serve as the communication backbone of the network. Accordingly, the clustering approach to topology management can be modeled as the relevant minimum WCDS problem in graph theory.

2.3. Dominating Set Extension. In this subsection, we extend the dominative capabilities of nodes for finding a small WCDS for a WSN. Wu et al. proposed the notion of an extended dominating set (EDS) [12]. A subset S of V is an EDS if every node of V is (a) in the subset, (b) a regular neighbor of a node in S, or (c) 2-hop neighbor of k nodes in S.

Dominative capabilities extension of nodes: each node is extended such that it dominates not only itself and its 1-hop distance neighbors fully, but also its 2-hop distance neighbors partly. For example, in Figure 2, the node dominates not only itself and nodes d, c, b fully, but also nodes g, e, f partly. This extension extends the dominative capability of a node from its 1-hop neighbors to its 2-hop distance neighbors.

In [12], they used a notion of contribution; each forward node contributes 1 to all its 1-hop neighbors, and $1/k$ to all its 2-hop neighbors. The effective contribution of u to v is u's contribution to v before the signal energy of v reaches 1. The initial signal energy of each node is 0. A node is said to have the maximum effective contribution if it has the maximum total effective contribution to its neighbors and 2-hop neighbors. If we consider the contribution of each forward node as its dominative capability to all its neighbors, thus each forward node can fully dominate its 1-hop neighbors, and partly dominate its 2-hop neighbors. The following definitions will be used throughout the paper.

Definition 1. For any vertex uV, $v\Gamma_2(u), v$ is a dominator neighbor of u if v is a cluster-head (or dominator).

Definition 2. For a vertex v, the 2-hop-independent neighbors of v are $P_2 \subseteq \{\Gamma_2(v) - \Gamma(v)\}$, such that if $v_1, v_2 \in P_2$, then v_1 and v_2 are independent.

Definition 3. Let vertex w be called as connector if it is common neighbor between dominators u and v, where v is the 2-hop neighbor of u.

2-hop WCDS is also a CDS. It requires that, for any two nodes with distance equal to 2, there exists at least one shortest path between them, whose intermediate node should be included in 2-hop WCDS. The formal definition is shown in details as follows.

Definition 4. The 2-hop shortest path weakly connected dominating set problem (2-hop WCDS) is to find a minimum-size node set $S \subseteq V$ such that

(1) $\forall u \in V \setminus S$, $\exists v \in$ S such that $(u, v) \in E$,

(2) the induced graph $G[S]$ is connected, and

(3) $\forall u, v \in V$, if $d(u, v) > 2$, then $\exists p_i\ (u, v) \in p(u, v)$, $p_i(u, v) \setminus \{u, v\} \subseteq S$.

We do not consider the situation of $d(u, v) = 1$. The reason is that our WCDS aims to reduce transmission cost. When we select a WCDS, neighbors of $\forall v \in V$ must be known to v during selecting process. As a result, when v has a packet destined to u, v will not inform adjacent nodes in WCDS to help deliver the packet, because v knows that u can receive packets from v directly and no consecutive forwarding will happen. However, once $d(u, v) > 1$, consecutive forwardings are needed to deliver packages to the destination node. Thus, a good selection of forwarding nodes will influence on network performance greatly. We hope to select a CDS with minimum size, but keep the value of $d(u, v)$, $\forall u, v \in V$ through this CDS the same as that in original graph. It is the goal of WCDS. We redefine a node's degree in details as follows.

Definition 5. The degree of a node u is denoted by $d(u)$. Define the rank of node u to be an ordered pair (d_u, id_u), where d_u is the node degree and id_u is the node ID of u. We say that a node u with rank (d_u, id_u) has a higher order than a node v with rank (d_v, id_v) if $d_u > d_v$, or $d_u = d_v$ and $id_u < id_v$.

Definition 6. The "diameter" X of a set of nodes S in a graph G is the maximum of the pairwise shortest paths between these nodes $X = \max_{i,j \in S} d(i, j)$, where $d(i, j)$ is the shortest number of hops needed to go from node i to node j in G.

When WCDS is constructed, only nodes in WCDS may forward data. In broadcasting [16], nodes in WCDS can help spread data to the whole network. In routing, data will be sent to WCDS and be delivered via nodes in WCDS. Thus, how to construct a WCDS is closely related to the performance of WCDS-based broadcasting and routing. Our approach to establishing a minimal WCDS is based on two phases that implement the data aggregation tree (DAT) and WCDS elections, respectively. We discuss the construction of WCDS in the following sections.

3. Algorithm Description

The aim of the proposed algorithm is to construct a minimum WCDS as a clustering scheme for WSN. We employ a CDS in this paper since it can behave as the virtual backbone of a sensor network. Our clustering schemes construction algorithm includes two phases: DAT construction and then to select connectors to make the MIS nodes connected into a WCDS construction. In the first phase, we construct a maximal DAT of the network. The second phase of the algorithm is to choose the nodes (called connectors) to make the WCDS connected.

3.1. Construction of Data Aggregation Tree. We assume that each node knows the node ID and degree of all its 1-hop neighbors and 2-hop neighbors, this can be achieved through requiring each node to broadcast its node ID initially. After each node knows all its neighbors, it can broadcast its degree, one more round of "Hello" message is needed to construct 2-hop information.

Let the target region be A, and sensor node set in the region be

$$S = \{s_i(x_i, y_i) \mid s_i \in A\}, \tag{3}$$

where (x_i, y_i) is the position coordinate of the node s_i, external of the target region is set $K = \{k_i(x_i, y_i) \mid k_i \notin A\}$, and DAT is definite as T has

$$T(s_i - > K) = \bigcup_i \text{path}(s_i - > k_i), \quad k_i \in K, \tag{4}$$

where path $(s_i - > k_i)$ is the greatest span path from node s_i to node k_i in graph G, and its length is diameter l. In this path, the minimum distance between each node is bigger or equals to the minimum distance $d(s_i, k_i)$ in any other path from s_i to k_i, and the node number is the smallest in graph G.

Dynamic topology has a significant impact on DAT algorithms. Two actions of a node lead to network topology changes: withdrawing and joining. Withdrawing refers to the functional termination of a node in the network, and it happens when a node fails, runs out of power, or exits from the network. Joining refers to the functional start of a node in the network, and it happens when a new node is added, or a node recovers from a failure. Moving of a node can be treated as two separate actions of withdrawing and joining if the node can be assumed to stop receiving and transmitting messages when in motion. To cover a broader range of situations, this paper assumes no special notification sent from the withdrawing or joining node. Relying on such notification, even if possible, imposes high expectation on the ability of nodes. The neighbors of a changing node must rely on other mechanisms to detect the changes.

The changing neighborhood resulted from a node withdrawing or joining affects the generated DAT. Generally, there are two methods to handle it: recalculating and updating. With the recalculating method, a distributed DAT algorithm starts at a fixed interval or is triggered by some event (e.g., when disconnection of the dominating set is detected), and a new DAT is generated from scratch. With the updating method, the DAT is maintained by updating a portion of the existing dominating set according to the topology changes. A practical strategy may use the updating method most of the time and use the recalculating method

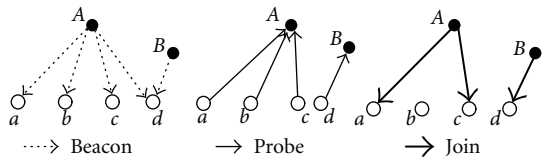


Figure 3: Exchange of message to construct the DAT.

when necessary. This paper only discusses the updating method.

Let depth of the tree T be p. Algorithm 1 constructs a DAT with given depth. When a node chooses its two children, it will choose the two biggest span nodes, ensuring that the tree T covers more target regions as far as possible. In the process of the multiple regressions, it can achieve the high accuracy. After the DAT is formed, in each subdomain all residual nodes send data to the nearest tree node away from themselves. In this paper, we used the literature [19] design method to construct the aggregation tree. That is, constructing process through three kinds of messages: Beacon, Probe, and Join. Figure 3 describes the process about the exchange of different signals to construct the tree. For more details, see literature [19].

After given data aggregation tree (DAT), a data communication operation consists of (possibly repeated) two phases: a propagation phase where the query demands are pushed down into the sensor network along the tree, and an transmission phase where the aggregated values are propagated up from the children to their parents.

3.2. Clustering Formation. In this section, a data aggregation tree-based algorithm (called DAT-WCDS) is proposed for clustering formation in WSN, which focuses on finding a WCDS problem in the network graph. In the algorithm, a special dominating set using a MIS of the network is constructed, and then a CDS is constructed to connect dominators and the other nodes.

Given T be a DAT and D is a dominating set of T containing. It suffices to determine an independent set J of vertices which is disjoint from D and contains a neighbor of every vertex in D, because a maximal independent set I which contains J but is disjoint from D is clearly a dominating set of T. A simple strategy to select the elements of J is to root T in some vertex x in D and to select a child of every vertex in D which itself is not contained in D.

If this strategy succeeds, then the selected vertices will clearly form an independent set. Nevertheless, this strategy fails in the presence of vertices u in D all children of which are also in D. For such a vertex, we have to choose its parent. Working out the consequences of this reasoning leads to Algorithm 1 in the following sections.

We will hope that there are some dominator(s) and some dominatee(s) in maximal independent set of each layer of DAT. Here a connector node x (a dominatee of a dominator u) is said to be redundant for the dominator u if removing x will not disconnect any of the 2-hop dominators of u from u. For every dominatee, it has at least one-dominator neighbor in the same or upper level. Thus, every dominator (except the root) has at least one dominator in the upper level within 2 hops. Using this property, we can ensure that all the data in the dominators can reach the root finally if every dominator transmits its data to some dominator in upper level within two hops. From another point of view, considering dominators in the decreasing order of their levels, a dominator u in level L aggregates data from all dominators in level $L+1$ or $L+2$ that are within two hops of u leads to Algorithm 2 in the following sections.

In Algorithm 2, we only concentrate on communications between dominators. Since dominators cannot communicate directly, we have to rely on some dominates (NT node), each of which acts as a bridge between two dominators. The algorithm runs from lower level to upper level in DAT, every dominator will remain silent until the level where it locates begins running. When it is its turn, the dominator will try to gather all the data from other dominators in lower levels that have not been aggregated. If a dominator's data have been collected before, then it is unnecessary to be collected again. After the end of the second phase, the algorithm has identified MIS and the connectors. Iteratively, the dominator nodes are picked which connects independent set nodes in different components. The following phases are performed to establish and form clusters

Initially, the sink creates an empty cluster associated with an unclustered node of S. Each sensor $\{s_1, s_2, \ldots, s_N\}$ transmits its position (x_i, y_i) to the sink. To accomplish this step any efficient sensor routing algorithm can be used. Thus, the clustering algorithm is not bound to how the sink receives this information. If there is an unconnected node in the network, it cannot announce itself and thus will not be considered in the algorithm. Then, the sink finds the qualified unclustered nodes for joining to that first member. When no more nodes can be added to the cluster, the sink takes a new unclustered node and begins a new cluster. Then, each first member sends a packet to the members of his cluster notifying them about the cluster which they belong to. Each node is in one of the four states: unmarked, clustered-head (CH), cluster member (CM), and half-dominated. In the following, we describe the algorithm in detail.

Algorithm 3 is executed by the sink once upon deployment, and thus all nodes will become clustered. If a node joins to the network, it has to send its position (x_i, y_i) to the sink for announcing itself as a new node. The sink computes the highest rank of the new node and finds the first cluster that can accept it as a new member. Then, the sink sends a message to the first member in order that this node reorganizes the cluster with the new member. On the other hand, each node periodically sends a Hello message to the first member notifying that it is alive.

When a node dies, the first member will notify the rest of the members about the new cluster set and will reconfigure any parameter related to the cluster. The first member also periodically notifies to its cluster members about its availability. If a first-member dies, the cluster members will notify to the sink their availability to belong to another cluster or to create a new cluster. Note that the beaconing among cluster

Input: a data aggregation tree T and threshold K, P
/* W be vertex set in tree T, P be depth of the tree T, K be stage number */
Output: A maximal independent dominating set
Let D denote the dominating set constructed at stage K and be initially set to null
(1) begin
(2) $D \leftarrow \Phi$
(3) *While* $W \neq \Phi$
(4) Choose a vertex $i \in W$
(5) $D = D \cup \{i\}$
(6) $W = W \setminus \Gamma(i)$
(7) End while
(8) Choose a vertex $x \in D$ of degree $d(x) = \min\{d(u) \mid u \in D\}$;
(9) $J \leftarrow \Phi$ /* J be an independent set */
(10) while $\exists u \in D$ such that $u \notin \Gamma(J)$ and all children of u inclusion in $D \cap (\Gamma(J) \times J)$ do
(11) Let v be the parent of u;
(12) $J \leftarrow J \cup \{v\}$;
(13) partner$(u) \leftarrow v$;
(14) end
(15) while $\exists u \in D$ such that $u \notin \Gamma(J)$ do
(16) Choose a child v of u such that $v \notin D \cup \Gamma(J)$;
(17) $J \leftarrow J \cup \{v\}$;
(18) end
(19) Let I be a maximal independent set of T with $J \subseteq I$ and $D \cap I = \Phi$;
(20) Increment stage number k
(21) Until depth of the tree T is greater than P or the stage number k is threshold K
(22) end

Algorithm 1

Input: The DAT tree with root v_0 and depth P, data d_i stored at each tree node v_i.
(1) Let T be the final data aggregation tree.
(2) Initially all independent set nodes form different components, each node in I broadcasts dominatees message so that dominatees can know of adjacent independent set nodes in different components.
(3) for $i = P - 1,\ P - 2, \ldots,\ 0$ do
(4) while a dominatee node v exists having i-adjacent independent nodes of I in different components do
(5) Choose all dominators, denoted as B_i, in level i of T tree.
(6) For every dominator $u_i \in B_i$ do
(7) Node u_i broadcasts itself as the dominator.
(8) Node u_i finds the set $\Gamma_2(u_i)$ of unmarked dominators that are within 2-hops of u in T, and in lower level $i + 1$ or $i + 2$, mark all nodes in $\Gamma_2(u_i)$.
(9) Dominatees w_i on receiving this message keep a count of neighbouring dominators at level $i + 1$ or $i + 2$ and broadcasts the final count.
(10) Each level $i + 1$ or $i + 2$ dominators on receiving the counts from the potential connectors, select among them the node with highest bank as its connector and informs it.
(11) Node w_i, then becomes a connector; $B_i \leftarrow B_i \cup \{w_i\}$.
(12) Every node w in $\Gamma_2(u_i)$ sends aggregated data to the parent node (a connector node) in T.
(13) Every node z that is a parent of some nodes in $\Gamma_2(u_i)$ sends original data to node u_i (which is the parent of z in T).
(14) End for
(15) $i = i - 1$
(16) End for /* The identified DAT nodes connect the dominator nodes. Thus, independent set nodes and DAT nodes forms the CDS of G */
(17) The root v_0 sends the result to the sink using the shortest path.

Algorithm 2

Input: The DAT tree with has identified I and the connectors;
/* Let V_p, V_q be the cluster, $v \in V_p$ and v_p its cluster-head. $H = I \cap \Gamma_2(v_p)$ be the set of 2-hop neighbors of v_p in I; V_u be the cluster dominated by dominator node u; */
(1) begin
(2) Initially, all nodes are unmarked.
(3) u sends out CH message with the information of V_u to its neighbors, $v \in V_u$, v sends it's *rank* though CM message to dominator $u \cdot u$ receiving CM message from all its members computes its relative proximity as *rank*. u ranks each node v in V_u based on non-increasing ordering using proximity *rank*;
(4) u node with the highest *rank* among its unmarked 2-hop neighbors becomes a cluster-head and broadcasts CH messages to all its neighbors;
(5) After receiving a CH message, for a node v, v sends a message so that all available nodes in w$\in \Gamma(u)$ become its dominates;
(6) Each w node in turn sends another message to its 2-hop neighbors, making the available nodes as potential dominators;
(7) For all V_h, $v_h \in H$ and v_h is the cluster-head of V_h do
(8) v becomes a cluster-member if it is a 1-hop neighbor of node u, and its current state is unmarked. If it is the first time that v receives a CH message, v will broadcast CM messages to all its neighbors;
(9) If v is a 2-hop neighbor of node u, its current state is unmarked, and it is the first time that v receives a CH message, v becomes half-dominated;
(10) If node v is a 2-hop neighbor of node u and its current state is half-dominated, v becomes a cluster-member if v receives a different CH message for the second time. V will broadcast CM messages to all its neighbors;
(11) If node v is a 2-hop neighbor of node u and its current state is half-dominated, v becomes a cluster-head if v does not receive a different CH message again. v will broadcast CH messages to all its neighbors;
(12) Dominator node v_p sends CH message to its neighboring dominator; v_q receiving CH message switched to become dominator in DAT tree and sends out CM message to the dominator node;
(13) Connectors $c_k \in V_q$ among its neighboring nodes are activated on receiving CM message and sends out CM message to its independent dominators;
(14) Switching from v_p to v_q takes place through local messages;
(15) The same procedure is repeated among the remaining nodes, until each node in DAT becomes either a cluster-head or cluster-member;
(16) End for
(17) End

Algorithm 3

members implies low overhead since cluster sizes have few nodes.

4. Analysis of Algorithm

In the next subsections we first analyze the correctness of the algorithm and then analyze its complexity for running time and messages exchanged of the algorithm.

4.1. Correctness of the Algorithm

Theorem 7. *The output of the proposed Algorithm 1 is a maximal independent set.*

Proof. By contradiction, we consider the first execution of the while-loop in line 11 for which the vertex u has no parent which does not belong to D; that is, either u is the root x of T or the parent of u belongs to D.

Let D' denote the set of vertices u' from D which can be reached from u on a path P of the form

$$P: \; u_0 w_1 v_1 u_1 w_2 v_2 u_2 \ldots \cdot w_1 v_1 u_1 \tag{5}$$

with $u_0 = u$, $u_l = u'$, $l \in N$, $w_i \notin D$, and partner $(u_i) = v_i$ for $1 \leq i \leq l$. Note that w_1 is a child of u. Let the set D'' contain the parent of the parent of u'—the grandparent of u' for every vertex u' in D'. Let $\check{D} = (D \setminus D' \cup \{u\}) \cup D''$.

Let w'' be a child of u. Clearly, $w'' \notin J$. If $w'' \in D$, then $w'' \in \check{D}$. If $w'' \notin D$, then w'' has a child v'' which belongs to J, and v'' has a child u'' which belongs to D such that partner $(u'') = v''$. Since $uw''v''u''$ is a path as in (1), we obtain, by the definition of D', that $u'' \in D'$. This implies that $w'' \in D''$, and hence $w'' \in \check{D}$. Therefore, in both cases, $u, w'' \in \Gamma[\check{D}]$, and all vertices which were dominated by u in D are still dominated by vertices in $\check{D}$.

Let $u' \in D'$. Let P be as in (1) with $u' = u_l$. Since $w_l \in \check{D}$, we have $v_l \in \Gamma[\check{D}]$. If w'' is a child of u', then exactly the same argument as above implies that $w'' \in \check{D}$. Hence again all vertices which were dominated by u'' in D are still dominated by vertices in $\check{D}$.

Altogether, we obtain that $\check{D}$ is a dominating set of T which contradicts the assumption that D is a minimum dominating set. By the claim, the while-loop in line 11 successfully adds to the set J the parents of vertices in D which do not belong to D. By the condition for the while-loop in line 11, just before the execution of the while-loop in line 16, the set J is independent, and every vertex $u \in D$ with $u \notin \Gamma(J)$ has at least one child which does not belong to D

and is nonadjacent to the vertices in J. During the executions of the while-loop in line 16, only children of vertices in D are added to J, and this property is maintained throughout the remaining execution of Select. Hence, the while-loop in line 16 successfully adds to the set J the children of vertices in D which do not belong to D such that after the last execution of the while-loop in line 16, the set J is independent, disjoint from D and $D \subseteq \Gamma(J)$. By the above remarks, the set I defined in line 20 is an independent dominating set of T which completes the proof. □

Theorem 8. *After the above two phases, the constructed DS is a WCDS of the whole graph.*

Proof. After the first phase, Algorithm 1 constructs a DAT with the given depth. When a node chooses its two children, it will choose the two biggest span nodes, ensuring that the tree T covers more target regions as far as possible. It is possible that there exist two dominators that are apart by at least 2 hops in the graph. However, these dominators are apart by at most 3 hops. According to the definition of WCDS, we know that the IDS constructed in the second phase is a WCDS. Although the second phase reduces the size of dominators, the connectivity is not destroyed. Therefore, after the two phases, the constructed IDS is a WCDS of the whole graph. □

4.2. Complexity Analysis

Theorem 9. *The algorithm DAT-WCDS has time complexity $O(n)$ time and $O(D)$ rounds, where D is the network diameter and message complexity of $O(n \times d^2)$, where d is max degree of node in G.*

Proof. Assume that in a given unit disk the size of an MIS is always less than maximum degree of a node in G; therefore, $|\text{MIS}| \leq d$. Each node sends at most two messages to become dominatee and at most d messages per degree to update neighbor's information and d^2 to get neighbors of the neighbor to become dominator. Thus, message complexity is $O(n \times d^2)$, where d is the maximum node degree.

While establishing the relationship between connectors and dominators, the message complexity is only size of CDS which is at most $O(n)$. Thus, in the message complexity of algorithm $O(n \times d^2)$, each node is explored one by one, so the time complexity $O(n)$. The number of synchronous rounds is $O(D)$, where D is network diameter, which is bounded by shortest distance of farthest node from a given leader. □

5. Simulation and Discussion

In simulations, all algorithms in discussion are implemented by using MATLAB, and all nodes are randomly deployed in a square area A. Every node uses a radio range $r(r = 10, 60$ units). The network size and node density determine the number of nodes (N) in the network. Node density p is defined as the average number of nodes per unit area. Relative node density is defined as the number of neighbors per node. For example, given node density $p = 0.01$, $r = 10$, the relative node density is $\pi \times r^2 \times p = 3.14$. Table 1 summarizes all the network configurations used in simulations.

Table 1: Values of simulation parameters used.

Parameter	Value
Size of area A	$160 \times 160\,\text{m}^2$
Transmission range r	$r \in [10, 60]$ m
Number of nodes N	$N \in [20, 160]$
Density of network p	$p = A/N$
Event area A_s	$40 \times 40\,\text{m}^2$
Size of messages m	500 b

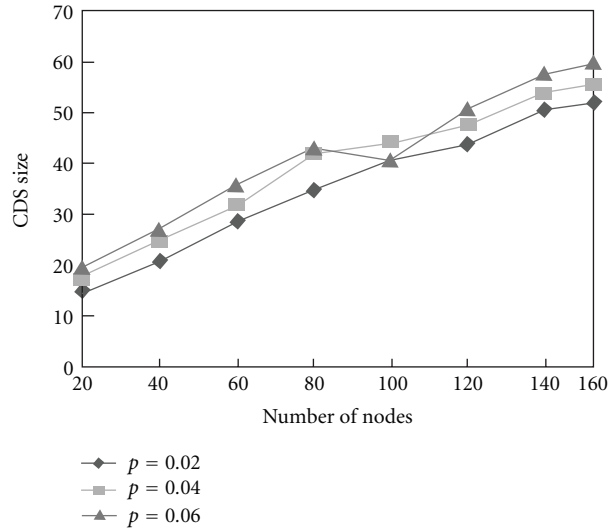


Figure 4: Impact of node density on CDS size.

5.1. Node Density. Node density determines how many neighbors a node can have. With a higher node density, a node has more neighbors to compete with to become a dominator. But after a node becomes a dominator, all its neighbors are covered as NT nodes. Usually, a node that can cover more neighbor nodes has a greater chance to become a dominator because of its greater degree. Thus, a new dominator will try to cover a new area of the network by given a connected network. Therefore, if the algorithm is well designed, the CDS size should be mainly determined by the network size and has less to do with the node density. Figures 4 and 5 show that DAT-WCDS generates CDS of almost the same size and the same diameter in networks with various node densities. But it takes longer time for the algorithm to converge in high-density networks (Figure 6).

5.2. Size of WCDS. Figure 4 shows the results when the node's transmission range is set as 30 units and the number of nodes in the networks ranges from 20 to 160. When the transmission range increases, as more nodes may be connected, the network becomes denser. In this case, the size of WCDS only increases slightly as the size of the network

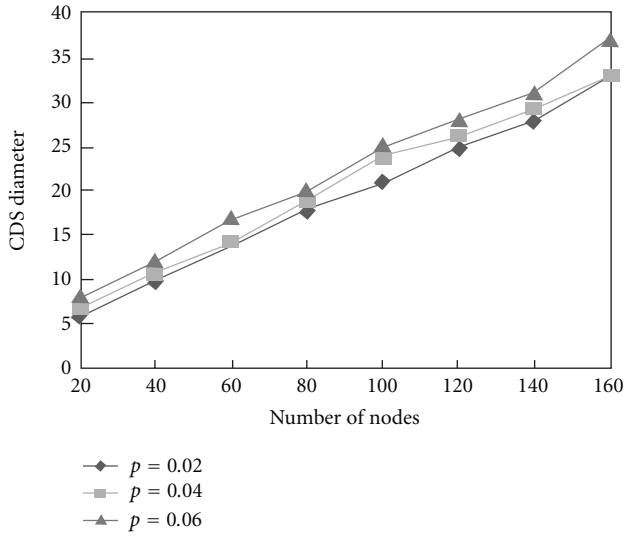


Figure 5: Impact of node density on CDS diameter.

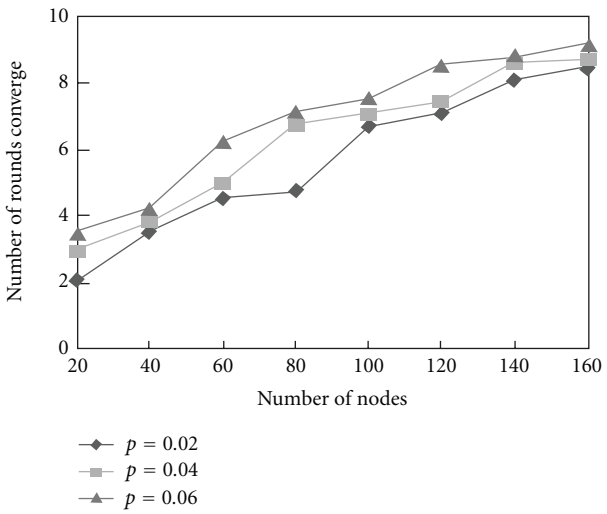


Figure 6: Impact of node density on convergence time.

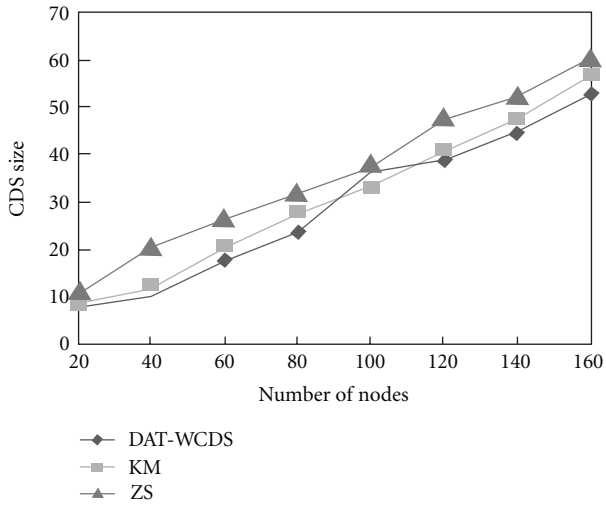


Figure 7: Comparison with other algorithms.

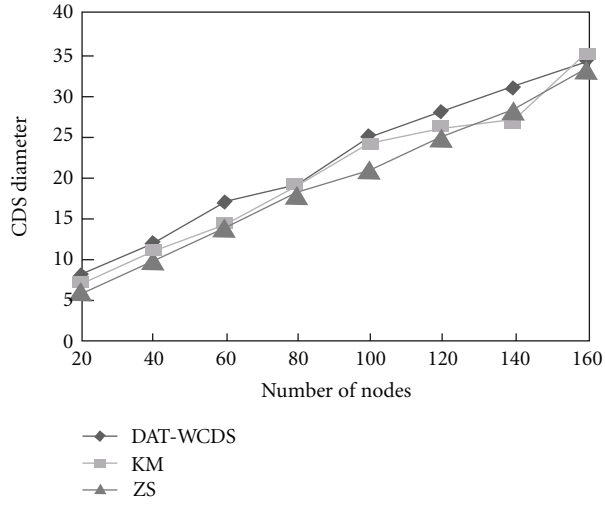


Figure 8: Comparison with other algorithms.

increases. When the number of nodes in the network reaches 160, the number of nodes in the WCDS constructed by the DAT-WCDS algorithm is only about 31% of that constructed. The reason why our algorithm always outperforms is that for each pair of 2 hops is away cluster-heads adds one additional node to the WCDS, whereas our algorithm only "weakly connects" 2 hops away cluster-heads in different areas. We find that increasing the node's transmission range can increase the coverage area of each node, and therefore, increasing the density of the network, which leads to a smaller size of the WCDS.

5.3. Comparison with Other Algorithms. The DAT-WCDS algorithm is compared with two multiple-phase CDS algorithms: ZS [20] and KM [21]. However, KM does not generate smallest size CDS, but it converges fast. Therefore, KM here serves as a good comparison candidate as we will show various aspects of algorithms at different performance levels.

Figure 7 shows that, in terms of CDS size, DAT-WCDS performs better than KM and ZS. The connected dominating sets built by KM have smaller diameters in large networks (Figure 8), but the tradeoff is much greater dominator population. DAT-WCDS converges much faster than ZS, as illustrated in Figure 9. The DAT-WCDS algorithm always converges in no more than 11 rounds for a wide range of network sizes in our simulations. Here, each round of ZS is the time for generating a new layer of dominators. The convergence time of ZS is mainly affected by the network size and the node radio range.

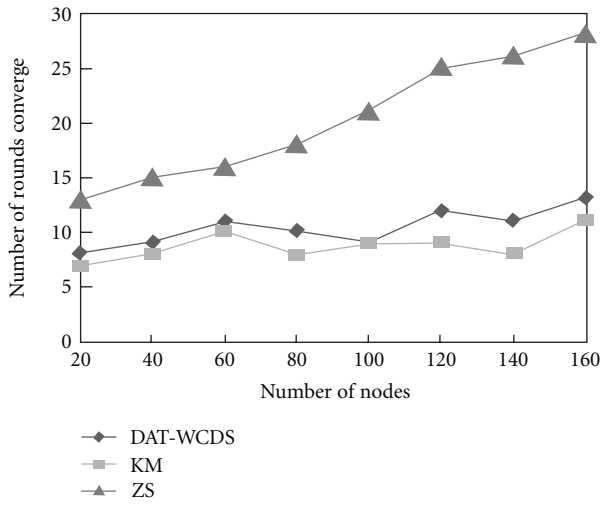


Figure 9: Comparison with other algorithms.

6. Conclusion

In this paper, we extend the dominative capabilities of nodes, and a data aggregation tree-based algorithm called DAT-WCDS is proposed for clustering formation in WSN, which focuses on finding a WCDS problem in the network graph. Our clustering schemes construction algorithm includes two phases: DAT is constructed and a special dominating set using a MIS of the network is constructed, then selecting connectors to make the MIS nodes connected into a WCDS construction. The correctness and performance of our algorithms are confirmed through theoretical analysis and comprehensive simulations.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants nos. 61070169 and 61170021, Natural Science Foundation of Jiangsu Province under Grant no. BK2011376 Specialized Research Foundation for the Doctoral Program of Higher Education of China no. 20103201110018, and Application Foundation Research of Suzhou of China no. SYG201118 and sponsored by Qing Lan Project.

References

[1] M. Agarwal, J. H. Cho, L. Gao, and J. Wu, "Energy efficient broadcast in wireless ad hoc networks with hitch-hiking," in *Proceeding of the 23rd AnnualJoint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2096–2107, Hong Kong, March 2004.

[2] R. Rajaraman, "Topology control and routing in ad hoc networks: a survey," *SIGACT News*, vol. 33, no. 2, pp. 60–73, 2002.

[3] J. Wu, "Extended dominating-set-based routing in ad hoc wireless networks with unidirectional links," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 866–881, 2002.

[4] K. M. Alzoubi, P.-J. Wan, and O. Frieder, "Message-optimal connected dominating sets in mobile ad hoc networks," in *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '02)*, pp. 157–164, Lausanne, Switzerland, June 2002.

[5] K. M. Alzoubi, W. Peng-Jun, and O. Frieder, "Distributed heuristics for connected dominating sets in wireless ad hoc networks," *Journal of Communications and Networks*, vol. 4, no. 1, pp. 22–29, 2002.

[6] L. Ding, W. Wu, J. Willson, H. Du, W. Lee, and D.-Z. Du, "Efficient algorithms for topology control problem with routing cost constraints in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1601–1609, 2011.

[7] Y. Liang and V. V. Veeravalli, "Cooperative relay broadcast channels," *IEEE Transactions on Information Theory*, vol. 53, no. 3, pp. 900–928, 2007.

[8] N. Jindal, U. Mitra, and A. Goldsmith, "Capacity of ad-hoc networks with node cooperation," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '04)*, pp. 271–272, July 2004.

[9] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[10] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Communications Magazine*, vol. 42, no. 10, pp. 74–80, 2004.

[11] D. N. C. Tse, P. Viswanath, and L. Zheng, "Diversity-multiplexing tradeoff in multiple-access channels," *IEEE Transactions on Information Theory*, vol. 50, no. 9, pp. 1859–1874, 2004.

[12] J. Wu, M. Cardei, F. Dai, and S. Yang, "Extended dominating set and its applications in ad hoc networks using cooperative communication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 851–864, 2006.

[13] Y. P. Chen and A. L. Liestman, "Maintaining weakly-connected dominating sets for clustering ad hoc networks," *Ad Hoc Networks*, vol. 3, no. 5, pp. 629–642, 2005.

[14] Y. P. Chen and A. L. Liestman, "A zonal algorithm for clustering ad hoc networks," *International Journal of Foundations of Computer Science*, vol. 14, no. 2, pp. 305–322, 2003.

[15] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and Systems*, vol. 5, no. 1, pp. 66–77, 1983.

[16] B. Han and W. Jia, "Clustering wireless ad hoc networks with weakly connected dominating set," *Journal of Parallel and Distributed Computing*, vol. 67, no. 6, pp. 727–737, 2007.

[17] S. Basagni, M. Mastrogiovanni, and C. Petrioli, "A performance comparison of protocols for clustering and backbone formation in large scale ad hoc networks," in *Proceeding of the IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MAHSS '04)*, pp. 70–79, October 2004.

[18] K. K. Chintalapudi and R. Govindan, "Localized edge detection in sensor fields ," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, pp. 59–70, May 2003.

[19] S.-K. Zhang, Z.-M. Cui, S.-R. Gong, Q. Liu, and J.-X. Fan, "A data aggregation algorithm based on splay tree for wireless sensor networks," *Journal of Computers*, vol. 5, no. 4, pp. 492–499, 2010.

[20] D. Zhou, M.-T. Sun, and T.-H. Lai, "A timer-based protocol for connected dominating set construction in IEEE 802.11 multihop mobile ad hoc networks," in *Proceeding of the 5th Symposium on Applications and the Internet (SAINT '05)*, pp. 2–8, February 2005.

[21] K. M. Alzoubi, P.-J. Wan, and O. Frieder, "Maximal independent set, weakly connected dominating set, and induced spanners for mobile ad hoc networks," *International Journal of Foundations of Computer Science*, vol. 14, no. 2, pp. 287–303, 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 182561, 9 pages
doi:10.1155/2012/182561

Research Article

Cooperative Data Processing Algorithm Based on Mobile Agent in Wireless Sensor Networks

Shukui Zhang,[1,2] Yong Sun,[1] Jianxi Fan,[1] and He Huang[1]

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

Correspondence should be addressed to Shukui Zhang, zhangsk2000@163.com

Received 7 January 2012; Accepted 25 March 2012

Academic Editor: Hongli Xu

Mobile agent (MA) systems provide new capabilities for energy-efficient data processing by flexibly planning its itinerary for facilitating agent-based data collection and aggregation. In this paper, we present a cooperative data processing algorithm based on mobile agent (MA-CDP), and considers MA in multihop environments and can autonomously clone and migrate themselves in response to environmental changes. MA accounts for performing data processing and making data aggregation decisions at nodes rather than bringing data back to a central processor, and redundant sensory data will be eliminated. The results of our simulation show that MA-based cooperative data processing provides better performance than directed diffusion in terms of end-to-end delivery latency, packet delivery ratio, and energy consumption.

1. Introduction

The advances in Microelectromechanical System (MEMS) and wireless communication have enabled the development of a new kind of network—the wireless sensor network (WSN). One of the unique features of WSN applications is the necessity of cooperation. Each sensor node normally has limited sensing and processing capabilities, constrained power resources, and reduced communication bandwidth. Therefore, cooperation among sensor nodes is important in order to compensate for each other's capabilities as well as to improve the degree of fault tolerance, and the key to an effective cooperation is a combination of low-level sensor processing and local exchange of data to reach consensus in the neighborhood of the occurring event. This characteristic of WSNs brings up some important issues for cooperation communication, including energy efficiency, scalability, and reliability [1].

To address such challenges, most of researches focus on prolonging the network lifetime, allowing scalability for a large number of sensor nodes, or supporting fault tolerance (e.g., sensor's failure and battery depletion) [2, 3]. Most energy-efficient proposals are based on the traditional client/server computing model, where each sensor node sends its sensory data to a processing center or a sink node. Because the link bandwidth of a wireless sensor network is typically much lower than that of a wired network, a sensor network's data traffic may exceed the network capacity. To solve the problem of the overwhelming data traffic, Qi et al. [2] proposed the mobile-agent-(MA-) based distributed sensor network (MADSN) for collaborative signal and information processing.

Generally speaking, an MA is a special kind of software that can execute autonomously, with identification, itinerary, data space, and method as its attributes. An MA [4] is a computational process which has several characteristics: (1) "reactivity" (allowing agents to perceive and respond to a changing environment), (2) "social ability" (by which agents interact with other agents), and (3) "proactiveness" (through which agents behave in a goal-directed way). An MA may need to cooperate in order to achieve better and more accurate performance or need additional capabilities that it does not have. This cooperation takes place by doing a coalition formation which it is created by the fusion center agent. By cooperation we mean sharing data and resolving conflicts.

By transmitting the software code, namely, mobile agent (MA) to sensor nodes, the large amount of sensory data can be reduced or transformed into small data by eliminating

the redundancy. For example, the sensory data of two closely located sensors are likely to have redundant or common parts when the data of two sensors are merged. Therefore, data aggregation is a necessary function in densely populated sensor networks in order to reduce the sensory data traffic.

MA-based algorithm is a promising design paradigm that can be utilized to solve the overwhelming data traffic [5], especially over low bandwidth links, an MA selectively migrates among sensor nodes by moving the processing function to the target nodes, performs local processing by using resources available at the local nodes rather than bringing the data to a central processor (sink), and incrementally fuses the local decisions on each sensor node to reach a progressively accurate global decision.

This limitation is tackled by MADD algorithm [6]. The processes involved in MADD are divided into some sections. First, the MA is dispatched from sink to the first source node, and in the next place, the MA migrates from first source node to last source node, visiting selected source nodes in between. This algorithm does not always guarantee the best sequence of nodes to be visited.

In addition, node failures are a frequent occurrence in WSN. When a node fails, all data and MA residing on that node are permanently lost. While this may cause application failure, cooperative data processing provides the capability for applications to self-heal. Specifically, an application can heal itself by cloning or moving its agents onto the replacement node when it is installed. Unlike other in-network reprogramming systems, cooperative data processing enables application developers to control over the self-healing process. For example, in the fire tracking application, a node will fail when it catches on fire. The Fire Tracker agent heals itself by detecting this failure and cloning itself around the failed node to ensure the integrity of the perimeter.

Towards this end, we propose cooperative data processing based on MA algorithm (MA_CDP). With this algorithm, large amount of sensory data can be reduced or transformed into small data by eliminating the redundancy. Furthermore, to have better understanding in evaluating the performance of the algorithm, we present detailed analytical algorithm of data dissemination. With appropriate parameters set, the results of our simulation show that cooperative data processing provides better performance in terms of packet delivery ratio, energy consumption, and end-to-end delay.

2. Related Work

A traditional approach for WSN adaption is to reprogram it over the wireless network. Systems that enable this can be divided based on what is reprogrammed, that is, native code, interpreted code, or both. Two systems that reprogram native code are Deluge [7] and MOAP [8]. They are designed to transfer large program binaries, enable the network to be arbitrarily reprogrammed, but incur high overhead and latency. To address this, SOS [9], Contiki [10], and Impala [11] are systems that enable partial reprogramming of binary code by providing a microkernel that supports dynamically linked modules. Since modules are relatively small, the cost of reprogramming is lower.

Recently there has been a growing interest on the design, development, and deployment of MA systems for high-level inference in WSNs [12–15]. In [12], the agent design in WSNs is decomposed into four components, that is, architecture, itinerary planning, middleware system design, and agent cooperation. Among the four components, itinerary planning determines the order of source nodes to be visited during agent migration, which has a significant impact on energy performance of the MA system. It has been shown that finding an optimal itinerary is a NP-hard problem. Therefore, heuristic algorithms are generally used to compute competitive itineraries with a suboptimal performance.

In [13], two simple heuristics are proposed: (i) a local closest first scheme that searches for the next node with the shortest distance to the current node, and (ii) a global closest first scheme that searches for the next node closest to the dispatcher. These two schemes only consider the spatial distances between sensor nodes and, thus, may not be energy efficient in many cases.

Sharma and Mazumdar have investigated the use of limited infrastructure, that is, networks with a number of wired connections between sensor nodes, in [16]. Their approach establishes a small-world graph by utilizing wired links between a subset of nodes to reduce the overall energy demands as well as the different energy consumption rates of participating nodes. The additional efforts required for the wiring however make it suited for long-term deployments of sensor networks only.

Wagenknecht et al. also propose to deploy nodes with higher computational capabilities within a WSN to act as cluster-heads for sensor subnetworks, that is, partitions of the sensor network [17]. They use embedded systems with a 233 MHz clock frequency and 128 megabytes of RAM as the backbone to interconnect the sensor subnetworks through a wireless mesh network. Although deploying additional gateways allows for shorter multihop routes, the energy savings are possibly counterbalanced by the greater energy requirements of the gateways, which are not analyzed in detail in the paper.

A different approach to shift computational tasks into the network is the use of mobile agents. In such networks, data is not forwarded to an external sink, but instead, the processing application (the mobile agent), including its state variables, is sent to the node and executed locally [18]. As all process context data are contained within the agent, it can be supplied with input data at one node, while the processing can be performed at a different and more powerful system. We thus consider it a well-suited supplement to migrate tasks between nodes.

3. Collaborative Data Processing

An MA is a special process that can autonomously migrate across nodes. The migration transfers both the code and state, allowing the agent to resume execution at the destination. It is useful for performing computations that span

multiple nodes. When an agent migrates, it can either clone or move. If an agent is cloned, a copy of it arrives and starts executing at the destination while the original one resumes on the original node. If an agent is moved, it will no longer exist on the original node after it arrives at the destination. An agent's life cycle begins when it is either injected into the network from a base station or cloned from another agent already in the network. Each agent executes autonomously performing application-specific tasks, and multiple agents may reside on the same node. When an agent completes its tasks, it dies by freeing the computational resources which it used.

The order of source nodes to be visited by the MA can have a significant impact on energy consumption. Finding an optimal source-visiting sequence is an NP-complete problem [6]. In [19], a genetic algorithm-based solution to compute an approximate solution is presented. Though global optimization can be achieved by using genetic algorithm, it is not a lightweight solution for sensor nodes that are constrained in energy supply [20]. This paper adopts a gradient-based solution for the MA to dynamically decide the route.

3.1. Algorithm Overview. Figure 1 shows the sequence of operations to processes involved in WSN environment. When a sink receives a task request assigned by an application, the sink broadcasts the query packet. The query packet contains the sensing task description, interest region representation, along with other information. If the node finds it can satisfy the interest query, it declares itself as a source node. Each source node generates a response by exploratory data. Then, the sink receives a large number of exploratory data packets from various source nodes and decides the source-sequence list to be visited by MA. The MA-related operation begins at the point of the sink dispatching MA and ends when the MA returns to the sink with collected results. In most cases, each source is expected to generate the sensory data periodically with some interval, which means the same code (MA) needs to be stored for multiple running. Thus, when the MA arrives at the FirstNo, it will be stored. Then, it sets a Create-MA-Timer, which is used to trigger the next round to dispatch the MA to collect data from the relevant sources again. Obviously, the interval between the successive rounds will be equal to the sensory data generating rate which is set to the value of the Create-MA-Timer. This round will be repeated until the task is finished. A round can also be defined as the interval from the time during which an MA collects the data packet in the FirstNo to the time during which it collects the data packet in LastSrc. At the end of the last round, the task is finished.

When the MA arrives at the first source node, it continues visiting other source nodes until it reaches the last source node. Firstly, aggregated data is sent back to the sink along the reinforced path. Therefore, an MA reduces the amount of data to be transmitted via aggregating relevant data.

Consider an MA dispatched by the sink node to collect data from n source nodes. Let S_{code} be the size of the MA processing code, S_{head} the size of agent packet header, and S^0_{ma} the agent size when it is first dispatched by the sink node.

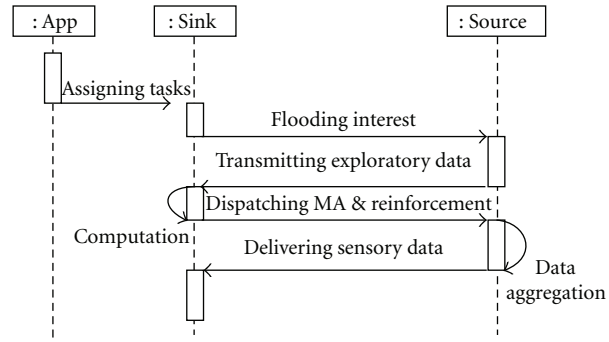


Figure 1: Routing sequence diagram.

Then we have $S^0_{\text{ma}} = S_{\text{code}} + S_{\text{head}} \cdot S_{\text{data}}$ is the size of raw data at a source node. The reduced data payload collected by the MA at each source is denoted as R_1, $R_1 = (1 - \rho) \times S_{\text{data}}$. where ρ is an aggregation ratio ($0 \leq \rho \leq 1$), a measure of the compression performance. Let S^k_{ma} be the MA size when it leaves the kth source ($1 \leq k \leq n$). Since there is no data aggregation at the first source, we have $S^1_{\text{ma}} = S^0_{\text{ma}} + R_1$.

Since the MA visits the second source node, it begins to perform data aggregation to reduce the redundancy between the data collected in the source and the data it carries. The MA size after it leaves the second source node is $S^2_{\text{ma}} = S^0_{\text{ma}} + R_1 + (1 - \rho)R_1$, and so forth. After visiting the kth source node, accumulated by MA can be calculated using the following formula (1):

$$\begin{aligned} S^k_{\text{ma}} &= S^{k-1}_{\text{ma}} + (1-\rho)R_1. \\ &= S^0_{\text{ma}} + [1 + (k-1)(1-\rho)]R_1. \end{aligned} \tag{1}$$

After visiting all the n source nodes, the MA has a size S^n_{ma} in the range $[S^0_{\text{ma}} + R_1, S^0_{\text{ma}} + n \times R_1]$. The lower bound $S^0_{\text{ma}} + S_{\text{re-data}}$ corresponds to a perfect aggregation model where multiple packets are compressed into a single one, while the upper bound $S^0_{\text{ma}} + n \times R_1$ corresponds to the case of no aggregation performed at the MA.

During the MA migration from one node to another, it aggregates sensory data and removes redundant data at the same time. The aggregated data can be calculated using the following equation:

$$S^i = \sum_{k=1}^{i} R_k. \tag{2}$$

Secondly, the energy cost is reduced. In client/server-based sensor network, all source nodes in target region transmit sensory data individually back to sink with a specific interval. In agent-based algorithm, the MA carries both the processing code as well as the source-visiting sequence.

3.2. MA Packet Format. The structure of MA packet is shown in Figure 2. The pair of SinkID and MA-SeqNum is used to identify an MA packet. Whenever a sink dispatches a new MA packet, it will increment the MA-SeqNum. FirstNo and LastNo are the source nodes scheduled to be visited firstly and lastly by the MA, respectively. The pair of FirstNo and

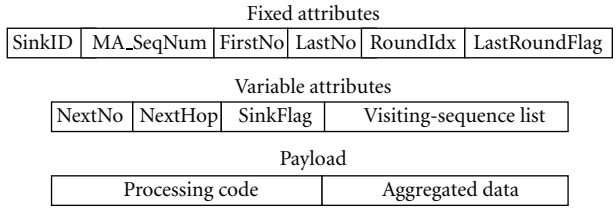


Figure 2: Structure of MA packet.

LastNo indicates the beginning and ending points of MA's data gathering. RoundIdx is the index of current round. The value is initially set to 1 by the sink in the first round and will be incremented by the FirstNo in the following rounds. Last Round Flag indicates that the current round is the last round of the whole task. The flag is set by FirstNo. When an MA with Last Round Flag set arrives at a source node, it can make the system unmount the corresponding processing code after its execution.

When an MA migrates, it may change variable attributes. NextNo specifies the next destination source node to be visited. NextHop indicates the immediate next hop node which is an intermediate sensor node or a target source node. If NextHop is equal to NextNo, it means that the next hop node is current destination source. Visiting-Sequence List contains the identifiers (IDs) of target sensor nodes that remain to be visited in the current round. It does not contain any information of source-visiting sequence since NextNo is dynamically decided when an MA arrives at a source node (except LastNo). Visiting-Sequence List initially contains all the IDs of source nodes when an MA is created. The corresponding ID will be deleted after the MA visits the source node. If all the target sources have been visited by the MA, Sink Flag is set to indicate that the destination of the MA is the sink. NextNo, NextHop, Visiting-Sequence List, and Sink Flag hint the dynamical route of MA migration. Payload includes two kinds of data. One is Processing Code which is used to process sensed data; the other is Aggregated Data which carries the accumulated data result. The size of Aggregated Data is zero when an MA is generated and increases while the MA migrates from source to source.

3.3. Cooperative MA Routing. The proposed MA_CDP mechanism is based on the original DD. In the DD, the sink initially diffuses an interest for notifications of low-rate exploratory events. Once target sources receive the corresponding interest, they send exploratory data, possibly along multiple paths, toward the sink. If the sink has multiple previous hop nodes, it chooses a preferred neighbor to receive subsequent data messages for the same interest (e.g., the one which delivered the exploratory data earliest). To do this, the sink reinforces the preferred neighbor, which in turn, reinforces its preferred previous hop node, and so on. Periodically, the source sends additional exploratory data messages to adjust gradients in the case of network changes (due to node failure, energy depletion, or mobility), temporary network partitions, or to recover from lost exploratory messages [21].

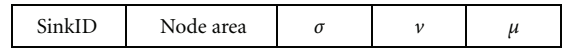


Figure 3: Instruction packet.

Suppose that the sensory data generating rate is ν, the aggregated data rate is μ, the number of transmitted aggregated data is u. To balance the energy consumption of the sensors in the sensing area, and to control the number of aggregated data sent to the sink in a distributed way, a variables σ is used, which is defined as $\sigma = u/N$. The sink calculates σ, and then sends the instruction (σ, ν, μ) back to the sensing area through the n_r relay nodes. The instruction is encapsulated with a packet header as shown in Figure 3.

We assume all sensors in the sensing area are well scheduled to wake up to receive the instruction from the sink for the next round of operation. However, the transmitted instructions are subject to packet losses in the WSN. If a sensor has not received the instruction by a predetermined time period, it sends a request through the n_r relay nodes to the sink and asks the sink to resend the instruction. Each sensor receiving the instruction creates a random number uniformly distributed in $[0, 1]$ and compares the random number with σ given in the instruction. If the random number is larger than σ, the sensor turns to sleep, otherwise it turns to sample the source signal and quantize its readings with ν bits in the next round of operation. If the random number of a sensor is larger than σ, the sensor is also selected to aggregated data with aggregated data rate μ and sends its aggregated data to the sink.

Since the ultimate goal is the detection of events in sensor networks [22], the sink may stop handling any exploratory message flows if it considers that the number of source nodes is large enough to meet the requirement of reliable event detection. Thus all the source nodes or only a subset of these nodes will be chosen to be visited by MA. Among the target source nodes to be visited, the sink will choose the first and last source nodes. Then, the sink generates an MA with the packet format described in Figure 4 and dispatches it to the first source. At the same time, the sink reinforces the path to the last source. When the MA arrives at the first source node, it is stored in the node. We divide the whole task period into rounds, where each round requires the MA to visit all the chosen target sensors and to return the data result to the sink. The MA starts from the first source (or from the sink only in the first round) and arrives at the last source. Finally, the MA will carry the data results to the sink along the reinforced path. In the first round, in addition to that the MA moves from source to source to collect and aggregate information, it also copies processing code into the memory of each source node. At the beginning of each round, the first source node will construct another MA from its memory and dispatch it to initiate the new round. Since processing code has already resided in each source node after the first round, the MA does not carry the processing code any more in the following rounds. When the whole task is finished, all the source nodes will discard the processing code.

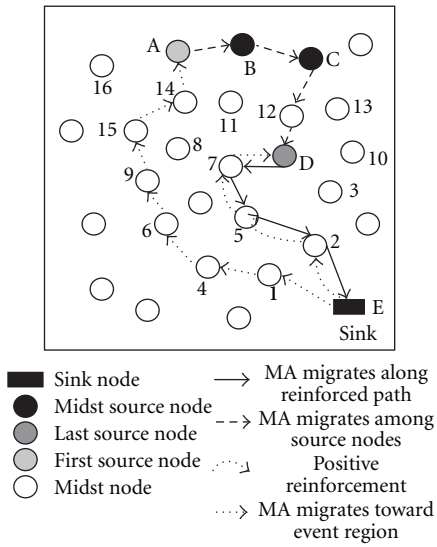


Figure 4: Cooperative MA routing.

In MA_CDP, target source nodes flooding exploratory messages enable sensor nodes to set up ToSourceEntry, which is a kind of gradient toward each target source. ToSourceEntry is used for MA to roam among source nodes. In this paper, a time-to-live (TTL) field is set in exploratory message to mandate only the sensor nodes within the target region to set up their ToSourceEntries. The value of TTL is decreased as exploratory message is propagated hop by hop. If the value is equal to 0, sensor nodes do not set up ToSourceEntry anymore. Among all the neighbors of a sensor node, only the neighbor who first relays the exploratory message of a specific target source will be chosen as the sensor node's NextHop in the ToSourceEntry. In Figure 4, nodes A, B, C, and D are the target source nodes. The ToSourceEntries set up by nodes A, B, C, 12, and D are shown in Table 1.

Based on the gradients and ToSourceEntries, a migrating route is decided by the following three operating elements.

(1) Choose *FirstNo* and *LastSrc*. According to (1), the size of an MA is the minimum in *FirstNo* while it becomes the maximum in *LastSrc*. Thus, to reduce total communication overhead, *FirstNo* should be the farthest target sensor from the sink, while *LastNo* should be the closest one. In this paper, the target source which is the last (first) to send exploratory messages to the sink is chosen as *FirstNo*(*LastSrc*). The sink will reinforce the path to *LastSrc*.

(2) Decide Source-Visiting Sequence. Except that *FirstNo* and *LastNo* are chosen by the sink, the sequence of visiting the other source nodes is dynamically decided by each target sensor in *Visiting-Sequence List*. For example, when an MA arrives at node A in Figure 4, the node will choose the closest next source node based on its ToSourceEntry shown in the first row of Table 1. Since the lowest latency of node B is the least, it implies that node B is the closest source node from node A and is chosen as *NextNo*.

(3) Find the next hop node to route an MA along the entire path from sink to source, source to source, and source to sink. Dispatched by the sink, an MA migrates to *FirstNo* in the same manner as a reinforcement message is forwarded in original DD. When the MA migrates among target sources, its next hop node will be decided according to current node's ToSourceEntry. The MA will return to the sink using the reinforced path (e.g., path D-7-5-2-E in Figure 2).

4. Performance Analysis

There are n nodes $S_k(1 \le k \le n)$ in the network. Since the transmission of instruction and request packets are to ensure all the sensors in the sensing area receive the instruction, the average number of instruction packets sent by the sink in an operation round is determined by the following analysis.

Without loss of generality, we assume that m among n_s aggregated data are received by the sink. We assume the packet loss rate per hop in the wireless channels is uniformly distributed with mean of φ. The probability of receiving v quantized readings is

$$p_m = C_{n_s}^{m}\left(1 - p_\rho\right)^m p_\rho^{n_s - m}, \tag{3}$$

where p_ρ denotes the end-to-end packet loss rate from the sensors in the sensing area to the sink and is

$$p_\rho = 1 - (1 - \varphi)^{n_s + 1}. \tag{4}$$

The probability that after i multicasts the instruction packet sent from the 1st relay node has been successfully received by all the N nodes in the sensing area is given by $(1 - \varphi^i)^N$. The probability that the instruction packet is successfully received by all the N nodes after exactly i multicasts is then given by

$$p_i = \left(1 - \varphi^i\right)^N - \left(1 - \varphi^{i-1}\right)^N. \tag{5}$$

Suppose after the sink sends exactly n $(n \geqslant i)$ instruction packets, all the N nodes successfully receive the instruction. This means the last transmission of instruction packet from the sink (through n_r hops) to the 1st relay node is successful and the nodes in the sensing area which are waiting for the instruction finally receive it. The probability that i transmissions (including the last successful transmission) among the n transmissions to the 1st relay node are successful is therefore given by

$$\begin{aligned}\varphi(n,i) &= C_{n-1}^{i-1}\left[1 - (1-\varphi)^{n_r}\right]^{n-i} \times \left[(1-\varphi)^{n_r}\right]^{i-1}(1-\varphi)^{n_r} \\ &= C_{n-1}^{i-1}\left[1 - (1-\varphi)^{n_r}\right]^{n-i}(1-\varphi)^{n_r}.\end{aligned} \tag{6}$$

Table 1: ToSourceEntry setup after exploratory messages flooding.

ToSourceEntry (SeqNum = 5)								
	A		B		C		D	
	Next hop	Cast (ms)	Next hop	Cast (ms)	Next hop	Cast (ms)	Next hop	Cast (ms)
A	—	—	B	4.46	B	8.24	B	16.32
B	A	4.47	—	—	C	4.43	C	12.89
C	B	8.16	B	4.32	—	—	12	8.52
12	C	9.65	C	7.56	C	4.86	D	5.08
D	12	14.15	12	12.67	12	8.73	—	—

Considering the joint distribution of p_i and $\varphi(n, i)$, the probability that all the N nodes receive the instruction after the sink sends exactly n instruction packets is given by

$$\begin{aligned}\varphi(n) &= \sum_{i=1}^{n}\varphi(n,i)\times p_i \\ &= \sum_{i=1}^{n} C_{n-1}^{i-1}\left[1-(1-\varphi)^{n_r}\right]^{n-i}(1-\varphi)^{n_r} \\ &\quad\times\left[\left(1-\varphi^{i}\right)^{N}-\left(1-\varphi^{i-1}\right)^{N}\right].\end{aligned} \tag{7}$$

The average number of instruction packets sent by the sink is then given by

$$N_1 = \sum_{n=1}^{\infty} n\varphi(n). \tag{8}$$

Since a request packet is sent from the sensing area to the 1st relay node if there is at least one node that does not receive the instruction, the fact that there are n instruction packets sent by the sink means there are $(n-1)$ request packets sent from the sensing area. The average number of request packets sent from the sensing area is therefore

$$N_R = N_1 - 1. \tag{9}$$

In addition, we assume that each node has a probability P_k of being able to successfully complete the agent's task and energy consumption E_k required for the agent to process the data at node k. The energy consumption for the agent to move between nodes is given by E_{kj}. MA_CDP is to minimize the expected energy consumption and the expected time (in terms of the number of hops) to successfully complete the task.

Without loss of generality, we make several assumptions to simplify the MA_CDP model.

(1) The target is stationary. Therefore, $Z_k(t)$, the measurement of node k at time t, remains constant during the MA migration process.

(2) $E_{kj} = E$ for all the migration steps.

(3) $E_k = e$ for all sensor nodes.

The expected energy consumption to complete the task or visit all nodes in failure for a route $R = \langle S_1, S_2, \ldots, S_n\rangle$ is

$$\begin{aligned}E_R &= E + e + p_1E + \sum_{i=2}^{n}\left(\prod_{j-1}^{i-1}\left(1-p_j\right)(E+e+P_iE)\right) \\ &\quad+\prod_{j=1}^{n}\left(1-p_j\right)E.\end{aligned} \tag{10}$$

The equation can be explained as follows. The first site, S_1 is always visited and consumes E amount of energy. Upon arrival, energy e must be spent regardless of success or failure. With probability p_1, the task is successfully completed and the agent can return to node 0 with energy cost of another E. However, with probability $(1 - p_1)$, that is, the failure rate, the agent migrates to node S_2. The expected energy consumption used by the MA moving from node S_1 to S_2 is $(1 - p_1)E$. Similarly, the MA consumes energy E to migrate from node $i - 1$ to node i, and then with probability p_i, it succeeds at node i and returns to node 0 consuming energy E. Hence, the accumulated energy consumption at node S_i is $\prod_{j-1}^{i-1}(1 - p_i)(E + e + p_iE)$. Finally, the last round arises when failure occurs at all nodes and the agent must return to the originating node 0 with an energy consumption E. Furthermore, the expected number of hops to complete the task or visit all nodes in failure, for a route $R = \langle S_1, S_2, \ldots, S_n\rangle$, is

$$\begin{aligned}H_{\mathrm{op}R} &= 1\cdot P_1 + \sum_{i=2}^{n}\left(i\cdot\prod_{j=1}^{i-1}\left(1-p_j\right)p_i\right) \\ &\quad+(n+1)\prod_{j=1}^{n}\left(1-p_j\right).\end{aligned} \tag{11}$$

We use the following equation to model the probability of success:

$$p_k = 1 - \frac{\left(D_{\mathrm{esire}} - \sum_{i=0}^{\mathrm{hop}} I_i\right)}{I_{\max}}, \tag{12}$$

where $I_{\max}$ is the maximum information gain a sensor node can provide. H_{op} is the total node numbers the MA has migrated through. We then have the following theorem.

Theorem 1. *The optimal route for MA_CDP is attained if the nodes are visited in the decreasing order of I_k, $k = 1, 2, \ldots n$, that is, $I_1 > \cdots > I_k > \cdots > I_n$.*

Proof. We employ a similar proof method in [20]. Consider the effect of switching the order of two adjacent nodes on the route, say k and $k+1$. We call this new route as R'; only the kth and $(k+1)$st terms are affected by the switch. The terms appearing before the kth term do not contain anything involving k or $k+1$. Terms that follow the $(k+1)$st term, on the other hand, all contain $(1-p_k)(1-p_{k+1})$ in the same way. Then the difference in the expected energy consumption is

$$E_R - E_{R'} = (E+e)\prod_{j-1}^{k-1}\left(1-p_j\right)(p_{k+1}-p_k), \tag{13}$$

and the difference in the expected number of hops is

$$H_{\mathrm{op}R} - H_{\mathrm{op}R'} = \prod_{j-1}^{k-1}\left(1-p_j\right)(p_{k+1}-p_k). \tag{14}$$

Since $p_k > p_{k+1}$, R is a better route with a smaller expected energy consumption and number of hops.

This indicates that when the kth node on the route has a smaller probability than the $(k+1)$st node in making the agent complete its job, then we can decrease the expected energy consumption and number of hops by switching them.

From (6), we find that the probability of success on a sensor node is directly related to the total information utility the MA accumulates. The higher the total information gain the MA carries, the more likely the agent will finish the task on the current sensor node—thus the higher the probability will be. So the optimal route for MA_CDP is the sequence with decreasing information gain. □

Theorem 2. *If $p_k = p$ for all sensor nodes, then the expected number of hops $H_{\mathrm{op}R}$ algorithm 1/p as the number of sensor nodes n increases.*

Proof.

$$\begin{aligned} H_{\mathrm{op}R} &= 1\cdot p + \lim_{x\to\infty}\left[\sum_{i=2}^{n}\left(i\cdot\prod_{j=1}^{i-1}\left(1-p_j\right)\right)P \right. \\ &\qquad \left. +(n+1)\prod_{j=1}^{n}(1-p)\right] \\ &= \sum_{i=1}^{\infty} i(1-p)^{(i-1)}P + \lim_{x\to\infty}(n+1)\prod_{j=1}^{n}(1-p) \\ &= \frac{1}{p}. \end{aligned} \tag{15}$$

□

This theorem shows that we need to improve the probability of success on each sensor node in order to reduce the number of hops for the MA to finish the task.

5. Simulation Experiments and Evaluation

In order to demonstrate the performance of MA_CDP, we choose a client/server-based scheme (i.e., DD) to compare with MA_CDP. We use NS2 for discrete event. Each task requires periodic transmission of data packets with a constant bit rate (CBR) of 1 packet/s. We assume that both the sink and sensor nodes are stationary. The parameter values used in the simulations are presented in Table 2. The basic settings are common to all the experiments. For each experiment, we simulate for sixty times with different random seeds and get the average results.

Table 2: Simulation setting.

Basic specification	
Network size	500 m∗500 m
Topology mode	Randomized
Total sensor node number	1500
Data rate MAC layer	1 Mbps
Transmission range of sensor node	60 m
Sensed data packet interval	1 s

Three performance metrics are evaluated: *packet*, *delivery,* and *ratio.* It is denoted by *u.* It is the ratio of the number of data packets delivered to the sink to the number of packets generated by the source nodes. *Energy consumption per successful data delivery* It is denoted by *e*. It is the ratio of network energy consumption to the number of data packets successfully delivered to the sink. Let E_{total} be all the energy consumption by transmitting, receiving, and processing during simulation. n_{data} denotes the number of data packets delivered to the sink. Then, $e = E_{\mathrm{total}}/n_{\mathrm{data}}$. *Average end-to-end packet delay* is denoted by T_{ete}. And we also use T_{dd} and T_{ma} to denote the average end-to-end delays in DD and MA_CDP, respectively.

Though these conditions are affected by many parameters, only a set of important parameters is chosen, such as Default, for all sensor nodes has a probability P of being able to successfully complete the agent's task, $P = 0.6$, the duration of the task (T_{task}), data reduction ratio ($r = 0.8$), MA accessing delay ($\tau = 9$ ms), aggregation ratio ($q = 0.2$), size of sensed data of each sensor ($S_{\mathrm{data}} = 1$ KB). If we set q to 0, it means that data aggregation does not work; all the reduced sensed data are concatenated. Only one parameter (e.g., T_{task}, r, q, and S_{data}) is changed in each group while the other parameters are fixed. Several groups of simulations are evaluated.

In Figure 5, in these experiments, we change T_{task} from 10 seconds to 600 seconds, e decreases as T_{task} increases. When the T_{task} is small (i.e., lower than 60 seconds), MA_CDP has higher e than DD because MA_CDP consumes energy (E_q) to transmit processing code from the sink to the target region. If T_{task} is small, n_{data} is small, and e is large. However, when T_{task} is beyond 100 seconds with r equal to 0.8 and q equal to 0.2, MA_CDP has lower e than DD. Thus, to amortize the cost of shipping the processing code once to source node, the source should process enough long streams of data.

In Figure 6, in these experiments, we change MA accessing delay (τ) from 0 seconds to 0.05 seconds, and T_{dd} is constant since changing τ has no effect on DD. Since the

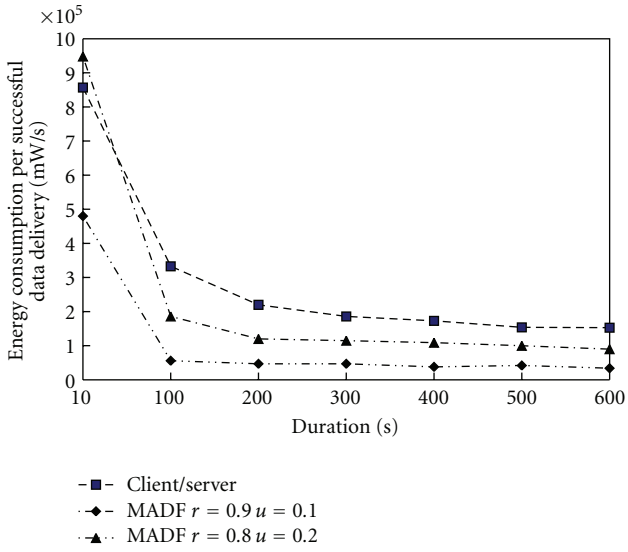


Figure 5: The impact of T_{task} on e.

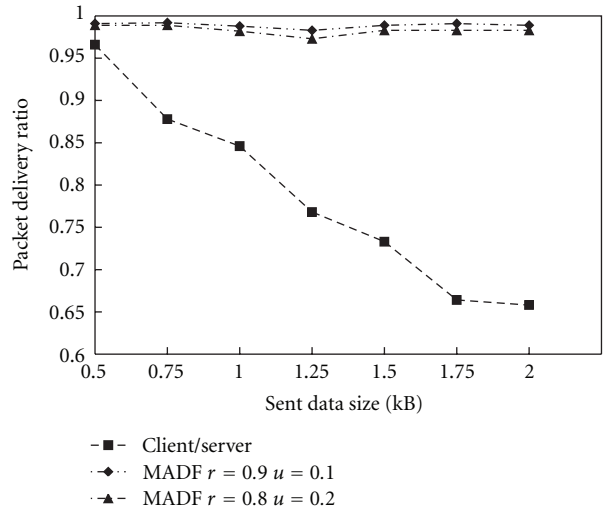


Figure 7: The impact of S_{data}, r, q on u.

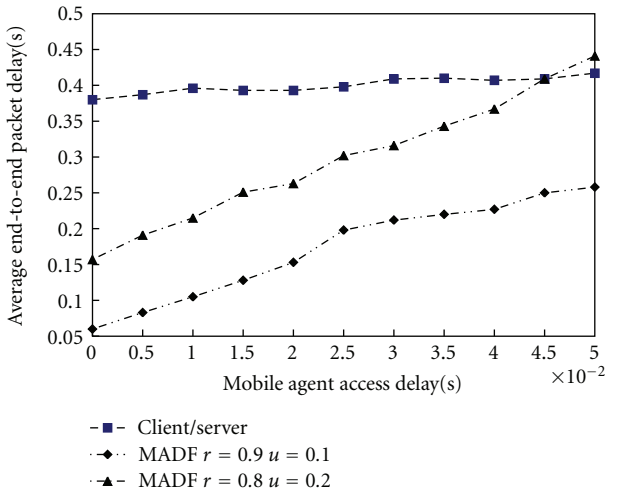


Figure 6: The impact of τ on T_{ete}.

delay of τ is introduced when MA visits each source, τ causes T_{ma}, increase fast if the value is set to a large value. When τ is beyond 0.042 seconds with r equal to 0.8 and q equal to 0.2, MA_CDP has larger end-to-end delay than DD. The value of τ is dependent on the middleware environments of MA system.

In Figure 7, in these experiments, we change the size of sensed data of each sensor (S_{data}) from 0.5 KB to 2 KB by increasing 0.25 KB each time and keep the other parameters unchanged. For MA_CDP, several groups of simulations are evaluated with variables r and q. In Figure 7, MA_CDP always outperforms DD in terms of q. In MA_CDP, only single data flow is sent for each round. In contrast, multiple data flows from individual source nodes are sent in DD. Thus, congestion in DD is more likely to happen than in MA_CDP. When S_{data} increases, the congestion is more serious and q of DD will decrease more.

In Figures 5, 6, and 7, MA-CDP exhibits more consistent and relatively higher reliability, lower energy consumption than DD by compromising end-to-end delay bound possibly in most scenarios. These figures also give hints that MA_CDP should choose r and q appropriately. It can be observed that MA_CDP has no advantage if q is equal to 0.2 and r is smaller than 0.4. Note that the value of r and q is dependent on the type of application. Before we adopt MA-CDP for data dissemination, the features of the application should be investigated. MA_CDP will be selected if enough high r and/or q can be attained.

6. Conclusion

In the environments where the source nodes are close to one another and generate a lot of sensory data traffic with redundancy, transmitting all sensory data by individual nodes not only wastes the scarce wireless bandwidth, but also consumes a lot of battery energy. Recently, MA-based distributed sensor network for collaborative signal and information processing is proposed as a solution to overcome these problems. In this paper, we addressed the problem of optimized itinerary planning for MAs in dense WSNs. Based on a general data aggregation model, we presented a cooperative data processing based on mobile agent algorithm (MA-CDP) and considered that MA dynamically enter a network and can autonomously clone and migrate themselves in response to environmental changes. MA-CDP routing scheme is proposed for MA to efficiently migrate from sink to source, source to source, and source to sink. We showed that the proposed schemes achieve considerable improvements in energy savings, packet delivery ratio, and end-to-end delivery delay.

Disclosure

The material in this paper was presented in part at The 4th International Conference on Wireless Communications, Networking and Mobile Computing,October, 2008 Dalian, China.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grants nos. 61070169, 61170021 and Natural Science Foundation of Jiangsu Province under Grant no. BK2011376 Specialized Research Foundation for the Doctoral Program of Higher Education of China no. 20103201110018, and Application Foundation Research of Suzhou of China no. SYG201034, SYG201118, and sponsored by Qing Lan Project.

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] H. Qi, Y. Xu, and X. Wang, "Mobile-agent-based collaborative signal and information processing in sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1172–1183, 2003.

[3] J. J. Chang, P. C. Hsiu, and T. W. Kuo, "Search-oriented deployment strategies for wireless sensor networks," in *Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC '07)*, pp. 164–171, Santorini Island, Greece, May 2007.

[4] M. Wooldridge and N. R. Jennings, "Intelligent agents: theory and practice," *The Knowledge Engineering Review*, vol. 10, no. 2, pp. 115–152, 1995.

[5] I. Joe, "A path selection algorithm with energy efficiency for wireless sensor networks," in *Proceedings of the 5th ACIS International Conference on Software Engineering Research, Management, and Applications (SERA '07)*, pp. 419–423, August 2007.

[6] M. Chen, K. Taekyoung, and C. Yanghee, "Data dissemination based on mobile agent in wireless sensor networks," in *Proceedings of the IEEE Conference on Local Computer Networks (LCN '05)*, pp. 527–528, Sydney, Australia, November 2005.

[7] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys" 04)*, pp. 81–94, ACM, November 2004.

[8] T. Stathopoulos, J. Heidemann, and D. Estrin, "A remote code update mechanism for wireless sensor networks," Tech. Rep. CENS-TR-30, UCLA, 2003.

[9] C. C. Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava, "A dynamic operating system for sensor nodes," in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys '05)*, pp. 163–176, ACM, June 2005.

[10] A. Dunkels, B. Gronvall, and T. Voigt, "A lightweight and flexible operating system for tiny networked sensors," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 455–462, IEEE Computer Society, 2004.

[11] T. Liu and M. Martonosi, "Impala: A middleware system for managing autonomic, parallel sensor systems," in *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pp. 107–118, ACM, June 2003.

[12] M. Chen, S. Gonzalez, and V. C. M. Leung, "Applications and design issues for mobile agents in wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 20–26, 2007.

[13] H. Qi and F. Wang, "Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '01)*, Helsinki, Finland, June 2001.

[14] M. Chen, T. Kwon, Y. Yuan, Y. Choi, and V. C. M. Leung, "Mobile agent-based directed diffusion in wireless sensor networks," *Eurasip Journal on Advances in Signal Processing*, vol. 2007, Article ID 36871, 13 pages, 2007.

[15] Y. C. Tseng, S. P. Kuo, H. W. Lee, and C. F. Huang, "Location tracking in a wireless sensor network by mobile agents and its data fusion strategies," *Computer Journal*, vol. 47, no. 4, pp. 448–460, 2004.

[16] G. Sharma and R. Mazumdar, "Hybrid sensor networks: a small world," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 366–377, usa, May 2005.

[17] G. Wagenknecht, M. Anwander, T. Braun, T. Staub, J. Matheka, and S. Morgenthaler, "MARWIS: a management architecture for heterogeneous wireless sensor networks," in *Proceedings of the 6th International Conference on Wired/Wireless Internet Communications (WWIC '08)*, 2008.

[18] X. Wang, A. Jiang, and S. Wang, *Advances in Intelligent Computing*, vol. 3645/2005, Springer, Berlin, Germany, 2005.

[19] M. G. Lee and S. Lee, "Data dissemination for wireless sensor networks," in *Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC '07)*, pp. 172–179, Santorini-Island, Greece, May 2007.

[20] W. Choi and S. K. Das, "A novel framework for energy-conserving data gathering in wireless sensor networks," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1985–1996, Miami, Fla, USA, 2005.

[21] Q. Wu, N. S. V. Rao, J. Barhen et al., "On computing mobile agent routes for data fusion in distributed sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 6, pp. 740–753, 2004.

[22] H. Chen, H. Mineno, and T. Mizuno, "An energy-aware routing scheme with node relay willingness in wireless sensor networks," in *Proceedings of the 1st International Conference on Innovative Computing, Information and Control (ICICIC '06)*, pp. 397–400, Beijing, China, August 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 320108, 10 pages
doi:10.1155/2012/320108

Research Article

A Hole-Tolerant Redundancy Scheme for Wireless Sensor Networks

Juhua Pu,[1,2] Yu Gu,[1,2] Yi Zhang,[1,2] Jia Chen,[1,2] and Zhang Xiong[1,2]

[1] *School of Computer Science and Engineering, Beihang University, Beijing 100191, China*
[2] *Research Institute of Beihang University, Shenzhen 518057, China*

Correspondence should be addressed to Juhua Pu, pujh@buaa.edu.cn

Received 12 January 2012; Revised 21 March 2012; Accepted 23 March 2012

Academic Editor: Shukui Zhang

Maximizing network lifetime while not sacrificing coverage and connectivity in wireless sensor networks (WSNs) has attracted many researches during the past few years. One common approach is node scheduling which dynamically schedules some redundant nodes to shut down and keeps alive some necessary nodes to preserve network performance. Previous researches focus either on guaranteeing coverage and connectivity or sacrificing coverage and connectivity to conserve energy. In this paper, we introduce a new hole-tolerant redundancy scheme (HRS) which can prolong network lifetime while maintaining coverage and connectivity performance. This HRS scheme can tolerate some coverage holes when determining redundancy eligibility, so it shuts down more nodes when hole tolerance is higher. Our work takes into account both homoradius WSNs and heteroradius WSNs. The simulation results show that (1) the average coverage percentage varies mildly but network lifetime is prolonged as hole tolerance increases; (2) HRS outperforms several existing lifetime maximization schemes.

1. Introduction

Wireless sensor networks (WSNs) are network systems composed of a large number of inexpensive sensor nodes, deployed in a region to provide monitoring or communication capabilities for commercial or military applications. The positions of sensor nodes need not be engineered or predetermined. This allows random deployment in inaccessible terrains or hazardous environments. Some of the most important application areas of sensor networks include military, natural disasters, health, and the home.

The sensors are unreliable, which have short lifetime, limited power, computational capacities, and memory. They must be regularly recharged or replaced. However, a wild distribution in a complicated environment makes this impossible. To extend the lifetime of a sensor network, one common approach is node scheduling which dynamically schedules sensors between sleep and active cycles based on the cooperative communications and computation among adjacent nodes. It alternately shuts down some redundant nodes and keeps alive some necessary nodes to preserve network performance. Hence, node scheduling in WSNs has become a focus of considerable research in the past few years.

Maximizing the network lifetime while maintaining the degree of coverage and connectivity requested by applications has attracted considerable attention during the past few years [1–4]. However, most of these works have such drawbacks as only focusing on the homogenous sensor network, or having high computing complexity. Thus, trying to solve these problems, our contributions in this paper include three parts.

First, we present a node scheduling scheme that renders the network capable of maintaining monitoring performance with a longer network lifetime. This is a surprising result since prior work either guarantees the degree of coverage and connectivity with more nodes than needed or they sacrifice the degree of coverage and connectivity completeness to save energy. The importance of monitored target area changes with time or site, and it may tolerate some uncovered hole like a part of low monitoring significance. Thus, we take hole tolerance into the consideration which allows more nodes to be shut down when its value is higher.

Second, we introduce a redundancy algorithm applicable when different areas request different degree of coverage. Much research assumes a uniform degree of coverage for the whole area that needs to be monitored. However, some parts

of the monitored area are more important than others, for example, in a war scenario, the control center of an enemy army is far more important to be monitored than any other areas. If we maintain the same performance level for all areas, there will be a waste of power in less important monitoring zones. It is essential that the redundancy algorithm be appropriate for different degrees of coverage. Each node sets its requested degree of coverage based on the monitoring priority of the area to which it belongs. Then, the actual degree of coverage is calculated by the redundancy algorithm which decides the redundancy of different nodes separately.

Third, we provide another redundancy algorithm which applies in networks in which sensor nodes have different sensing ranges.

The rest of this paper is organized as follows. After surveying the related work in Section 2, assumptions and preliminaries are presented in Section 3. Section 4 illustrates the proposed node scheduling scheme HRS whose effectiveness is shown by simulation results in Section 5. Finally, we conclude the paper in Section 6.

2. Related Work

We now summarize the typical work that proposes distinctive node scheduling schemes to prolong network lifetime and maintain a good performance.

In [5], the sensor nodes are divided into disjoint sets which are activated successively to perform the area monitoring tasks individually. At any one time, there is only one set active, all nodes in other sets are in a sleep mode. The goal of this approach is to maximize the number of disjoint sets, as this has a direct impact on prolonging the network lifetime. The solution of the problem which is called the SET-K COVER problem is centralized and has been proven to be NP-complete in [6]. In [7], Berman makes an improvement on [5] by not demanding each node belong to only one single set, that is to say some nodes can be in more than one set. Abrams [8] gives another variation of the SET-K COVER problem. It aims to make as many regions as possible to be covered by as many sets as possible, rather than requiring each set to fully cover the monitored area.

In PEAS [9], each node broadcasts a probe message after sleeping for a random period and enters the on-duty mode only if it receives no replies from neighbors within transmission range; otherwise, it will stay in the sleep mode. Though this method is decentralized, offering high scalability and low cost, it cannot guarantee the degree of coverage. When a node goes to sleep, it may cause a coverage hole.

Paper [10] proposes a distributed scheduling mechanism which, called TIAN hereinafter, can preserve sensing coverage. The mechanism allows a sensor to turn off only if its sensing area is completely covered by its neighbors' sensing areas. The neighbors are called this node's off-duty sponsors, and the sector that a neighbor covers with its sensing area is called a sponsored sector. However, this mechanism only considers those neighbors located within a node's sensing area to be potential off-duty sponsors, while other neighbors are ignored even if their coverage may overlap with this node's sensing area, so this solution may underestimate the number of sensors that can be turned off. Besides, it only guarantees 1-degree coverage which reduces scalability.

Some work is done to give a sufficient and necessary condition to find out off-duty eligible nodes. Huang et al. [11] and Liu et al. [12] propose perimeter coverage algorithms, by which they calculate the coverage degree of every arc on a node's sensing circle to judge if the node is redundant or not. The conditions proposed in these algorithms are sufficient and necessary, but the computing complexity is too high. Paper [13] proposes a protocol called CCP which is able to configure itself to any feasible degree of coverage and connectivity in order to support different applications and environments with diverse requirements. This flexibility allows the network to self-configure for a wide range of applications and environments. But this mechanism has a quite high computing complexity of $O(n^3)$, where n denotes the number of neighbors of the sensor. Paper [14] extended to *arbitrary region*. Zhang and Hou [15] propose a distributed mechanism, optimal geographic density control (OGDC), to maximize the number of sleeping sensors while ensuring that the working sensors provide complete 1 coverage and 1 connectivity. OGDC tries to minimize the overlapping area between the working sensors. OGDC's protocol is quite similar to that of the sponsored sector mechanism, except that they use different on-duty/off-duty eligibility rules and the sponsored sector mechanism is more conservative when turning off sensors.

To sum up, existing work on node scheduling has the following characteristics. (1) They cannot achieve a good balance between prolonging the network lifetime and guaranteeing the degree of coverage and connectivity with low computing complexity. (2) Much research assumes a uniform coverage degree for the area that needs to be monitored. (3) They focus on WSN whose nodes have the same sensing and transmission radius (we call it homoradius WSNs). We will try to solve these problems in this paper.

3. Network Model and Preliminary

3.1. Network Model. We consider a WSN consisting of a set S of n static sensor nodes. Each node $u \in S$ can sense events of interest in its sensing range and communicate with nodes in its transmission range. We make the natural assumption that there are no two sensors at the same location. The sensors are distributed over the monitoring area, a large 2-dimensional area that we are interested in monitoring. The monitoring area is typically significantly larger than the sensing range of a single sensor.

For any sensor node u whose sensing radius and communication radius are $\mathrm{Rs}(u)$ and $\mathrm{Rc}(u)$, its *sensing area* $S(u)$ and *transmission area* $T(u)$ are open discs, centered at u, with radius $\mathrm{Rs}(u)$ and radius $\mathrm{Rc}(u)$, respectively, and we assume $\mathrm{Rc}(u) > 2\mathrm{Rs}(u)$ under which condition, network coverage implies network connectivity. Each point in $S(u)$ is said to be *covered* by node u. Any two sensor nodes u and v are termed adjacent or *neighbors* if they are located within the transmission area of each other. If a point in monitored area is covered by and only by k different nodes (neighbors), we call this point is *k-covered*, and the coverage degree of this point is k.

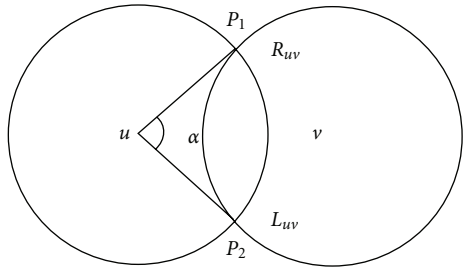


Figure 1: Contributing angle.

For any area, if it requests all the points in this area having a coverage degree no less than k, we call this area has a requested coverage degree of k. For any node u, if its sensing range can guarantee its requested coverage degree by its neighbors, u is a redundant node and can be turned off.

3.2. Preliminary and Problem Statement. In this paper, we focus on the hole-tolerant scheme in both *homoradius* and *heteroradius* WSNs. A WSN is a homoradius WSN if all its sensors have the same sensing and transmission radius. Otherwise, it is a heteroradius WSN.

In hole-tolerant scheme, the WSN can tolerate coverage hole(s) in the monitor area to some extent. For a monitor area with a requested coverage degree of k, one of its subarea is called a *coverage hole* if this subarea is less than k-*covered*. The percentage of the subarea whose coverage degree is at least k to the whole monitor area is called *coverage percentage*. In hole-tolerant WSN, the network can fulfill normal monitor when the coverage percentage is no less than a specific value. We define a parameter *hole tolerance* which indicates the biggest coverage hole a WSN can withstand before normal monitoring operation is impaired.

Before we state the problem, let us give two necessary definitions, namely, contributing node and contribution angle. Suppose nodes u and v are neighbors in Figure 1, v is a *contributing node* (to node u) if v's coverage to $S(u)$ is taken into consideration in the redundancy decision phase (shown in Section 4.2). Suppose both sensing areas $S(u)$ and $S(v)$ intersect at points P_1 and P_2, then angle α is defined as the *contribution angle* of node v to node u, L_{uv} and R_{uv} are the left bound and right bound of this contribution angle between node u and v.

With the definition above, the problem this paper tries to solve lies in the following. (1) Similar to TIAN, how to provide a redundancy decision scheme by explore the relationship among contributing angles of different neighbors. (2) How can this scheme achieve high performance without underestimation or overestimation as well as possible? (3) How can this scheme provide configuration of different requested coverage degree and can be used in hole-tolerant WSNs?

4. Lifetime-Extending Scheme HRS

Our scheme consists of three parts: initialization, redundancy decision, and status transfer. The initialization phase randomly distributes the sensors, informing each sensor of some parameters. In the redundancy decision phase, each sensor node runs the redundancy algorithm to decide its eligibility to turn off. If it is eligible, then the node shifts to the status transfer phase which puts the node into a sleep mode properly. The following sections will describe them in detail.

4.1. Initialization. An initialization phase is executed at the beginning of the network operation. During the initialization phase, each sensor acquires the following local information: location, sensing range, initial status, remaining energy, and requested coverage degree.

If the whole monitoring area in our network model is composed of subareas with different requested coverage degree, how can we set the requested coverage degree for a sensor u when judging its off-duty eligibility. If the sensing area of u is within one subarea, u's requested coverage degree is equal to the requested coverage degree of this subarea. If the sensing area of u is divided by more than one subareas, we use the demanded coverage degree of the sub-area which is the nearest to a sensor u as u's requested coverage degree when judging its off-duty eligibility.

4.2. Redundancy Decision

4.2.1. Redundancy Decision in Homoradius WSN. In the redundancy decision phase, a sensor node runs the redundancy algorithm to figure out if it is eligible to go to the sleep mode. Algorithm 1 provides the redundancy decision algorithm in homoradius WSN.

Consider two sensors u and v located in (x_u, y_u) and (x_v, y_v), respectively. Denote the distance between u and v by $d(u,v) = \sqrt{|x_u - x_v|^2 + |y_u - y_v|^2}$.

As shown in Figure 1, we take the contribution angle α as an approximation of the sector (denoted by $S_{u,P_1 \to P_2}$) bounded by radius uP_1, radius uP_2, and inner arc P_1P_2. That is to say, we assume the sensing area of node v with a contribution angle α (to sensor node u) covers $S_{u,P_1 \to P_2}$. Though there is a coverage hole in $S(u)$, with the tolerance parameter *toler*, we know that this assumption will not affect monitoring operation too much, which can be seen from the simulation results in Section 5.

We gradually increase the contributing nodes loop by loop as illustrated in Figure 2 (by step 14 in Algorithm 1). Initially, the contributing nodes are within area 1, then, we add nodes in loop area 2 as contributing nodes, next, loop area 3, and so on. Obviously, there will be an increase in coverage holes as contributing nodes are added, and this is where the flexibility of our intelligent algorithm comes into its own.

The algorithm operates as follows: we define a contributing node v as one whose distance to u is within $(D_1, D_2]$, where $D_1 = 0$, $D_2 = \mathrm{Rs}(u)$ for initialization.

Step 1. For each contributing node v, we determine the left bound L_{uv} and right bound R_{uv} of the contribution angle.

Step 2. Place all the points L_{uv} and R_{uv} on the line segment $[0, 2\pi]$ which is considered end-to-end, and mark them with node *id* and the flag L or R.

Begin
Input Kr(n), Rs(n), toler
/* Kr(n) are the demanded coverage degrees for all nodes
/* Rs(n) are the sensing radius for all nodes
/* toler is hole tolerance of the WSN.
(1) $D_1 = 0$
(2) $D_2 = \text{Rs}(u)$
(3) **while** $D_2 \leq (1 + \sqrt{\text{toler}})\text{Rs}(u)$ **do**
(4) **for** each node v of u's neighbors
(5) **if** $D_1 < d(u, v) \leq D_2$ **then**
(6) calculate the left bound $\alpha_L(u,v)$ and right bound $\alpha_R(u,v)$ of the contribution angle
(7) **end if**
(8) **end for**
(9) calculate the coverage degree of each bound point to get the minimum coverage degree $K_{\min}(u)$
(10) **if** $K_{\min}(u) \geq \text{Kr}(u)$ **then**
(11) **Return** *eligible* =1/* node u is eligible to be turned off
(12) **else**
(13) $D_1 = D_2$
(14) $D_2 = D_1 + \Delta d$/*Δd is the step size
(15) **end if**
(16) **end while**
(17) **Return** *eligible* = 0/* node u is not eligible to be turned off
End

Algorithm 1: The redundancy algorithm in homoradius WSN.

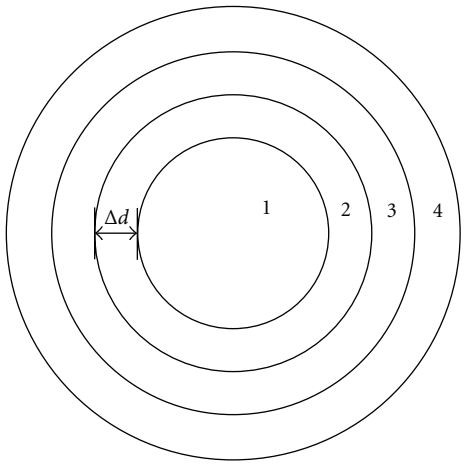


Figure 2: Increase of contributing nodes.

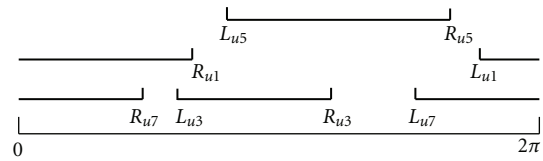


Figure 3: Line segment coverage diagram.

Step 3. Determine the coverage degree of each point gotten in Step 2. In Figure 3, a coverage degree of a point, P, demonstrates the coverage degree of the angle range bounded by P and its nearest left neighbor on the segment. For example, coverage degree of point L_{u3} demonstrates the coverage degree of the angle $[R_{u7}, L_{u3}]$. Operation is as follows.

For each point on the line segment $[0, 2\pi]$, if it is marked as "R", then visit points from right to left starting from its first left neighbor until meeting the point with the same node *id* as itself, increase the coverage degree of all the nodes covered during the traverse, respectively, by 1 degree.

If the minimum coverage degree of all the angle ranges is not smaller than the demanded coverage degree of $S(u)$, then u is redundant, end the process. Else, go to Step 4.

Step 4. Let $D_1 = D_2$, $D_2 = D_2 + \Delta d$ (Δd is a small step size to increase the range), *if* $D_2 \leq (1 + \sqrt{\text{toler}})\text{Rs}(u)$; *then,* add neighbors whose distance to u is within $(D_1, D_2]$ into contributing nodes, and *go to* Step 1; *else*, end the process.

We now explain more about the parameters in the algorithm. First, we can set $D_1 = 0$, $D_2 = (1 + \sqrt{\text{toler}})\text{Rs}(u)$ directly as we do in heteroradius WSN in Section 4.2.2, then all nodes whose distance to u are within $(0, (1+\sqrt{\text{toler}})\text{Rs}(u)]$ are taken into computation of the coverage degree. The reason why we bother to increase the contributing nodes loop by loop is that, when the nodes are densely scattered, the algorithm can finish the computation earlier to get a higher efficiency.

Second, the value of Δd influences the performance of the algorithm, for example, convergence rate. We can adjust it according to the nodes density, whether there exists a best value for each density needs a further study.

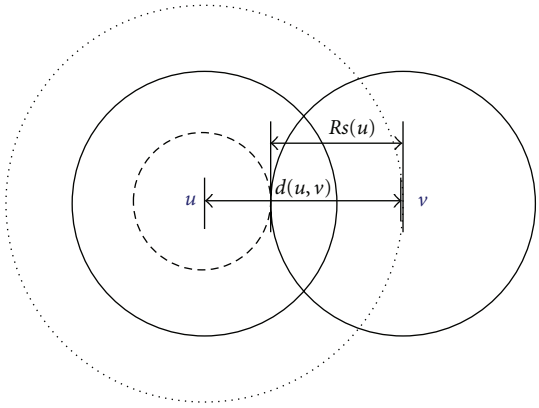


Figure 4: Condition explanation.

With the hole tolerance parameter, *toler*, we achieve a tradeoff between network performance and lifetime. From the condition: $d(u,v) \leq (1+\sqrt{\text{toler}})\text{Rs}(u)$, we can see that the larger *toler* is, the more contributing nodes are used and a greater number of coverage holes occur which also means poorer network performance. However, at the same time, more nodes will be determined to be redundant, so we will achieve a longer network lifetime.

Let us explain the condition $d(u,v) \leq (1+\sqrt{\text{toler}})\text{Rs}(u)$.

In Figure 4, we see node v is a distant contributing node of node u, both have a sensing range of $\text{Rs}(u)$. The circle centered at u with radius $(d(u,v) - \text{Rs}(u))$ is a definite coverage hole. Without lose of generality, we assume the area of coverage holes in other areas of $S(u)$ is ε ($\varepsilon \geq 0$), then

$$\frac{\pi(d(u,v)-\text{Rs}(u))^2+\varepsilon}{\pi\text{Rs}(u)^2} = \text{toler}. \tag{1}$$

So we get $d(u,v) \leq (1+\sqrt{\text{toler}})\text{Rs}(u)$.

Now, we will show how the algorithm works by an example. Suppose node u has 8 neighbors as shown in Figure 5(a), and the demanded coverage degree is 1.

First, for each contributing node directly within our sensing range, 1, 3, 5, and 7, calculate the left and right bounds, and mark them with node *id* and flag *L* or *R* as in Figure 5(b).

Second, mark the covered areas on a line segment coverage diagram as shown in Figure 3.

Third, calculate the coverage degree of each point. In Figure 6, we visualize the coverage for each section of the line segment coverage diagram. We can see that the minimum coverage degree is 1, which is the demanded coverage degree, so node u is determined to be redundant. If the required coverage degree is 2, we would have to add more neighbors as contributing nodes by increasing the radius of our comparison loop.

4.2.2. Redundancy Decision in Heteroradius WSN. In heteroradius WSN, the redundancy algorithm is provided in Algorithm 2 and shares most of the characteristics of the homoradius WSN, so next we only describe the different parts.

Suppose the sensing radius of node u and v meets $\text{Rs}(u) < \text{Rs}(v)$, their relative position is shown in Figure 7. Though $0 < d(u,v) \leq \text{Rs}(u)$, we cannot calculate the left and right bounds of the contribution angle, L_{uv} and R_{uv}, since there is no such angle. In this case, we add 1 coverage degree to each existing coverage line segment.

4.3. Status Transfer. Each node determines its redundancy using the redundancy algorithms (See Algorithms 1 and 2) and may switch status dynamically when its redundancy eligibility changes. Redundant nodes should enter sleep mode, while nonredundant ones are working. However, if more than one node goes to sleep simultaneously, it may cause coverage hole; or if they turn active at the same time, network energy may be wasted. To resolve this problem, different collision prevention mechanisms or scheduling schemes are introduced into this field [10, 11]. But this is still an open issue. Thus, in this paper, we also introduce a different mechanism. But the main contribution of this paper is that it proposed algorithms for nodes to acquire their off-duty eligibility (i.e., to determine if they are redundant or not). Also, our algorithms are independent to the collision prevention mechanisms. So, in the analysis and simulation part, we only focus on evaluating the performance of Algorithms 1 and 2. Evaluating and further research on the collision prevention mechanism are our future work.

Now, we describe in detail this mechanism. Each sensor at any moment is in one of the following five states: ACTIVE, that is, the sensor monitors its monitoring region and communicates with other sensors; SLEEP, that is, the node is put into a low power mode to save energy; JUDGE: the sensor collects information of its neighbors and runs the redundancy decision algorithm; OFF-DELAY: the sensor waits for a period before going to SLEEP; ON-DELAY: the sensor waits for a period before going to ACTIVE.

(1) When a node is in ACTIVE state: we trigger a timer T_a, if the sensor is going to run out of energy, it sends a Drop message and then goes to SLEEP, else, if T_a expires, it switches to the OFF-DELAY state.

(2) When nodes being in SLEEP state: when a sensor goes sleep, we start a timer T_{sp} which makes each sleep node wake up after a certain period of time. When it expires, the sensor transfers its state to the JUDGE state.

(3) When nodes being in JUDGE state: in this state, the sensor does two things. First, it broadcasts a Hello message to regenerate its neighbor table and learn their positions from its neighbors' reply messages. Then, it runs the redundancy decision algorithm to determine whether it is redundant or not. It transfers to the ON-DELAY state if it is not redundant; else it enters the OFF-DELAY state.

(4) When nodes being in ON-DELAY state: we set a timer T_{ond}, if the sensor receives a Join message, then it goes back to the JUDGE state. If T_{ond} expires, the sensor switches to ACTIVE and sends a Join message.

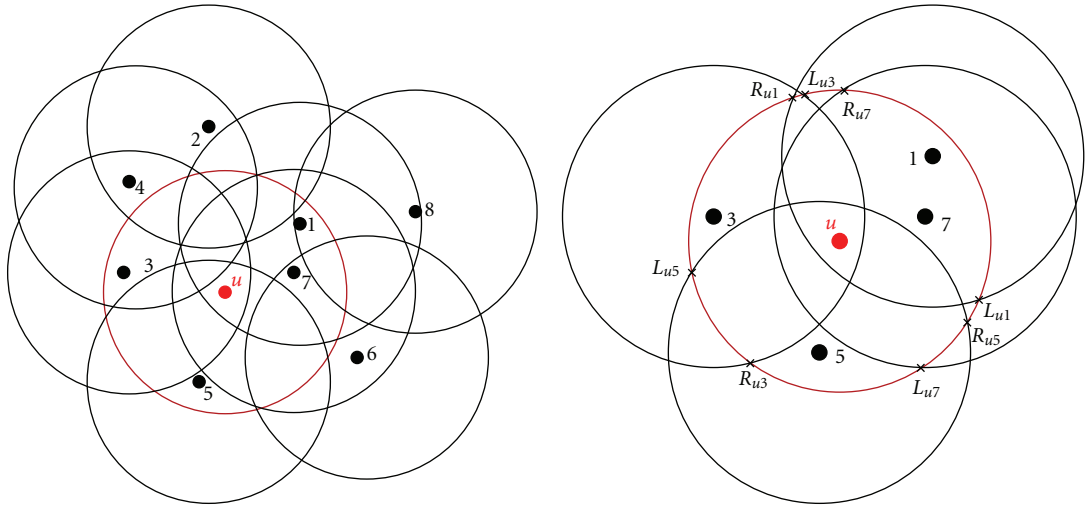


Figure 5: Example of the algorithm.

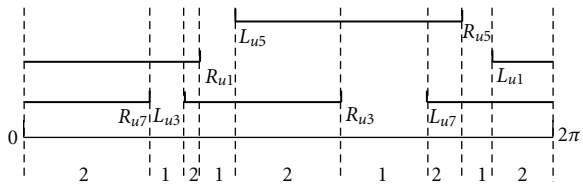


Figure 6: Line segment with coverage degree.

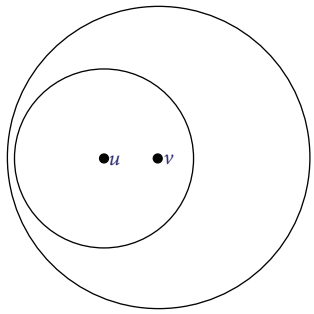


Figure 7: A different case between heteroradius WSN and homoradius WSN.

(5) When nodes being in OFF-DELAY state: set a timer T_{offd}. If the sensor receives a Drop message, then it goes back to the JUDGE state. If T_{offd} expires, the sensor sends a Drop message and goes to SLEEP.

The values of the timers will affect the responsiveness of HRS. T_a and T_{sp} should be considerably greater than T_{ond} and T_{offd}, otherwise nodes may spend too much time in decision states and thus no long enough time for sleep. But they must not be too much greater, otherwise the decisions cannot be made timely, and thus there may be many coverage holes appear or additional unnecessary redundancy may occur. Moreover, these timers are related to the remaining energy. For example, a node with more remaining energy should increase T_a to balance energy consumption of the network. In addition, the density of nodes in the network has an effect on the timers. For example, for a denser network where a node has more neighbors, T_{ond} and T_{offd} should be long enough for a node to collect all the Join or Drop messages from its neighbors. Whether there exist any optimal values for these timers is left as our future work.

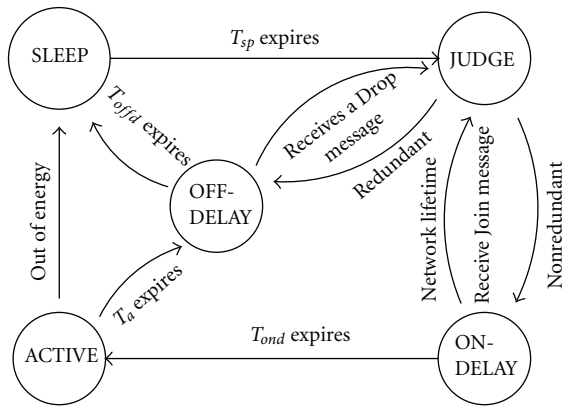


Figure 8: Status transfer FSM.

Figure 8 provides a useful visualization of the status transfer in an FSM.

5. Analysis and Simulation

Now, we evaluate the performance of HRS by simulations. Similar with CCP and TIAN, we also let each node decide whether to turn off or not in a random sequence, and the decision of each node is visible to all the other nodes. Namely, according to this random sequence, after calculating its actual coverage degree by HRS, CCP, or TIAN, each node

Begin
Input Kr(n), Rs(n), toler
/* Kr(n) are the demanded coverage degrees for all n nodes
/* Rs(n) are the sensing radius for all n nodes
/* toler is hole tolerance of the WSN.
(1) **for** each node v of u's neighbors
(2) **if** Rs(u) > Rs(v) && $d(u, v)$ < Rs(u) − Rs(v) **then**
(3) **continue**
(4) **end if**
(5) **if** Rs(u) < Rs(v) && $d(u, v)$ < Rs(u) − Rs(v) **then**
(6) add 1 coverage degree to each existing bound point
(7) **else**
(8) **if** $d(u, v) \leq \text{Rs}(v) + \sqrt{\text{toler}}\text{Rs}(u)$ **then**
(9) Calculate the left bound $\alpha_L(u,v)$ and right bound $\alpha_R(u,v)$ of the contribution angle
(10) **end if**
(11) **end for**
(12) Calculate the coverage degree of each bound point to get the minimum coverage degree $K_{\min}(u)$
(13) **if** $K_{\min}(u) \geq \text{Kr}(u)$ **then**
(14) **Return** *eligible* = 1/* node u is eligible to be turned off
(15) **else**
(16) **Return** *eligible* = 0/* node u is not eligible to be turned off
End

Algorithm 2: The Redundancy Algorithm in heteroradius WSN.

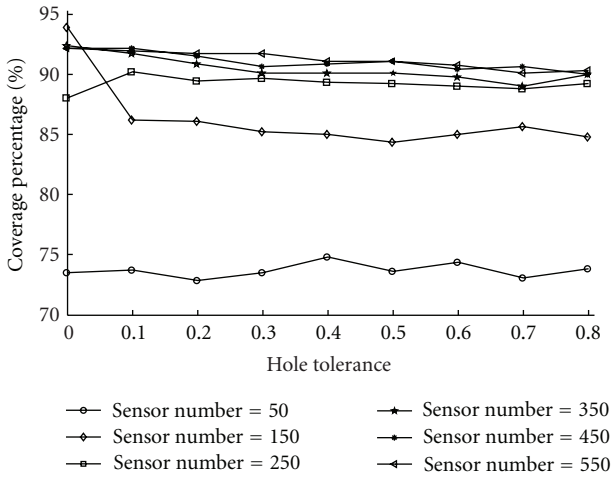


Figure 9: Coverage percentage affected by hole tolerance.

compares this degree with the requested coverage degree. If the former is no less than the later, this node turns itself off.

We have taken monitored area as 20 × 20 units, demanded coverage of 1. We experimented with 50 to 600 sensor nodes, with sensing and transmission ranges of 2 and 5 units, respectively. All sensor nodes are positioned according to uniform random distribution. We uniformly assigned initial energy of 300 units to each sensor node. According to the research about the energy consumption in paper [16], we assume that the energy each node consumes for redundancy decision (including in the temporary status as ON-DELAY and JUDGE), SLEEP and ACTIVE are 0.5, 1, and 10, respectively. In our simulations, the *toler* for each node set ranges from 0 to 0.8, increasing by 0.1 each time. We do two sets of simulations. The first one tests the impact of hole tolerance on network coverage and lifetime, and the second one evaluates the performance of HRS. In all of our simulations, we assume the sensing network is alive until the network coverage percentage is less than 50% (including the simulation of CCP and TIAN in this paper).

5.1. The Impact of Hole Tolerance. All nodes run HRS independently. We compute the network coverage percentage at every time slot until the coverage percentage of the network is lower than a demanded limit (we set it as 50% here) and then figure out the average coverage percentage. The results are shown in Table 1 and Figure 9. We can see that, for a certain number of nodes, the average coverage percentage does not appear to be much affected by the hole tolerance level.

Next, we examine how hole tolerance affects network lifetime. Figure 10 shows a 3D surface plot of the network lifetime for different hole tolerance values. We can see that increasing the hole tolerance of the network results in a longer network lifetime. To have a clearer view, we calculate the average lifetime of WSNs with different number of nodes (say 200, 400, or 600 in our simulations) and different hole tolerance (say from 0.1 to 0.8 in our simulations). As shown in Figure 11, it is clear that the increase in hole tolerance from 0 to 0.4 has a great effect after which its performance increase is much less. We can also get the information that gives a certain network (obviously the node number is certain), then network lifetime will be increased as tolerance increases.

From Figure 11, we can easily observe that the curves representing the results gotten from WSNs with more nodes are steeper than those gotten from WSNs with fewer nodes.

Table 1: The impact of hole tolerance on coverage percentage.

n	Toler								
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
50	71.9	75.1	75.8	70.0	76.6	71.7	76.2	71.6	71.0
100	89.7	93.3	93.8	95.1	91.7	91.9	93.3	87.5	91.9
150	97.3	83.1	85.0	83.6	86.2	84.5	85.1	84.2	86.3
200	88.6	87.7	88.5	86.9	88.2	89.5	88.5	86.6	89.7
250	85.5	88.8	89.0	87.9	88.5	87.3	88.3	87.1	88.8
300	88.4	89.8	89.8	89.9	91.5	91.1	88.9	88.8	89.6
350	93.5	92.3	88.9	90.8	88.7	89.4	89.7	88.8	89.7
400	92.3	91.9	91.0	90.6	89.6	90.7	90.2	89.8	89.9
450	91.6	93.3	90.3	91.2	90.8	91.5	90.0	90.5	88.3
500	93.0	92.7	92.9	91.5	89.0	91.5	92.4	89.3	92.1
550	93.3	92.4	92.8	92.4	91.1	91.6	90.9	89.5	91.9
600	94.4	90.9	91.5	92.1	90.7	91.6	91.4	90.6	91.2

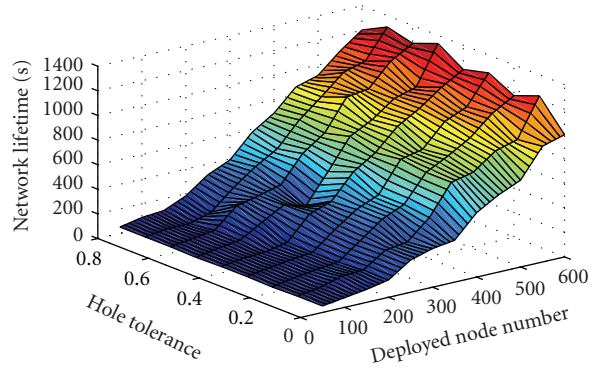


Figure 10: The impact of hole tolerance.

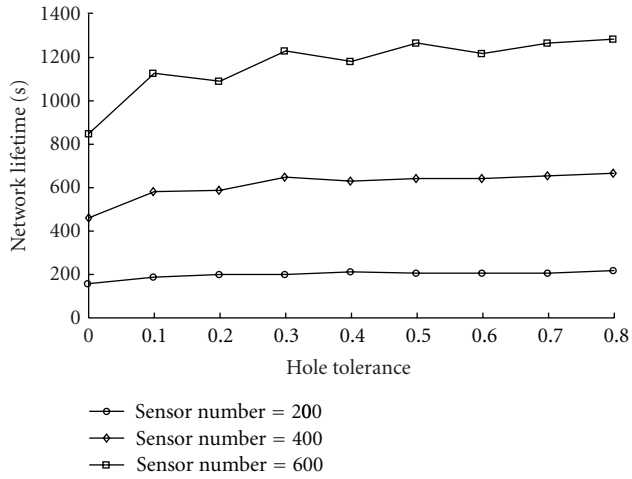


Figure 11: Hole tolerance versus network lifetime.

This is because of the little coincidence phenomena. That is to say, when the number of nodes is small, the nodes are sparsely scattered so that there is little or no coincidence of any two sensing area. In this case, the increase of *toler* has little effect on the network lifetime. Thus, our HRS scheme fits denser network better. Since WSN is normally composed of a large number of sensor nodes by definition, our scheme fits WSN very well.

From the above two experiments, we see that average coverage percentage varies moderately and the hole tolerance does not have obvious effect on it. However, network lifetime is increased to a certain degree by increasing hole tolerance, though, after 0.4, the increase is much less. So we can achieve a longer network lifetime when using our hole tolerance mechanism.

5.2. The Performance of HRS. This experiment compares the performance of our HRS to two very popular protocols TIAN and CCP. Similar to HRS, TIAN and CCP are decentralized protocols designed to preserve coverage by turning off redundant nodes to conserve energy in a sensor network. The eligibility rules in the TIAN protocol, CCP, and HRS are different. The main advantage of HRS lies in its ability to configure the network to a specific hole tolerance level, which is not supported in TIAN and CCP protocols.

We perform the same experiment as before with HRS using tolerance levels of 0, 0.4, and 0.8. The experiment results, displayed in Figure 12, show that, even when hole tolerance is 0.8 (toler = 0.8), HRS gets close to the same average coverage percentage as TIAN and CCP. When toler = 0, the curve of HRS and TIAN almost coincide, which means, through the tolerance modulation, HRS can achieve the same coverage performance as TIAN. We then compare the average number of alive nodes gotten by different schemes. As can be seen from Figure 13, when the hole tolerance is greater than 0 (toler > 0), HRS has a considerably smaller number of active nodes and hence leads to more energy conservation than the other two protocols do. We subsequently draw a picture of the lifetime comparison in Figure 14 from which we clearly see that HRS gets longer lifetime than others as we expect.

5.3. Complexity Analysis. Now, we analyze the communication complexity and computing complexity of these three algorithms. For the communication complexity, since

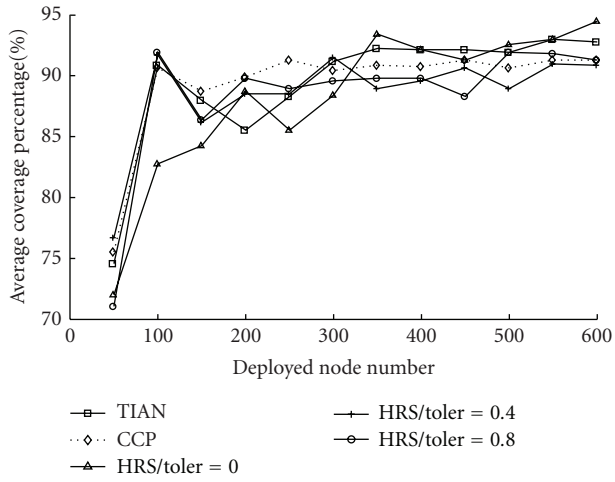


Figure 12: Average coverage percentage comparison.

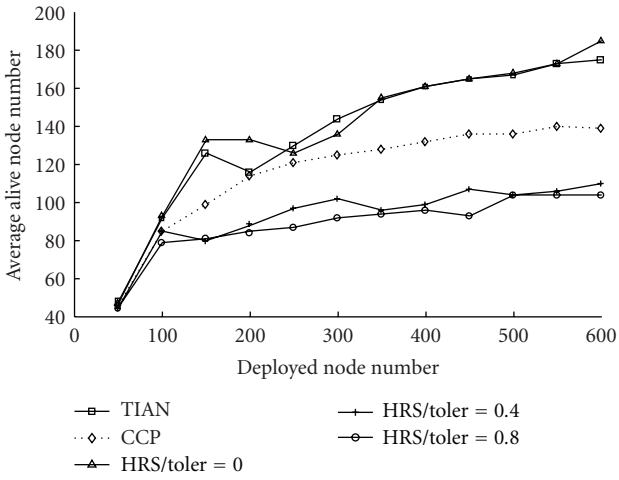


Figure 13: Average alive node number comparison.

HRS, CCP and TIAN all need and only need to collect neighbor nodes to determine the off-duty eligibility by communication, their communication complexity is similar.

For the computing complexity, as can be seen from the steps HRS follows, the step to calculate the coverage degree of each bound point to get the minimum coverage degree takes the longest time. This step has a computing complexity of $O(n \log n)$ to sort all the left and right bounds of contribution angles by Quicksort, while all the other steps with the computing complexity of or less than $O(n)$. Thus, the total computing complexity of HRS is $O(n \log n)$, which is much lower than CCP of $O(n^3)$. Since our work focuses on the redundant WSNs where the number of nodes is large and each node may have enough neighbors, this decrease is significant in entire networks. The computing complexity of TIAN is similar to that of HRS, while HRS performs better than TIAN.

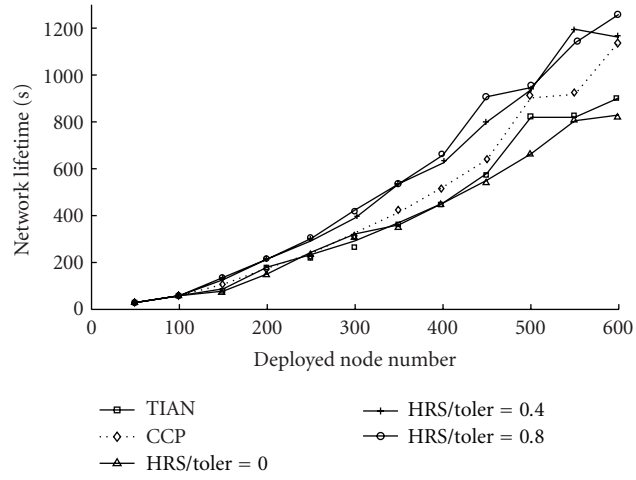


Figure 14: Network lifetime comparison.

6. Conclusion and Future Work

In this paper, we have designed a hole-tolerant redundancy scheme, HRS, for WSNs. This scheme introduces a parameter, called *hole tolerance*, which renders the network capable of varying from 100% strict coverage performance to moderately poorer ones to achieve longer network lifetime. It allows different areas to set different requested coverage degrees, and it is applicable in both homoradius networks and heteroradius networks.

Our experiments show that *hole tolerance* has no remarkable impact on average coverage percentage and that network lifetime will be extended as hole tolerance is increased. We also compare the performance of HRS with another two famous schemes, TIAN and CCP. HRS achieves a similar average coverage percentage to TIAN and CCP, and using hole tolerance can reduce the number of active nodes resulting in a considerable increase in network lifetime.

For future work, on the one hand, we will evaluate and do more research on the collision prevention mechanism proposed in this paper. On the other hand, HRS is a hole-tolerant scheme, the network can fulfill normal monitor task when the coverage percentage is not less than a specific value. We demonstrated how the hole tolerance affects the coverage percentage. But we did not give the mathematical relationship between hole tolerance and coverage percentage. This would be our future work.

Acknowledgments

This work was supported in part by China's Natural Science Foundation (61173009 and 61070169), the Chinese National Programs for High Technology Research and Development (2011AA010502), Doctoral Fund of Ministry of Education of China (20091102110017), Natural Science Foundation of Jiangsu Province (BK2011376), and Specialized Research Foundation for the Doctoral Program of Higher Education of China (20103201110018).

References

[1] H. M. Ammari and S. Das, "On the design of k-covered wireless sensor networks: self-versus triggered sensor scheduling," in *Proceedings of the 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks and Workshops (WoWMoM '09)*, pp. 1–9, Kos, Greece, June 2009.

[2] J. Hu and X. Hu, "Nonlinear filtering in target tracking using cooperative mobile sensors," *Automatica*, vol. 46, no. 12, pp. 2041–2046, 2010.

[3] F. Y. Shen, C. L. Liu, and J. Zhang, "A distributed coverage-aware sleep scheduling algorithm for wireless sensor networks," in *Proceedings of the 6th International Conference on Information Technology (ITNG '09)*, pp. 524–527, IEEE Computer Society, Las Vegas, Nev, USA, April 2009.

[4] Q. Zhao and M. Gurusamy, "Lifetime maximization for connected target coverage in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 6, pp. 1378–1391, 2008.

[5] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proceedings of the International Conference on Communications (ICC '01)*, pp. 472–476, Helsinki, Finland, June 2001.

[6] M. Cardei and D. Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wireless Networks*, vol. 11, no. 3, pp. 333–340, 2005.

[7] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Power efficient monitoring management in sensor networks," *Proceedings of the IEEE Wireless Communications and Networking Conference*, vol. 4, pp. 2329–2334, 2004.

[8] Z. Abrams, A. Goel, and S. Plotkin, "Set K-cover algorithms for energy efficient monitoring in wireless sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 424–432, April 2004.

[9] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: a robust energy conserving protocol for long-lived sensor networks," in *Proceedings of the 23rd IEEE International Conference on Distributed Computing Systems (ICDCS'03)*, pp. 28–37, Providence, RI, USA, May 2003.

[10] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 32–41, New York, NY, USA, September 2002.

[11] C. Huang, Y. Tseng, and H. Wu, "Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network," *ACM Transactions on Sensor Networks*, vol. 3, no. 1, 2007.

[12] Y. Liu, J. Pu, S. Zhang, Y. Liu, and Z. Xiong, "A localized coverage preserving protocol for wireless sensor networks," *Sensors*, vol. 9, no. 1, pp. 281–302, 2009.

[13] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 36–72, 2005.

[14] G. Fan and S. Jin, "A simple coverage-evaluating approach for wireless sensor networks with arbitrary sensing areas," *Information Processing Letters*, vol. 106, no. 4, pp. 159–161, 2008.

[15] H. Zhang and J. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *International Journal of Wireless Ad Hoc and Sensor Networks*, vol. 1, no. 1, pp. 89–124, 2005.

[16] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, 2002.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 863634, 10 pages
doi:10.1155/2012/863634

Research Article

Cooperative Transmission in Cognitive Radio Ad Hoc Networks

Juncheng Jia[1] and Shukui Zhang[1,2]

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjin 210093, China*

Correspondence should be addressed to Juncheng Jia, jiajuncheng@suda.edu.cn

Received 11 January 2012; Accepted 21 March 2012

Academic Editor: Yong Sun

Cognitive radio technology is the key to realize dynamic spectrum access system and promote the spectrum utilization through exploiting the spectrum holes left by primary users. However, the spatial heterogeneity of spectrum availability imposes special challenges for efficient utilization of the spectrum resources for *cognitive radio ad hoc networks* (CRAHNs). The cross-layer cooperative transmission scheme is a promising approach to improve the efficiency of spectrum utilization and improve the performance of cognitive radio networks. Such an approach leverages relay-assisted discontiguous OFDM (DOFDM) for data transmission at physical and MAC layers in a basic three-node configuration. With this scheme, a relay node will be selected that can bridge the source and the destination using its common channels between those two nodes. In this paper, we investigate the application of such a cooperative transmission scheme to address the spectrum heterogeneity issue in CRAHNs. In particular, we describe several types of cooperative transmission and formulate a new resource allocation problem with joint relay selection and channel allocation. We propose a heuristic algorithm to solve the resource allocation problem, which is based on the metric of utility-spectrum ratio of transmission groups. Simulations demonstrate the performance improvement of the cooperative transmission over the direct transmission.

1. Introduction

There have been a lot of innovations of wireless devices and wireless services in recent years. However, due to the current fixed spectrum assignment and allocation rules, there is virtually no available spectrum band to experiment and deploy these new wireless products. Meanwhile, recent spectrum measurement reports have shown significantly unbalanced usage of spectrum, with some frequency bands largely unoccupied most of the time and some other frequency bands heavily used [1]. Observing such inefficient utilization of the scarce and valuable spectrum resource, new spectrum access rules are undergoing rapid development, whose core idea is to allow *secondary users* (or *unlicensed users*) to access spectrum holes left by *primary users* (or *licensed users*). Cognitive radio [2] has been proposed as the means for secondary users to promote the efficient utilization of the spectrum by exploiting the existence of spectrum holes.

Cognitive radio ad hoc network (CRAHN) is one of the major application fields, where there exist special research challenges and opportunities [3]. One important issue is the spectrum heterogeneity faced by CRAHNs, which is due to the location difference among different secondary users, dynamic traffic of primary users, and opportunistic spectrum access nature of secondary users. Several spectrum measurement reports have already demonstrated such spectrum heterogeneity. For example, in the UHF spectrum band, television stations represent the largest incumbent users. Spectrum heterogeneity exists on both large and small scales. For a wide area, spectrum availability depends on the location of TV transmitters and the number of operating stations. For an area with a smaller scale, spectrum availability depends on obstructions, construction material, and the existence of active wireless microphones with typical transmission ranges of a few hundred meters. As reported in [4], for the UHF spectrum in the tested area, the median number of channels available at one point but unavailable at another is close to 7.

Such a spectrum heterogeneity imposes a lot of challenges for resource allocation in CRAHNs. In [5, 6], a cooperative transmission scheme is proposed to address the spectrum heterogeneity issue in the infrastructure mode

network with end user nodes served by a single access point node. With such a scheme, some end user nodes will relay the data traffic of other nodes based on the joint physical layer and MAC layer design. A centralized solution is proposed to address the new resource allocation problem with both relay selection and channel allocation. To demonstrate the feasibility and performance of cooperative relay for the cognitive radio infrastructure mode network, a new MAC protocol has been proposed and implemented in a testbed based on Universal Software Radio Peripheral (USRP) [7] and GNU Radio [8]. Experimental results show that the throughput of the whole system is greatly increased by exploiting the benefit of cooperative relay.

The exiting work mentioned above targets at relatively simple network structures, that is, a network with a single base station and its served end user nodes. There is little existing work on the general CRAHNs leveraging the above cooperative transmission scheme, which is the focus in this paper. In this paper, we study a general network model with multiple single-hop secondary transmission pairs where each node could act as a relay node for its neighbouring transmission pairs. Such a system model represents a typical application scenario. However, it complicates the problem due to the coupling of relay selection and channel allocation. To make the problem tractable, we impose some cooperation constraints and define two types of transmission groups with either one direct transmission pair or two cooperative transmission pairs. We propose to use utility-spectrum ratio as the metric to evaluate the transmission group. Based on this metric, an iterative algorithm is proposed to conduct relay selection and channel allocation.

The rest of the paper is structured as follows. Section 2 provides background knowledge of cooperative transmission scheme used in this paper and related works. Section 3 describes the system model and provides the problem statement. In Section 4 we propose our algorithm to solve the resource allocation problem. Section 5 presents the simulation results. Finally, Section 6 concludes the paper.

2. Background and Related Works

In this section, we provide a brief introduction to the cooperative transmission scheme in cognitive radio networks and summarize the related works.

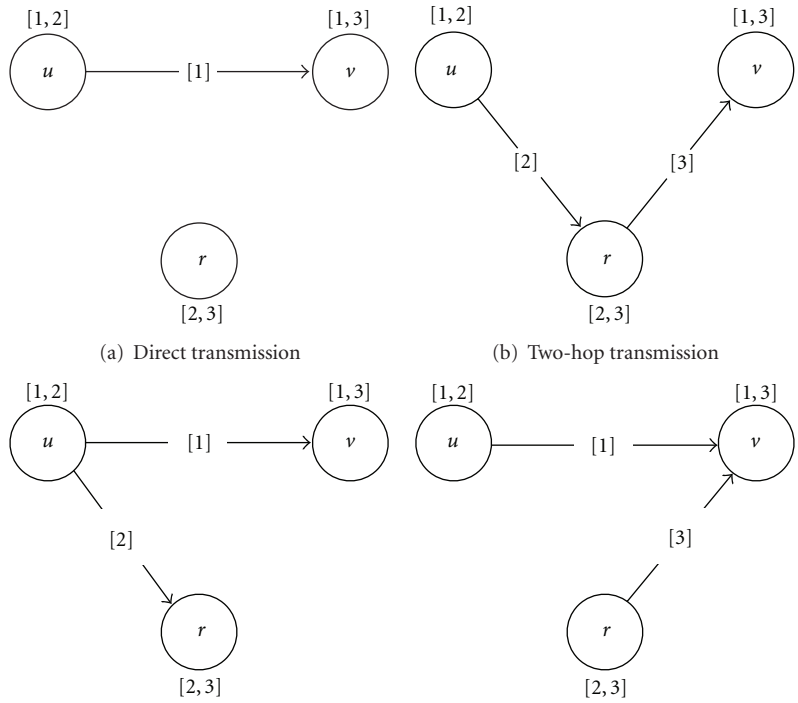
2.1. Improve Spectrum Utilization with Cooperative Relay. The spectrum availability of secondary users is heterogeneous due to the location difference among different users, the dynamic traffic of primary users, and the opportunistic nature of the spectrum access of secondary users. Meanwhile, the traffic demands of secondary users also demonstrate variation. One important problem in cognitive radio networks is to handle the unbalanced spectrum usage within the secondary network to fulfill the heterogeneous traffic demand from secondary users. The observation is that some secondary users can be utilized as helpers to relay the other secondary users traffic, which can significantly improve system performance.

We will use an example to illustrate the idea. Suppose there is a data transmission request from node u to node v as shown in Figure 1. Here the numbers besides the node represent the available channels for that node. Without loss of generality, we assume each common available channel between any pair of nodes can provide data rate of 1 unit. Note that each node has only one half-duplex radio for data transmission. In Figure 1(a), we use direct transmission between u and v on channel 1, which will result in data rate of 1 unit. Alternatively, we can use relay node r to conduct two-hop transmission, with r switching its single data radio between channel 2 (with u) and channel 3 (with v). However, the data rate is only 0.5 unit. Improvement is possible if we introduce cooperative relay. The scheme is shown in Figure 1(c). In time slot 1, u sends data on channel 1 to v, while u also sends data on channel 2 to r. In time slot 2, u sends data on channel 1 to v, while r sends data on channel 3 to v. Therefore, the total data rate increases to 1.5 unit.

2.2. Relay-Assisted Discontiguous OFDM. There exist challenges for the realization of the three-node cooperation technique. First, the sender must be able to transmit multiple packets on multiple channels at the same time using single-radio equipment. For this, we can adopt D-OFDM as the physical-layer technique, where signals on multiple channels can be transmitted simultaneously on single-radio equipment. Second, both relay and receiver should be able to alleviate the interference from other simultaneous transmitting channels to achieve a higher signal-to-noise ratio (SNR) on the specific channel. Third, relay and receiver should be able to decode the packet correctly using only some of all subcarriers that correspond to their working channel. We address these challenges with the following two methods in [5, 6].

We design a new transmission scheme based on discontiguous OFDM to realize the above three-node transmission, which is called relay-assisted D-OFDM. Such scheme can allow a node with a single radio to transmit or receive from multiple nodes on different channels. It addresses several issues such as both relay and receiver should be able to alleviate the interference from other simultaneous transmitting channels to achieve a higher SNR on the specific channel and should be able to decode the packet correctly using only part of the whole subcarriers that are corresponding to their working channel. To demonstrate its feasibility and performance, we implement it in a testbed consisting of USRP and GNU Radio. Experimental results confirm significant gain compared with traditional transmission.

2.3. Existing Works on Cooperative Communication and Cognitive Radio. In recent years, relay nodes have been widely used in various types of wireless networks. In multihop ad hoc networks, relay nodes are used to connect distant nodes that are otherwise disconnected. Meanwhile, relay nodes can increase the transmission rate of each link and maximize the spatial reusability. In cellular networks, relay nodes are used to increase reliability as well as enlarge the coverage range [9].



(a) Direct transmission

(b) Two-hop transmission

(c) Cooperative relay transmission (left in time slot 1, right in time slot 2)

Figure 1: An example of using cooperative relay for cognitive radio networks.

Recently, cooperative communication [10, 11] has also been extensively studied, where relay nodes are introduced to enable single antenna nodes share their antennas to form a virtual multiple-antenna transmitter, thus, transmit diversity is achieved and network capacity is increased. However, the relay nodes in this paper play a different role compared with that in all the above scenarios. Instead of maximizing transmission rate in multihop ad hoc networks, we focus on the maximized spectrum utilization and use relay to bridge the channel availability from the source to the destination node. Besides, by allowing simultaneous transmission of different channels, the throughput of the whole network is increased.

Several existing works investigate the spectrum heterogeneity issue in cognitive radio networks. Based on the design of cooperation relay in infrastructure-mode cognitive radio networks in [5, 6], the work in [12] proposes the routing protocols to improve end-to-end performance with a new link cost, which considers several aspects including channel availability, channel condition, channel utilization, and potential relays. A different system model is considered in [13], where there is a cognitive wireless relay network consisting of a source node that intends to communicate with a destination node aided by a number of secondary relay nodes. To exploit the maximum spectrum opportunities, a cognitive space-time-frequency coding technique is proposed that can opportunistically adjust its coding structure by adapting itself to the dynamic spectrum environment.

There are existing works on cooperative communication in cognitive radio networks. While cooperative communication can be applied within secondary users networks, some works propose schemes for the cooperation between primary users and secondary users. In [14], the authors propose a cooperation protocol in which multiple secondary users can use the spectrum from a primary user in exchange for cooperative transmission with the primary link. The cooperation has three phases: in the first phase, the primary transmitter transmits first, while secondary users receive; in the second phase, the secondary users use the space-time coded cooperative transmission to reply the received data to the primary receiver. At last, the secondary users conduct their own transmission. Following such a framework, the payment is also considered in [15] so that the cooperation opportunity is further enlarged. In [16], a two-phase cooperative relaying protocol is proposed for a primary transmission pair and a secondary transmission pair. In the first phase, the primary transmitter transmits its signal to the primary receiver, which is also received by the secondary transmitter and the secondary receiver and decoded. At the secondary receiver, the primary signal is regenerated and linearly combined with the secondary signal with appropriate power allocation. This combined signal is then broadcasted by the secondary transmitter in the second transmission phase. Different from these works, our cooperation scheme is within the secondary user ad hoc network and leverages the spectrum diversity.

3. System Model and Problem Statement

We consider a CRAHN. Primary users are located within the same region and have low spectrum utilization of their own spectrum. Based on spectrum usage policy such as

spectrum leasing, the unused primary spectrum channels can be temporarily used by the secondary network. In this paper, we assume each secondary node has one radio for data transmission and one radio for control messages, both of which are half-duplex. We assume there is a common control channel available for all secondary nodes. Some existing works of cognitive radio networks have the similar system model as ours [17–19].

There are M adjacent channels from primary users $\mathbf{M} = \{1, \ldots, M\}$ with equal bandwidth W. There are N secondary transmission pairs $\mathbf{N} = \{1, \ldots, N\}$, which correspond to the set of senders $\mathbf{S}$ and targeted receiver set $\mathbf{T}$. We define the node set $\mathbf{V} = \mathbf{S} \cup \mathbf{T}$, with $|\mathbf{V}| = 2N$.

For a particular location in the area, some channels may be occupied by primary users and thus cannot be used by secondary users. We use $\mathbf{A} = \{a_v^m \mid a_v^m \in \{0,1\}\}_{2N\times M}$ to denote the channel availability: $a_v^m = 1$ indicates that channel m at node $v \in \mathbf{V}$ is available, and 0 otherwise. It is assumed that the data radio of each node in $\mathbf{V}$ is capable of dynamically accessing any combination of available channels. However, the data radio cannot transmit and receive simultaneously.

Two nodes can form a communication link if and only if both nodes have common available channels and they are within each other communication range. The communication ranges of all nodes in all channels are the same, denoted by R_C. Similarly, the interference range is denoted by R_I, with $R_I \geq R_C$. Let $\mathbf{E} = \{e_{vu} \mid e_{vu} \in \{0,1\}\}_{2N\times 2N}$ denote the set of potential communication links, that is, $e_{v,u} = 1$ if and only if $\delta(v,u) < R_C$, where $\delta(v,u)$ is the distance between the two nodes. Similarly, let $\mathbf{I} = \{f_{vu} \mid f_{vu} \in \{0,1\}\}_{2N\times 2N}$ denote the set of interference/conflict relations between any two nodes in the network: $f_{vu} = 1$ if and only if $\delta(v,u) < R_I$. The interference set of node v is denoted by $\mathbf{N}_v = \{u \mid f_{vu} = 1, u \in \mathbf{V}\}$.

If a link uses a channel for transmission, it can achieve a certain data rate determined by the transmission power and channel condition. Since interference-free channel allocation is considered, there is no interference from the other active links. We use c_{vu}^m to denote the achievable data rate from node v to u using channel k, which is a constant.

3.1. Relay Selection. We use $\mathbf{R}^s = \{r_{ij}^s \mid r_{ij}^s \in \{0,1\}\}_{N\times N}$ to express the relay selection for the source nodes, where $r_{ij}^s = 1$ means that source node s_j of pair j performs as a relay node for pair i. Similarly, we use $\mathbf{R}^t = \{r_{ij}^t \mid r_{ij}^t \in \{0,1\}\}_{N\times N}$ to express the relay selection for the destination nodes. In this paper, during one cooperative operation we impose the following constraints.

(i) Each transmission pair can use at most one relay node for help:

$$\sum_{j\in\mathbf{N}, j\neq i} \left(r_{ij}^s + r_{ij}^t\right) \leq 1, \quad \forall i \in \mathbf{N}. \tag{1}$$

(ii) One relay pair can help at most one transmission pair:

$$\sum_{i\in\mathbf{N}, i\neq j} \left(r_{ij}^s + r_{ij}^t\right) \leq 1, \quad \forall j \in \mathbf{N}. \tag{2}$$

(iii) When one pair is served by another pair, this node cannot be the relay node of another node:

$$\begin{aligned} r_{ij}^s + \sum_{k\neq i} \left(r_{ki}^s + r_{ki}^t\right) &\leq 1, \quad \forall i \in \mathbf{N},\ j \in \mathbf{N}, \\ r_{ij}^t + \sum_{k\neq i} \left(r_{ki}^s + r_{ki}^t\right) &\leq 1, \quad \forall i \in \mathbf{N},\ j \in \mathbf{N}. \end{aligned} \tag{3}$$

(iv) When one node is serving another pair, the pair of this node cannot be helped by other nodes:

$$\begin{aligned} r_{ij}^s + \sum_{k\neq j} \left(r_{jk}^s + r_{jk}^t\right) &\leq 1, \quad \forall i \in \mathbf{N},\ j \in \mathbf{N}, \\ r_{ij}^t + \sum_{k\neq j} \left(r_{jk}^s + r_{jk}^t\right) &\leq 1, \quad \forall i \in \mathbf{N},\ j \in \mathbf{N}. \end{aligned} \tag{4}$$

According to the above relay constraints (1)–(4), we define a *transmission group* to be a basic component with a single direction transmission pair or two cooperating pairs. We have the following categories of transmission groups as shown in Figure 2:

(i) Category 1 has a single transmission pair,

(ii) Category 2 has two transmission pairs with one pair relaying for another pair.

3.2. Channel Allocation. Besides the relay selection, we need to decide the channel allocation for each transmission link. We use variable $\mathbf{X} = \{x_{vu}^m \mid x_{vu}^m \in \{0,1\}\}_{2N\times 2N\times M}$ to denote the channel allocation: $x_{vu}^m = 1$ if and only if channel m is allocated to the link between node v and node u. Note that we have $x_{vu}^m = x_{uv}^m$. According to the above link definition, we have

$$x_{vu}^m \leq e_{uv}, \quad \forall v \in \mathbf{V},\ \forall u \in \mathbf{V}. \tag{5}$$

The channel allocation should satisfy the channel availability constraint, that is,

$$x_{vu}^m \leq a_v^m \cdot a_u^m, \quad \forall v \in \mathbf{V},\ \forall u \in \mathbf{V},\ \forall m \in \mathbf{M}. \tag{6}$$

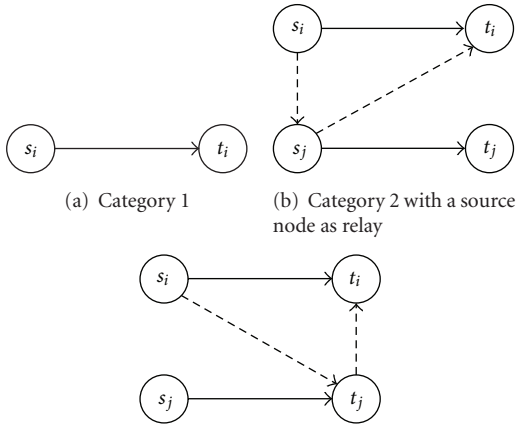
We use conflict-free channel allocation, which requires that all the interference constraints are satisfied, that is,

$$\begin{aligned} x_{vu}^m + \sum_{w\neq v, w\neq u, w\in\mathbf{N}_v} x_{vw}^m + \sum_{w\neq v, w\neq u, w\in\mathbf{N}_u} x_{uw}^m \\ \leq 1, \quad \forall v \in \mathbf{V},\ \forall u \in \mathbf{V},\ \forall m \in \mathbf{M}. \end{aligned} \tag{7}$$

3.3. Data Rate for Each Transmission Pair. We calculate the throughput of pair i under a certain strategy of relay selection $\mathbf{R}$ and channel allocation $\mathbf{X}$. According to whether i helps other pairs or is helped by others, the calculation is divided into the following cases.

Case 1. If pair i communicates without the other nodes help and is not acting as relay itself, which satisfies

$$r_{ij}^s = 0, \quad r_{ij}^t = 0, \quad r_{ji}^s = 0, \quad r_{ji}^t = 0, \quad \forall j \in \mathbf{N}, \tag{8}$$



(a) Category 1 (b) Category 2 with a source node as relay

(c) Category 2 with a destination node as relay

Figure 2: Categories of transmission groups.

its data rate can be expressed as

$$\theta_i^1 = \sum_{m\in\mathbf{M}} c_{s_it_i}^m x_{s_it_i}^m. \tag{9}$$

Case 2. If pair i is helped by the source node of pair j, that is,

$$r_{ij}^s = 1, \tag{10}$$

then the data rate of pair i is

$$\theta_i^2 = \sum_{m\in\mathbf{M}} c_{s_it_i}^m x_{s_it_i}^m + \min\left(\frac{1}{2}\sum_{m\in\mathbf{M}} c_{s_is_j}^m x_{s_is_j}^m, \frac{1}{2}\sum_{m\in\mathbf{M}} c_{s_jt_i}^m x_{s_jt_i}^m\right). \tag{11}$$

Case 3. If pair i is helped with the help of the destination node of pair j, that is,

$$r_{ij}^t = 1, \tag{12}$$

and then the data rate of pair i is

$$\theta_i^3 = \sum_{m\in\mathbf{M}} c_{s_it_i}^m x_{s_it_i}^m + \min\left(\frac{1}{2}\sum_{m\in\mathbf{M}} c_{s_it_j}^m x_{s_it_j}^m, \frac{1}{2}\sum_{m\in\mathbf{M}} c_{t_jt_i}^m x_{t_jt_i}^m\right). \tag{13}$$

Case 4. If the source node s_i of pair i acts as a relay for pair j, but the destination node c_i does not help, that is

$$r_{ji}^s = 1, \qquad r_{ji}^t = 0, \tag{14}$$

the data rate of pair i is then

$$\theta_i^4 = \frac{1}{2}\sum_{m\in\mathbf{M}} c_{s_it_i}^m x_{s_it_i}^m. \tag{15}$$

Case 5. Similar to Case 4, here the destination node t_i acts as relay, but the source node s_i does not, that is,

$$r_{ji}^s = 0, \qquad r_{ji}^t = 1, \tag{16}$$

the data rate of pair i is then

$$\theta_i^5 = \frac{1}{2}\sum_{m\in\mathbf{M}} c_{s_it_i}^m x_{s_it_i}^m. \tag{17}$$

We can use a single equation to denote each pair's data rate based on (9)–(17):

$$\Theta_i = \left(1-\sum_{j\in\mathbf{N}} r_{ij}^s\right)\left(1-\sum_{j\in\mathbf{N}} r_{ij}^s\right)\left(1-\sum_{j\in\mathbf{N}} r_{ij}^s\right)\left(1-\sum_{j\in\mathbf{N}} r_{ij}^s\right)\theta_i^1 + \sum_{j\in\mathbf{N}} r_{ij}^s\theta_i^2 + \sum_{j\in\mathbf{N}} r_{ij}^t\theta_i^3 + \sum_{j\in\mathbf{N}} r_{ji}^s\left(1-r_{ji}^t\right)\theta_i^4 + \sum_{j\in\mathbf{N}}\left(1-r_{ji}^s\right)r_{ji}^t\theta_i^5. \tag{18}$$

3.4. Optimization Problem. We consider that the channel availability is fixed for a relatively long term. The network utility is defined as $U(\boldsymbol{\Theta})$, where $\boldsymbol{\Theta} = \{\Theta_i\}_N$. Under such a quasistatic model, we want to maximize the network utility by optimizing over relay selection, channel allocation, and time partition, subject to the above constraints:

$$\max_{\mathbf{R},\mathbf{X}} U(\boldsymbol{\Theta}). \tag{19}$$

In this paper, we consider two typical formats of utility functions:

(i) Max-Sum: it maximizes the total utility of the network with

$$U_{\text{sum}}(\boldsymbol{\Theta}) = \sum_{i\in\mathbf{N}} \Theta_i. \tag{20}$$

(ii) Max-Fair: it maximizes the proportional fairness of the network with

$$U_{\text{fair}}(\boldsymbol{\Theta}) = \sum_{i\in\mathbf{N}} \log(\Theta_i). \tag{21}$$

Theorem 1. *The optimization problem in* (19) *is NP-hard.*

Proof. To prove this, we can check special cases where there is one single channel and there exist, only interference among pairs, but no communication links among pairs. Then, the problem is a weighted independent set problem, which is an NP-hard problem. □

4. Resource Allocation Algorithms

In this section, we propose a heuristic algorithm to solve the problem suboptimally. The algorithm iterates with multiple times. During each iteration, a transmission group is chosen for resource allocation. We first describe the overall algorithm. Then, we present the proposed metric for transmission group selection.

4.1. The Overall Algorithm. For the overall algorithm, we maintain a list of all possible transmission groups belonging to different categories, denoted as **G**. During a single iteration, we select the transmission group g with certain metric and allocate the channels accordingly. Such a selection metric will be presented in detail in the following part. After that, the group list **G** is updated: transmission groups whose nodes overlap with the selected group g will be deleted from the list. The channel availability matrix and group matrix are also updated.

4.2. Selection Metric of Transmission Group. To effectively utilize the spectrum resource, when we choose which transmission group to allocate in one iteration, we should consider both the achievable utility of a transmission group and the consumed spectrum resource.

We denote the selection metric of transmission group as μ_g, which is the *utility-spectrum ratio* of a transmission group for a particular channel allocation for this transmission group $\mathbf{X}_g$. $\mathbf{X}_g$ is a maximal channel allocation without violating the constraints, which will be described later. Specifically, given $\mathbf{X}_g$, we denote the utility of a transmission group by $U_g(\mathbf{X}_g)$, which can be calculated with (20) or (21). We denote the spectrum consumption of a transmission group by $h_g(\mathbf{X}_g)$, which is the number of occupied/interfered channels of the transmission group. Then, we have

$$\mu_g = \frac{U_g\left(\mathbf{X}_g\right)}{h_g\left(\mathbf{X}_g\right)}. \tag{22}$$

This is similar to the labelling rules suggested in [20, 21]. The difference is that we assess and allocate multiple channels for a node at once.

Depending on the category of a transmission group g, we determine the value of $\mathbf{X}_g$ and calculate μ_g as follows (Algorithm 1).

4.2.1. Category 1. Suppose the single pair is i for the transmission group g. For each channel, we check whether it is available at both source node s_i and destination node t_i in group g and allocate it if it is the case, that is,

$$x^m_{s_i t_i} = a^m_{s_i} \cdot a^m_{t_i}, \quad \forall m \in \mathbf{M}. \tag{23}$$

The contributed utility of this group can be easily calculated with (20) or (21):

$$U_g = \theta^1_i, \tag{24}$$

or

$$U_g = \log \theta^1_i. \tag{25}$$

The spectrum consumption is

$$h_g = \sum_{m\in\mathbf{M}} \left(x^m_{s_i t_i} \sum_{v\in\mathbf{N}_{s_i}\cup\mathbf{N}_{t_i}} \mathbf{1} \right). \tag{26}$$

Input: Transmission group g
Output: Utility-spectrum ratio of g, μ_g
If *g is Category* 1 **then**
  $i \leftarrow$ the single pair
  Allocate channels $\mathbf{X}_g : x^m_{s_i t_i} = a^m_{s_i} \cdot a^m_{t_i}, \forall m \in \mathbf{M}$
  Calculate $U_g(\mathbf{X}_g) = U_{\text{sum}}(\mathbf{X}_g)$
  Calculate $h_g = \sum_{m\in\mathbf{M}} (x^m_{s_i t_i} \sum_{v\in\mathbf{N}_{s_i}\cup\mathbf{N}_{t_i}} \mathbf{1})$
else
  $i \leftarrow$ the helped pair
  $j \leftarrow$ the relay pair
  $P \leftarrow \{(m,n) \mid m \in \mathbf{M}, n \in \mathbf{M}, m \neq n, \exists a^m_s = 1, \exists a^m_v = 1, \exists a^n_v = 1, \exists a^n_t = 1\}$
  while $P \neq \varnothing$ **do**
    $(m^*, n^*) \leftarrow \arg\max_{(m,n)\in P} \Delta^r_{(m,n)}$
    if $\Delta^r_{(m^*,n^*)} > \Delta^d_{(m^*,n^*)}$ **then**
      Allocate channels: $x^{m^*}_{sv} = 1, x^{n^*}_{vt} = 1$
      Update P
    else
      Break
    end
  end
  $M_r \leftarrow$ remaining channels
  for each $m \in M_r$ **do**
    if $c^m_{st} > 1/2c^m_{vu}$ **then**
      Allocate channel: $x^m_{st} = 1$
    else
      Allocate channel: $x^m_{vu} = 1$
    end
  end
  Calculate $U_g(\mathbf{X}_g) = U_{\text{fair}}(\mathbf{X}_g)$
  Calculate $h_g = \sum_{m\in\mathbf{M}} (x^m_{v_m u_m} \sum_{v\in\mathbf{N}_{s_i}\cup\mathbf{N}_{t_i}} \mathbf{1})$.
end
$\mu_g \leftarrow U_g/h_g$

Algorithm 1: Transmission group selection metric calculation algorithm.

4.2.2. Category 2. It depends on whether the source node or destination node of pair j is used for cooperation. In each case, one channel can be used for at most one active link among nodes s_i, t_i, s_j, and t_j due to the constraint. Although there are only four links for the transmission group, the number of all possible channel allocations is with the order of 4^M.

To reduce the computational complexity, we propose a heuristic approach to calculate the utility of the transmission group. The idea is that there is positive contribution via relay links for the transmission group if and only if both of the relay links (e.g., s_i-v_j and v_j-t_i) are allocated with channels. Therefore, we assign pairs of channels to relay links first when calculating the utility. Totally, there are at most $M \times (M-1)$ pairs of channels.

Specifically, we define the contributed data rate of a channel pair (m,n) with $m \neq n$ as $\Delta^r_{(m,n)}$, which is the data

rate increase with channel m and n allocated for the two relay links, that is,

$$\Delta^r_{(m,n)} = \min\left(r1 + \frac{1}{2}c^m_{s_i v_j} r2 + \frac{1}{2}c^n_{v_j t_i}\right) - \min(r1, r2), \quad (27)$$

where $r1$ and $r2$ are the data rate of link s_i-v_j and v_j-t_i, respectively. For each time, we find the channel pair with the maximum $\Delta^r_{(m,n)}$ for the transmission group.

We also examine the contributed data rate when channel m and n are allocated to the two direct links. We have 4 combinations: channel m and n both assigned to one direct link; channel m and n assigned to different links. We denote the maximum contributed data rate of these four possible allocations as $\Delta^d_{(m,n)}$.

If $\Delta^r_{(m,n)}$ is larger than $\Delta^d_{(m,n)}$, we allocate two channels to the two relay links. Otherwise, channel-pair-based allocation terminates. If there are any remaining channels unallocated, we continue to allocate them just as Category 1.

For each channel allocation, if the source node is the relay node, we have

$$U_g = \theta^2_i + \theta^4_j, \quad (28)$$

or

$$U_g = \log\theta^2_i + \log\theta^4_j. \quad (29)$$

If the destination node is the relay node, we have

$$U_g = \theta^3_i + \theta^5_j, \quad (30)$$

or

$$U_g = \log\theta^3_i + \log\theta^5_j. \quad (31)$$

Denote the nodes for the active link for channel m as v_m and u_m. The spectrum consumption is

$$h_g = \sum_{m\in\mathbf{M}}\left(x^m_{v_m u_m} \sum_{v\in\mathbf{N}_{s_i}\cup\mathbf{N}_{t_i}} \mathbf{1}\right). \quad (32)$$

For the complexity of the metric calculation algorithm (Algorithm 1), obviously the calculation of Category 1 is dominated by the calculation of Category 2. For Category 2, the complexity is with the order of $O(M^2(M^2+N))$, which is the complexity for the whole metric calculation algorithm.

5. Simulation

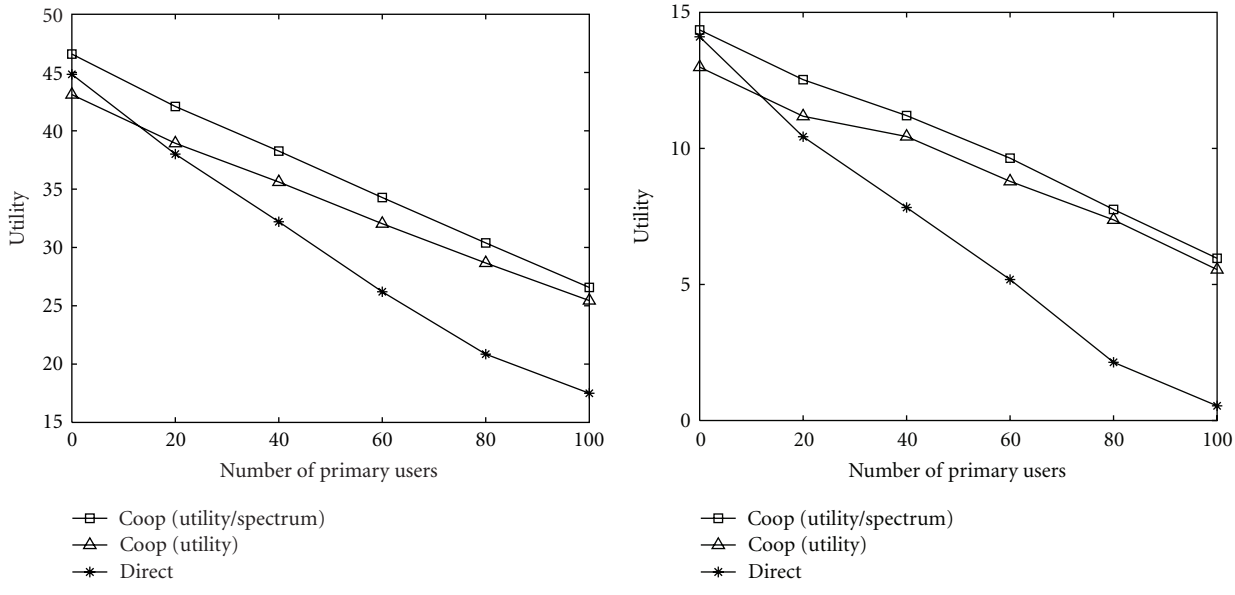
In this section, we use simulations to evaluate the performance of the proposed algorithm. We conduct the simulation in a static CRAHN. The simulation area is 1×1 with uniformly separated primary users in fixed locations. Each primary user occupies a single channel randomly picked from the channel set. Secondary users pairs are randomly placed in the same area. We compare our proposed cooperative transmission and resource allocation scheme with two other schemes. One scheme uses the same cooperative transmission with relaying, while the metric for transmission group selection during resource allocation depends solely on the achieved utilities of transmission groups. The other scheme uses the direct transmission without relaying and applies similar resource allocation as our scheme. We investigate the performance of the CRAHN in terms of the number of primary users, the number of secondary users, the number of channels, and the communication range of secondary users.

5.1. The Number of Primary Users. We first show the performance with different numbers of primary users. When we increase the number of primary users, the protected area expands, which reduces the number of available channels experienced by secondary users. In Figure 3, the utilities for both Max-Sum and Max-Fair decrease with increasing the number of primary users. The cooperative scheme with the metric of utility-spectrum ratio outperforms the other two schemes since our scheme allows secondary pairs to utilize more transmission links via potential relay nodes from neighbouring nodes on more channels while our resource allocation algorithm selects transmission groups and their channel allocation more efficiently. Besides, the gaps between the two cooperative schemes and the direct scheme increase as the number of primary users increases. This is because with more active primary users the degree of spectrum heterogeneity seen from secondary users increases.

5.2. The Number of Secondary Users. We check the performance with different numbers of secondary users next. Increasing the number of secondary users leads to the increased total system utility, as shown in Figure 4. Besides, the performance gaps between the two cooperative schemes and the direct scheme increase as the number of secondary users increases, which demonstrates the advantage of our scheme. Note that the curves are sublinear since more secondary users means more interference created among them.

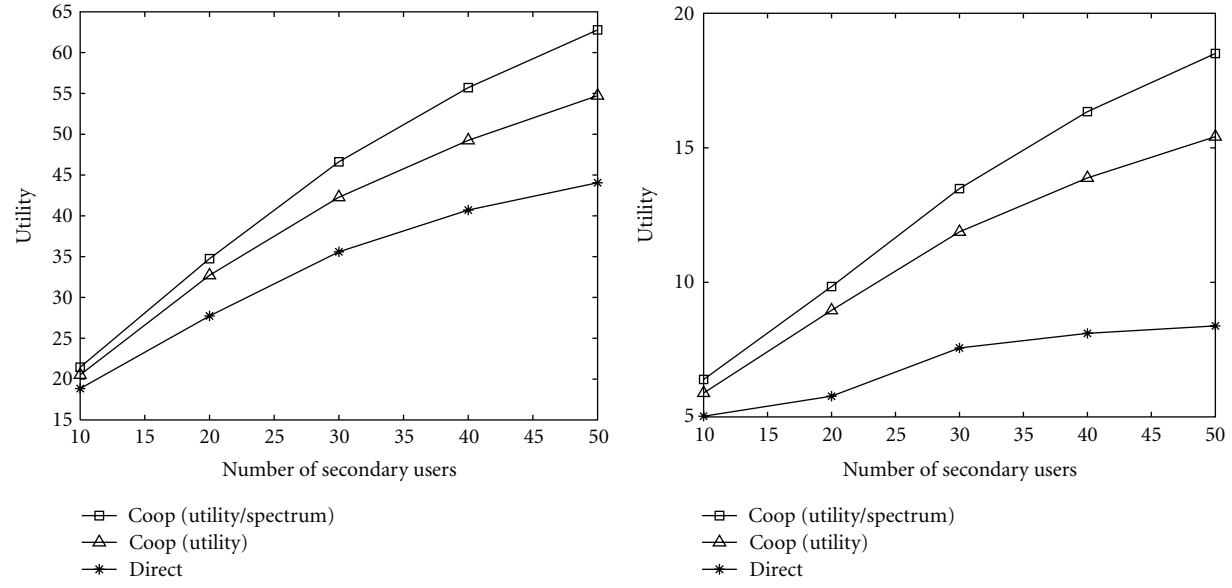
5.3. The Number of Channels. Here we examine the effect of the number of channels used in the system. Figure 5 shows that both types of utilities increase while increasing the number of channels. The cooperative scheme with the metric of utility-spectrum ratio outperforms the other two schemes, while the direct scheme performs the worst.

5.4. The Communication Range of Secondary Users. We also study the impact of the communication range of secondary users. On one hand, given a fixed topology, enlarging the communication range of secondary users increases the number of potential communication links among them, which creates more cooperation opportunities. On the other hand, the increased range also creates more interference among the secondary users, which will decrease the spatial reuse of the spectrum resource. According to the result in Figure 6, the system utilities degrade with the increased communication range. Therefore, the latter effect dominates the first one.



(a) Sum (b) Fairness

Figure 3: Performance with respect to the number of primary users.

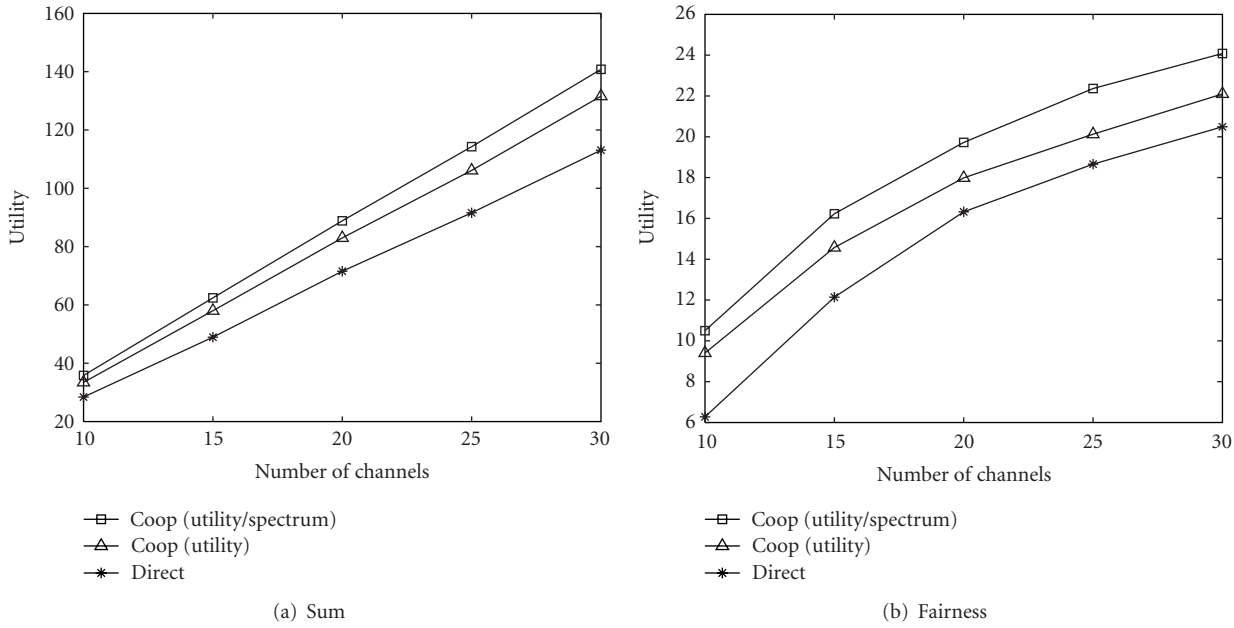


(a) Sum (b) Fairness

Figure 4: Performance with respect to the number of secondary users.

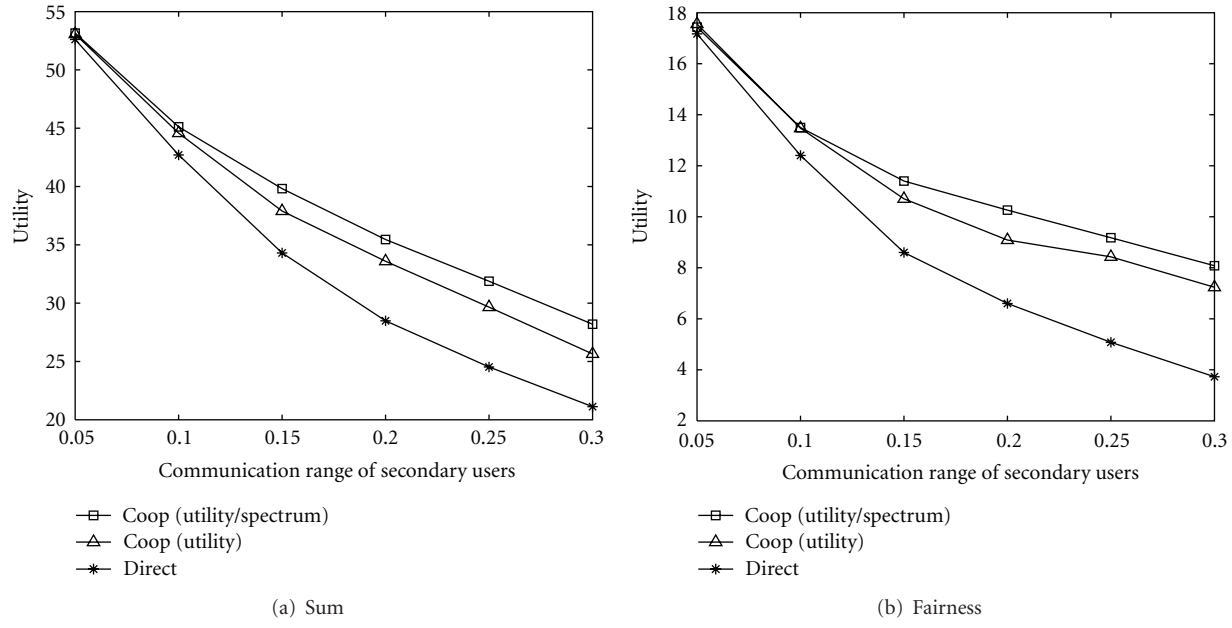
6. Conclusions

In CRAHNs, spectrum heterogeneity is quite a special issue that makes the resource allocation more challenging compared with the traditional wireless ad hoc networks. It is possible to use cooperative relay node utilizing spectrum holes to assist individual secondary transmission and improve link throughput. Based on this idea, we study a new resource allocation problem with relay selection and channel allocation in CRAHNs when applying the cooperative relay scheme. We propose algorithms to effectively solve the problem based on a metric of utility-spectrum ratio. We conduct simulations to evaluate the performance and conduct comparison with direct transmission scheme. The simulation results demonstrate the reasonable performance improvements of our scheme. In the future, we will further investigate the distributed algorithms for the relay selection and channel allocation in CRAHNs with cooperative relaying.



(a) Sum (b) Fairness

Figure 5: Performance with respect to the number of channels.



(a) Sum (b) Fairness

Figure 6: Performance with respect to secondary users communication range.

Acknowledgments

This work is supported in part by Starting Research Fund from Soochow University under Grant no. 14317436, National Natural Science Foundation of China under Grant no. 61070169, Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, Specialized Research Foundation for the Doctoral Program of Higher Education of China under Grant no. 20103201110018, and Application Foundation Research of Suzhou of China under Grant no. SYG201118.

References

[1] Federal Communications Commission et al., "Spectrum policy task force," Report ET Docket (02-135):1, 2002.

[2] J. Mitola III and G. Q. Maguire Jr., "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, 1999.

[3] I. F. Akyildiz, W. Y. Lee, and K. R. Chowdhury, "CRAHNs: cognitive radio ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 810–836, 2009.

[4] S. Deb, V. Srinivasan, and R. Maheshwari, "Dynamic spectrum access in DTV whitespaces: design rules, architecture and

algorithms," in *Proceedings of the15th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '09)*, pp. 1–12, September 2009.

[5] Q. Zhang, J. Jia, and J. Zhang, "Cooperative relay to improve diversity in cognitive radio networks," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 111–117, 2009.

[6] J. Jia, J. Zhang, and Q. Zhang, "Cooperative relay for cognitive radio networks," in *Proceedings of th 28th Conference on Computer Communications (IEEE INFOCOM '09)*, pp. 2304–2312, April 2009.

[7] M. Ettus, "Universal software radio peripheral (USRP)," *Ettus Research LLC*; http://www.ettus.com/.

[8] GNU Radio, "The gnu software radio," *World Wide Web*; http://gnuradio.org.

[9] Y. D. Lin and Y. C. Hsu, "Multihop cellular: a new architecture for wireless communications," in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies IEEE INFOCOM*, vol. 3, pp. 1273–1282, March 2000.

[10] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[11] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Communications Magazine*, vol. 42, no. 10, pp. 74–80, 2004.

[12] J. Jia, J. Zhang, and Q. Zhang, "Relay-assisted routing in cognitive radio networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, June 2009.

[13] K. B. Letaief and W. Zhang, "Cooperative communications for cognitive radio networks," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 878–893, 2009.

[14] O. Simeone, I. Stanojev, S. Savazzi, Y. Bar-Ness, U. Spagnolini, and R. Pickholtz, "Spectrum leasing to cooperating secondary ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 203–213, 2008.

[15] J. Zhang and Q. Zhang, "Stackelberg game for utility-based cooperative cognitiveradio networks," in *Proceedings of the 10th ACM international symposium on Mobile ad hoc networking and computing*, pp. 23–32, ACM, 2009.

[16] Y. Han, A. Pandharipande, and S. H. Ting, "Cooperative decode-and-forward relaying for secondary spectrum access," *IEEE Transactions on Wireless Communications*, vol. 8, no. 10, pp. 4945–4950, 2009.

[17] K. Karakayali, J. H. Kang, M. Kodialam, and K. Balachandran, "Cross-layer optimization for OFDMA-based wireless mesh backhaul networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '07)*, pp. 276–281, March 2007.

[18] T. Chen, H. Zhang, G. M. Maggio, and I. Chlamtac, "CogMesh: a cluster-based cognitive radio network," in *Proceedings of the 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 168–178, April 2007.

[19] K. R. Chowdhury and I. F. Akyildiz, "Cognitive wireless mesh networks with dynamic spectrum access," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 168–181, 2008.

[20] S. Ramanathan, "Unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81–94, 1999.

[21] C. Peng, H. Zheng, and B. Y. Zhao, "Utilization and fairness in spectrum assignment for opportunistic spectrum access," *Mobile Networks and Applications*, vol. 11, no. 4, pp. 555–576, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 785403, 9 pages
doi:10.1155/2012/785403

Research Article

CAC-MAC: A Cross-Layer Adaptive Cooperative MAC for Wireless Ad Hoc Networks

Chunguang Shi, Haitao Zhao, Shan Wang, Jibo Wei, and Linhua Zheng

BCNG, College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Chunguang Shi, c.g.shi@nudt.edu.cn

Received 19 December 2011; Accepted 1 March 2012

Academic Editor: Shukui Zhang

Cooperative communication has been recently proposed as a way to mitigate fading in wireless networks. A cross-layer adaptive cooperative MAC (CAC-MAC) protocol for IEEE 802.11 DCF-based wireless ad hoc networks is proposed. The novel aspect and core idea of our proposal is a cross-layer adaptive data transmission algorithm considering both the length of data frame at the MAC layer and instantaneous wireless channel conditions. Under this algorithm, direct transmission mode or proper cooperative transmission mode will be adaptively selected for data packets according to both MAC layer and physical layer information. Analytical results demonstrate the effectiveness of the adaptive data transmission algorithm. Simulation studies based on NS2 show that the CAC-MAC protocol can significantly improve network throughput and reduce packet delay compared with legacy IEEE 802.11 protocol, which illustrate a new paradigm for realistic cross-layer cooperative MAC protocol design for next-generation wireless ad hoc networks.

1. Introduction

Cooperative communication, which can achieve spatial diversity by exploiting distributed virtual antennas of cooperative nodes, has attracted much attention recently due to its ability to mitigate fading in wireless networks. The main feature of cooperative communication is the involvement of neighboring nodes in data transmissions. As depicted in Figure 1, the source has an inferior channel with destination and meanwhile no less than one neighboring node has a good channel with both the source and the destination. And hence, the source can transmit data packets via neighbor node(s) to the destination at a higher data rate instead of a direct transmission to the destination at a lower data rate.

The studies in [1–3] show that significant benefit is obtained through cooperative communication in terms of reliability, throughput, coverage range, and energy efficiency. Although cooperative communication originates from the physical layer, from the system point of view, in order to realize a fully cooperative network, researches at the physical layer should be coupled with those at the higher network layers, for example, MAC layer. However, so far no standard on cooperative MAC design has been achieved, and hence leave it an open research topic.

The neighboring nodes participating in the cooperative communication are called relay nodes or helpers. The relay nodes can operate on decode-and-forward (DF), amplify-and-forward (AF) or coded cooperation (CC) strategies. For further details readers are referred to [4, 5]. Generally, employing more relay nodes for a given source-destination pair may obtain more cooperative diversity gain, but the resultant lower spectrum efficiency and higher computational complexity may not lead to a beneficial performance-complexity tradeoff [6]. Therefore, this paper focuses on selecting no more than one relay node for each source-destination pair whenever cooperative communication is desirable.

There have been exiting literatures on cooperative MAC design utilizing just one relay node for IEEE 802.11 DCF-based wireless networks. rDCF [7] and CoopMAC [8] are two similar cooperative MAC protocols that take advantage of the multirate capability of the IEEE 802.11 in which high-data-rate nodes assist low-data-rate nodes to transmit data. In these two protocols, each node promiscuously

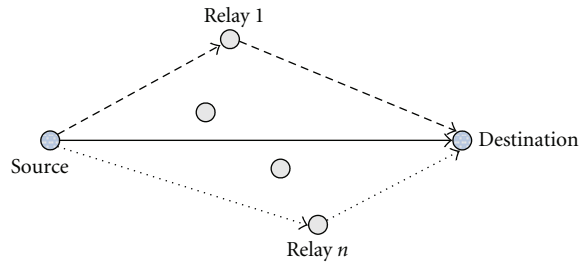


Figure 1: Illustration of cooperative communication.

listens to ongoing transmissions to establish and maintain a relay table. Therefore, the relay selection cannot adapt to dynamic channel conditions and network topology in wireless networks just as it is based on the observation of historical transmissions. Furthermore, in these researches, the cooperative diversity is not exploited and only one copy of a packet coming from the source or the relay node is processed at the destination. Hence, more specifically, rate adaptation is the main focus in data transmissions in these two protocols. Considering that CoopMAC is a cooperative protocol for infrastructure-based wireless LANs, Korakis et al. [9] extend it to the ad hoc network environment. Based on CoopMAC, Liu et al. [10] further propose a cross-layer cooperative protocol for wireless ad hoc networks to leverage cooperation in both MAC and PHY layers where the relay node adopts coded cooperation strategy [5]. Actually, for a cooperative MAC protocol, the data transmission mode should be dynamically selected according to both the length of data frame at the MAC layer and the time-varying channel status between source, relay, and destination, which is not considered in the above-mentioned studies [7–10].

In order to reap more benefit of cooperation, in this paper, a cross-layer adaptive cooperative MAC (CAC-MAC) protocol for IEEE 802.11 DCF-based wireless ad hoc networks is proposed, which forms a cross-layer approach to cooperation involving interaction between MAC layer and physical layer. The novel aspect and core idea of our proposal is a cross-layer adaptive data transmission algorithm considering both the length of data frame at the MAC layer and instantaneous wireless channel conditions. Under this algorithm, direct transmission mode or proper cooperative transmission mode will be adaptively selected for data packets according to both MAC layer and physical layer information. The key features of our proposal are as follows.

First, only when a data frame at the MAC layer is longer than a specified length, CAC-MAC initiates a RTS/CTS handshake, which brings down the overhead of network.

Second, for long data frames, RTS/CTS direct transmission or proper cooperative transmission will be selected according to the wireless channel conditions. Moreover, the cooperative transmission is divided into either "source-relay-destination" transmission scheme or receiver maximal ratio combining scheme according to the channel conditions between source, relay, and destination.

Third, the selection of the best relay node for a given source-destination pair is based on instantaneous wireless channel measurements instead of a relay table, which has the added cost to be created and maintained based on the observation of historical transmissions.

The remainder of this paper is organized as follows. Section 2 introduces the system model. Section 3 proposes a cross-layer adaptive cooperative MAC (CAC-MAC) protocol for IEEE 802.11 DCF-based wireless ad hoc networks. Section 4 analyzes the cross-layer adaptive data transmission algorithm. Section 5 evaluates the performance of CAC-MAC protocol based on NS2, and Section 4 concludes this paper.

2. System Model

We consider a wireless ad hoc network based on IEEE 802.11a that supports transmission rates of 6, 12, 24, and 54 Mbps. A single physical channel is available for wireless transmissions. We assume a slow fading channel that the channel conditions do not change within the duration of a MAC frame transmission. We assume that each node has constant transmission power and that the wireless channels are symmetric. It is also assumed that a relay node works on the decode-and-forward (DF) strategy [4]. The terms relay node and helper are of the same meaning in this paper.

Due to the broadcast nature of the channel, the destination will receive the signals transmitted by both the source and the relay node. Receiver combining technique [11], not supported by any existing wireless hardware, can be implemented in the next-generation wireless baseband chip. Hence, it is reasonable to assume that the destination can adopt maximal ratio combining diversity technique at the physical layer to combine the signals coming from the source and the relay node if the independent copies are in the same modulation scheme, enabling higher transmission rates and robustness against channel variations due to fading.

3. The Proposed CAC-MAC Protocol

In our proposal, each data transmission is based on two planes: control plane and data plane. The control plane is to determine the data transmission mode, in which the main issues include relay selection and the cross-layer adaptive data transmission algorithm. The data plane is in charge of

Input: L, R_{sr}, R_{rd} and R_{sd}
Output: transmission mode (mode) and the source transmission rate (rate)
(1) if $L <$ RTSThreshold
(2) mode = basic access scheme; rate = *6* Mbps
(3) ***end if***
(4) ***if*** $L \geq$ RTSThreshold (to initiate a RTS/CTS handshake)
(5) ***If*** $R_{sd} \geq 24\,Mbps$ || ($R_{sd} \leq 12$ Mbps &&($R_{sr} \leq 12$ Mbps || $R_{rd} \leq 12$ Mbps))
(6) mode = RTS/CTS direct transmission scheme; rate = R_{sd}
(7) ***else if*** $R_{sd} \leq 12$ Mbps && ($R_{sr} \geq 24$ Mbps && $R_{rd} \geq 24$ Mbps)
(8) (to initiate cooperative transmission)
(9) ***if*** $R_{sr} < R_{rd}$
(10) mode = relay transmission scheme; rate = R_{sr}
(11) ***else if*** $R_{sr} \geq R_{rd}$
(12) mode = maximal ratio combining scheme; rate = R_{sr}
(13) ***end if***
(14) ***end if***
(*15*) ***end if***

Algorithm 1: Cross-layer adaptive data transmission algorithm.

transmitting, receiving, or forwarding data packets according to the transmission mode.

3.1. Relay Selection. In CAC-MAC protocol, for long data frames, the RTS/CTS handshake is initiated, so the neighbor nodes can measure the instantaneous channel conditions toward source and destination via overhearing RTS and CTS frames. In addition, the neighbor nodes can extract the channel conditions between source and destination from the extended CTS frame. The extended CTS frame format will be described later in Section 3.3. With the channel quality information, by checking the threshold value, which is precalculated and guarantees a certain bit error rate for each modulation scheme, we can obtain the achievable transmission rate between source and relay, relay and destination, and source and destination, denoted by R_{sr}, R_{rd}, and R_{sd}, respectively.

We assume the length of a data frame is L bytes; if a direct transmission is adopted, the transmission time would be

$$T_{direct} = \frac{8L}{R_{sd}}. \tag{1}$$

If a cooperative transmission via node i is adopted, the transmission time would include two parts: the time consumed between the source and the relay and that consumed between the relay and the destination, namely,

$$T^{i}_{coop} = \frac{8L}{R_{sr}} + \frac{8L}{R_{rd}}. \tag{2}$$

For a neighbor node j, if it satisfies

$$T^{j}_{coop} < T_{direct}, \tag{3}$$

it becomes a candidate relay node. And the candidate relay node r^* that has the minimum transmission time will be the best relay node for the given source-destination pair, that is,

$$r^* = \arg \min_{r \in R} T^{r}_{coop}, \tag{4}$$

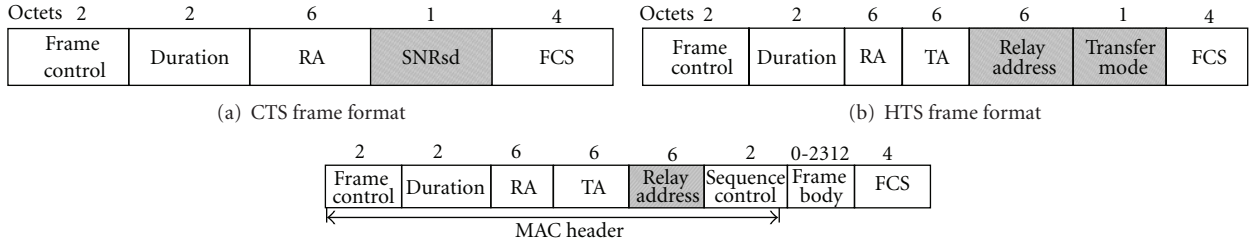
where R is the set of all candidate relay nodes.

In practice, to select the best relay node for a given source-destination pair, each candidate relay node r will start a timer T_r [6] in line with the parameter T^{r}_{coop}, and the longer T^{r}_{coop} means a lager T_r. And hence, the timer T_r of the relay node with the minimum T^{r}_{coop} will expire first. Once the timer expires, the relay node will transmit an HTS (helper ready to send) frame [8] to declare its capacity to participate in the cooperative transmission. All other candidate relay nodes waiting for their timer to expire that overhear the HTS frame from another relay node will back off. After the best relay node has been selected, it is ready to participate in cooperative transmission.

3.2. Cross-Layer Adaptive Data Transmission Algorithm. In CAC-MAC protocol, the data transmission modes are divided into four schemes, that is, basic access scheme, RTS/CTS direct transmission scheme, "source-relay-destination" transmission scheme, and receiver maximal ratio combining scheme, according to both the length of data frame at the MAC layer and the time-varying channel status between source, relay, and destination. The cross-layer adaptive data transmission algorithm is briefly summarized in Algorithm 1.

It is well known that there are two access schemes defined in IEEE 802.11 DCF, namely, the basic access scheme and the RTS/CTS scheme. Similarly, under our proposed algorithm, when a data frame is shorter than RTS threshold, the source transmits it directly to the destination by the basic access scheme. Otherwise, the source will initiate a RTS/CTS handshake. If the source has a good channel with the destination where the sustainable rate is equal to or larger than 24 Mbps, or in the case that the channels between source and destination, source and relay, relay and destination are inferior where all the sustainable rates are equal to or less than 12 Mbps, the source transmits the data frame by the RTS/CTS direct transmission scheme.

On the other hand, if the source has an inferior channel with the destination and meanwhile no less than one relay node has a good channel with both the source and the

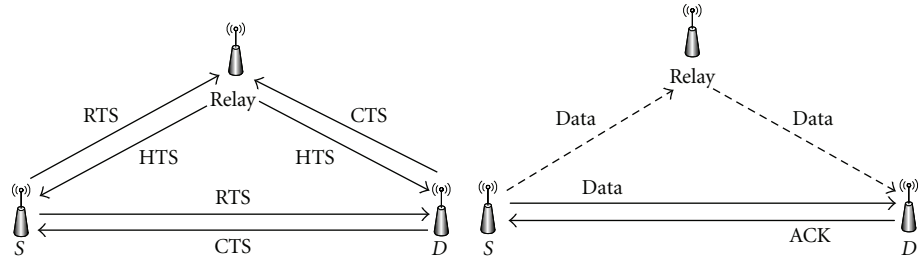


(a) CTS frame format

(b) HTS frame format

(c) DATA frame format

Figure 2: Frames format for CAC-MAC.



(a) Control frames exchange

(b) Data frame exchange

Figure 3: The exchange of messages.

destination, therefore the source can transmit the data packet via the relay node to the destination. Once the source initiates one cooperative transmission via a relay node, it will transmit a data packet at a rate of R_{sr}. When the relay node decodes and forwards the data packet toward the destination at a rate of R_{rd}, the destination will receive two copies of the packet: one from the source with the rate of R_{sr} and the other from the relay node with the rate of R_{rd}.

Since the destination has an inferior channel with the source, it will not be able to solely decode the data packet with the rate of R_{sr}, just as its sustainable rate is R_{sd}($R_{sr} > R_{sd}$, $R_{rd} > R_{sd}$). When $R_{sr} \geq R_{rd}$, if the destination adopts the maximal ratio combining scheme, the relay node should forward the data packet at a higher rate of R'_{rd} ($R'_{rd} = R_{sr} > R_{rd}$) due to the assumption that the independent copies are in the same modulation scheme for the receiver maximal ratio combining technique, which will improve the network performance compared with the "source-relay-destination" transmission scheme. Otherwise, when $R_{sr} < R_{rd}$, if the destination adopts the maximal ratio combing scheme, the relay node should forward the data packet at a lower rate of R''_{rd} ($R''_{rd} = R_{sr} < R_{rd}$), which will sacrifice the original higher transmission rate R_{rd}. So, under the circumstances, the "source-relay-destination" transmission scheme will achieve superior network performance.

3.3. Details of the CAC-MAC Protocol. To support CAC-MAC protocol, some minor modifications to the IEEE 802.11 frames format are required, and meanwhile the RTS/CTS handshake defined in IEEE 802.11 is further extended to an RTS/CTS/HTS handshake. The modified CTS, HTS, and DATA frames format and the exchange of messages are shown in Figures 2 and 3, respectively.

We now give a detailed description of CAC-MAC protocol from the views of source node, destination node, and relay node, respectively, as follows.

Source Node

(1) When the length of a data frame is less than the RTS threshold, the source will transmit it directly to the destination by the basic access scheme of IEEE 802.11 DCF, which brings down the overhead in the network; otherwise, the source will send an RTS frame and wait for a CTS frame from the destination.

(2) If the source receives a CTS frame but does not receive any HTS frame from neighbor nodes in a certain interval, it will transmit the data packet by RTS/CTS direct transmission scheme. If both CTS and HTS frames are received in sequence, the source transmits the data packet according to the "transfer mode" piggybacked in the HTS frame.

(3) If an ACK is not received after an ACK timeout, the source should perform random backoff; otherwise, the source will handle the next data packet in its queue.

Destination Node

(1) If the destination receives an RTS frame from the source, it sends a CTS frame including the measured channel conditions information between source and destination and waits for HTS frames from neighbor nodes.

(2) If any HTS frame is not received before receiving data packet, indicating that the source transmits data packet by RTS/CTS direct transmission scheme, the destination processes the unique data packet.

(3) If the destination receives an HTS frame before receiving data packet, it will process the received data packet according to the "transfer mode" piggybacked in HTS and then sends an ACK to the source.

Neighbor Node

(1) The neighbor node judges whether itself is a candidate relay node for a given source-destination pair according to (1)–(3) in Section 3.1. If it is, it will wait for the timer T_r to expire and then broadcasts an HTS frame to declare itself; if it receives an HTS frame before the timer reaches zero meaning it is not the best relay node for the given source-destination pair, the neighbor node should backoff.

(2) When overhearing a data packet, a candidate relay node extracts the "relay address" information to judge whether it is the relay node for the given source-destination pair. If it is, the node will decode and forward the data packet to the destination.

4. Analysis of Adaptive Data Transmission Algorithm

In this section, we analyze the saturation throughput and average packet delay of the cross-layer adaptive data transmission algorithm based on a Markov chain model, taking finite retry limits into account. For simplicity, it is assumed that there are no hidden nodes or capture effect in the network.

4.1. Markov Chain Model. IEEE 802.11 DCF adopts a binary exponential backoff scheme. At each packet transmission, the backoff time is uniformly chosen in the range $(0, CW - 1)$. The value of CW depends on the number of failed transmissions of a packet. At the first transmission attempt $W = CW_{\min}$, which is the minimum contention window. After each retransmission due to a collision, CW is doubled up to a maximum value, $W_{m'} = CW_{\max} = 2^{m'} \cdot CW_{\min}$, where m' represents the maximum backoff stage and $W_{m'}$ is the largest contention window size. Once the CW reaches $CW_{\max}$, it will remain at the value until it is reset. Therefore, we have:

$$W_i = \begin{cases} 2^i W, & i \le m' \\ 2^{m'} W, & i > m' \end{cases} \tag{5}$$

where i is the backoff stage, $i \in (0, m)$ and m represents the maximum retransmission limits.

Let $b(t)$ be the stochastic process representing the backoff time counter for a given node and $s(t)$ be the stochastic process representing the backoff stage $(0, \ldots, m)$ of the node at time t. So we model the bidimensional process $\{s(t), b(t)\}$ with the discrete-time Markov chain depicted in Figure 4.

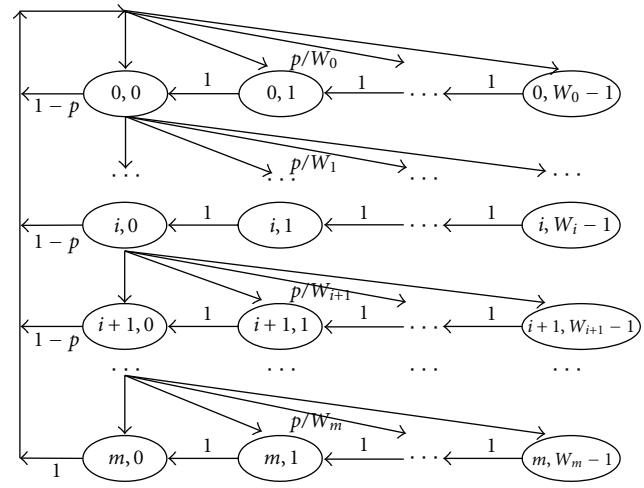


Figure 4: Markov chain model.

Let $b_{i,k} = \lim_{t\to\infty} P\{s(t) = i, b(t) = k\}$, $i \in (0, m)$, $k \in (0, W_i - 1)$ be the stationary distribution of the Markov chain. It is assumed that each packet collides with constant and independent probability p. As any transmission occurs when the backoff time counter reaches zero, the probability τ that a node transmits a packet in a randomly chosen slot time can be expressed as [12, 13]

$$\tau = \sum_{i=0}^{m} b_{i,0} = \sum_{i=0}^{m} p^i \cdot b_{0,0} = b_{0,0} \cdot \frac{1 - p^{m+1}}{1 - p}, \tag{6}$$

where $b_{0,0}$ can be obtained from

$$b_{0,0}^{-1} = \begin{cases} \dfrac{W \cdot \left(1 - (2p)^{m+1}\right)}{2(1-2p)} + \dfrac{1 - p^{m+1}}{2(1-p)}, & m \le m', \\ \dfrac{W \cdot \left(1 - (2p)^{m'+1}\right)}{2(1-2p)} + \dfrac{1 - p^{m+1}}{2(1-p)} \\ \quad + \dfrac{W \cdot 2^{m'} \cdot p^{m'+1} \cdot (1 - p^{m-m'})}{2(1-p)}, & m > m'. \end{cases} \tag{7}$$

From (6), we can see that the transmission probability τ depends on the collision probability p. The probability p that a transmitted packet encounters a collision is the probability that at least one of the $n - 1$ remaining nodes transmit in the same time slot. If all nodes transmits with probability τ, the collision probability p is

$$p = 1 - (1 - \tau)^{n-1}. \tag{8}$$

Therefore, (6) and (8) form a nonlinear system with two unknowns τ and p, which can be solved by the numerical method. Note that $p \in (0, 1)$ and $\tau \in (0, 1)$.

4.2. Saturation Throughput. Let P_{tr} be the probability that there is at least one transmission in the considered slot

time. When n nodes contend on the same channel and each transmits with probability τ:

$$P_{\text{tr}} = 1 - (1-\tau)^n. \tag{9}$$

The probability P_s that an occurring packet transmission is successful is given by the probability that exactly one node transmits and the remaining $n-1$ nodes defer transmission, conditioned on the fact that at least one node transmits:

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_{\text{tr}}} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n}. \tag{10}$$

Considering that a random slot is empty with probability $(1 - P_{\text{tr}})$ and contains a successful transmission with probability $P_{\text{tr}}P_s$ and a collision with probability $P_{\text{tr}}(1-P_s)$, the saturation throughput S is given by

$$S = \frac{P_s P_{\text{tr}} L}{(1-P_{\text{tr}})\sigma + P_{\text{tr}}P_s T_s + P_{\text{tr}}(1-P_s)T_c}, \tag{11}$$

where L represents the length of data packet, T_s is the average time that the channel is sensed busy due to a successful transmission, T_c is the average time that the channel is sensed busy by each node during a collision, and σ is the duration of an empty slot time.

4.3. Average Packet Delay. A packet is dropped when it reaches the last backoff stage and experiences another collision. Let $E[T_{\text{drop}}]$ be the average number of slot times required for a packet to experience $m+1$ collisions in the $(0,\ldots,m)$ stages [14]:

$$E\left[T_{\text{drop}}\right] = \begin{cases} \dfrac{W\cdot(2^{m+1}-1)+(m+1)}{2}, & m \le m' \\ \dfrac{W\cdot(2^{m'+1}-1)+W\cdot 2^{m'}\cdot(m-m')+(m+1)}{2}, & m > m'. \end{cases} \tag{12}$$

The average packet delay for a successfully transmitted packet is defined as the duration from the time the packet is at the head of its MAC queue ready to be transmitted until an acknowledgement is received. So the average packet delay $E[D]$, provided that this packet is not discarded, is given by

$$E[D] = E[X]\cdot E[\text{slot}], \tag{13}$$

where the average length of a slot time is

$$E[\text{slot}] = (1-P_{\text{tr}})\sigma + P_{\text{tr}}P_s T_s + P_{\text{tr}}(1-P_s)T_c \tag{14}$$

Table 1: Parameters used in simulations.

Parameters	Values
MAC header	28 bytes
PHY header	24 bytes
RTS	44 bytes
CTS	39 bytes
HTS	51 bytes
ACK	38 bytes
Slot time	9 μs
SIFS	16 μs
DIFS	50 μs
aCWMin	15 slots
aCWMax	1023 slots
m	7
m'	5

and $E[X]$ is the average number of slot times required for successfully transmitting a packet given by:

$$E[X] = \begin{cases} \dfrac{W\cdot\left(1-(2p)^{m+1}\right)}{2(1-2p)} + \dfrac{1-p^{m+1}}{2(1-p)} - p^{m+1}\cdot E\left[T_{\text{drop}}\right], & m \le m' \\ \dfrac{W\cdot\left(1-(2p)^{m'+1}\right)}{2(1-2p)} + \dfrac{1-p^{m+1}}{2(1-p)} + \dfrac{W\cdot 2^{m'}\cdot p^{m'+1}\cdot(1-p^{m-m'})}{2(1-p)} - p^{m+1}\cdot E\left[T_{\text{drop}}\right], & m > m'. \end{cases} \tag{15}$$

4.4. Numerical Results. According to (11) and (13), we compare the saturation throughput and average packet delay achieved by our proposed adaptive data transmission scheme (adaptive) with the basic access scheme (basic access), RTS/CTS direct transmission scheme (rts/cts), "source-relay-destination" transmission scheme (s-r-d), and receiver maximal ratio combing scheme (mrc), through 1000 times Monte Carlo simulations. The main parameters are listed in Table 1 based on IEEE 802.11a standard.

Figure 5 reveals the relation between the saturation throughput and the number of nodes. It is shown that the throughput of all schemes deteriorate the number of nodes increases; however, the throughput of the adaptive data transmission scheme performs significantly better than that of the other schemes.

Figure 6 depicts the saturation throughput of five transmission schemes as the length of data frame at the MAC layer increases. When the data frame length is less than the RTS threshold, the saturation throughput of the adaptive data transmission scheme is equal to that of the basic access scheme. As the data frame length increases, the adaptive transmission scheme apparently outperforms the other four transmission schemes.

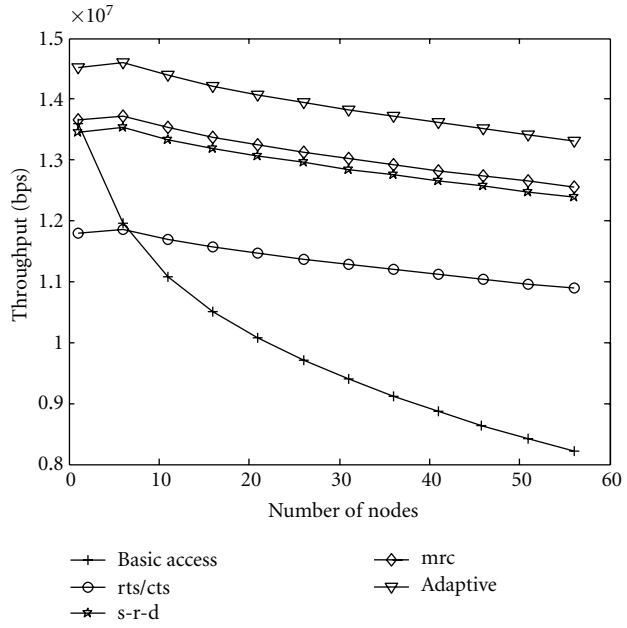


Figure 5: Throughput versus number of nodes ($L = 1500$ bytes).

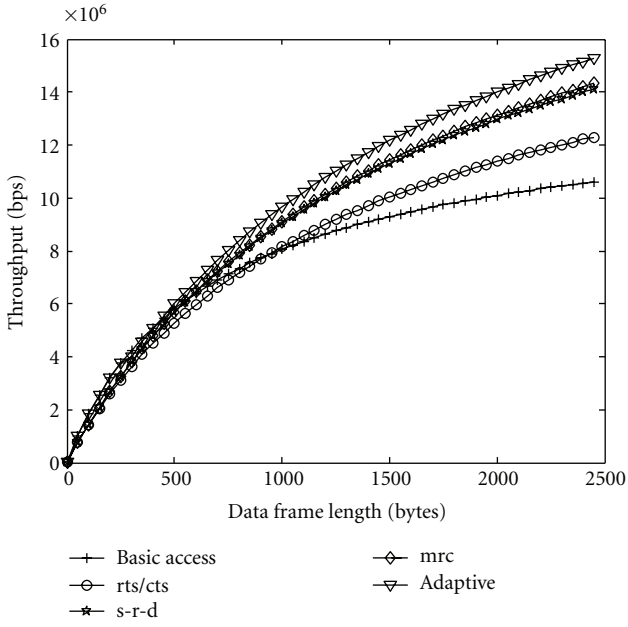


Figure 6: Throughput versus data frame length ($n = 20$ nodes).

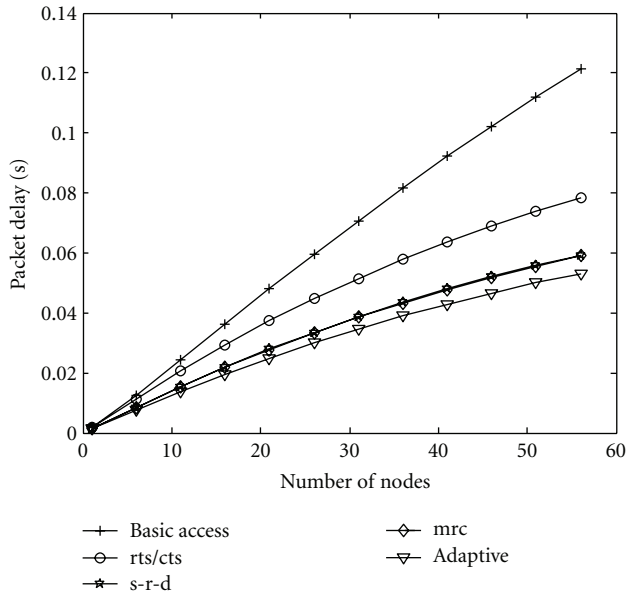


Figure 7: Packet delay versus number of nodes ($L = 1500$ bytes).

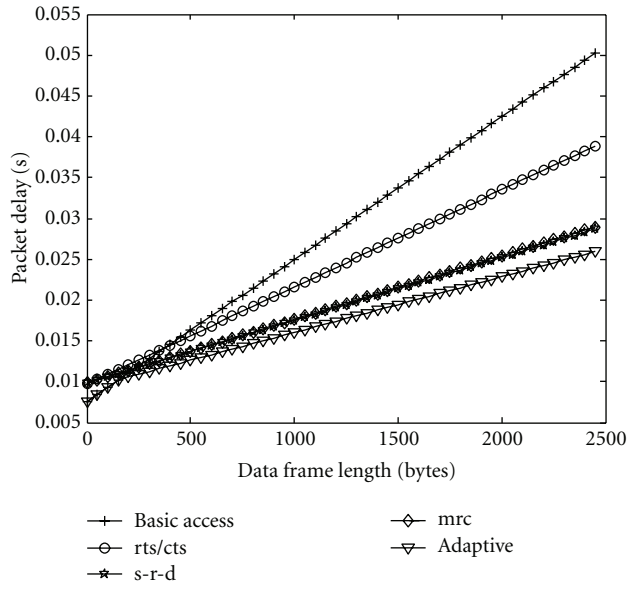


Figure 8: Packet delay versus data frame length ($n = 20$ nodes).

Figure 7 gives the packet delay varying with the number of nodes. It can be seen that the packet delay of the adaptive data transmission scheme is lower than that of the other schemes as the number of nodes increases.

Figure 8 describes the packet delay adopting different transmission schemes. Similarly, when the data frame length is less than the RTS threshold, the packet delay of the adaptive transmission scheme is equal to that of the basic access scheme. As the data frame length increases, the adaptive data transmission scheme outperforms the other schemes.

Analytical results shown in Figures 5–8 demonstrate the effectiveness of the adaptive data transmission algorithm. This is due to the fact that the adaptive data transmission scheme considers both the length of data frame at the MAC layer and instantaneous wireless channel conditions compared with the other transmission schemes when data packets are transmitted.

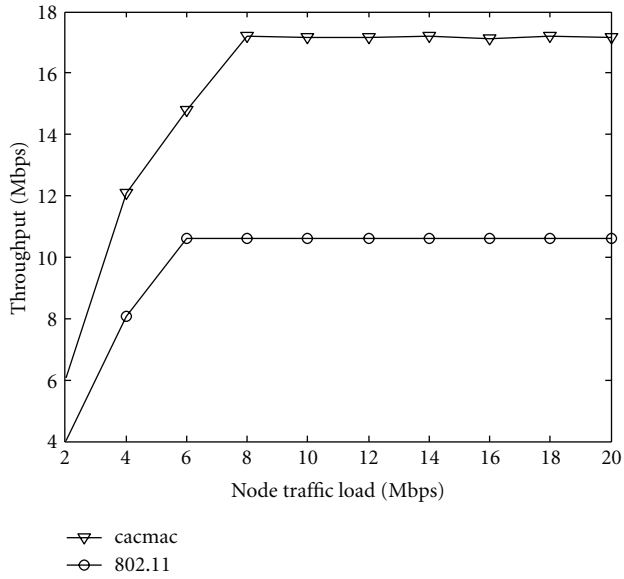


Figure 9: Network throughput versus node traffic load.

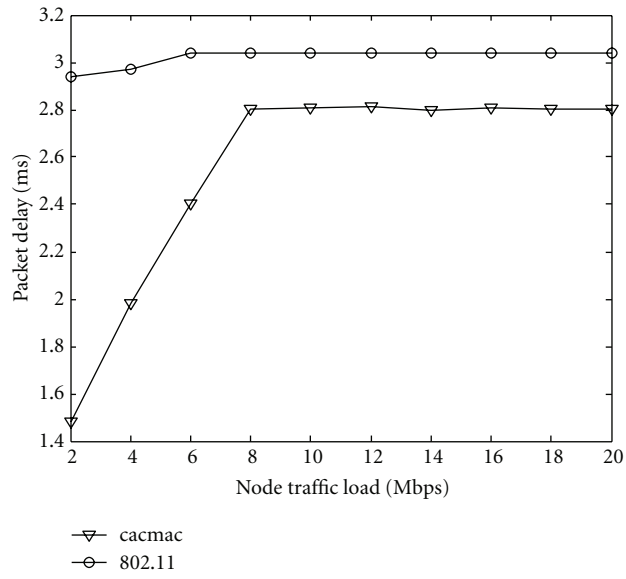


Figure 10: Packet delay versus node traffic load.

5. Evaluation of CAC-MAC Protocol

In this section, we evaluate the performance of the CAC-MAC protocol compared with the legacy IEEE 802.11 protocol based on NS2 [15, 16].

We consider a wireless ad hoc network based on IEEE 802.11a where 15 nodes are randomly deployed in the area of 300 m $\times$ 300 m. In this network, there exits three source-destination pairs that are selected randomly and the remaining nodes can be exploited as relay nodes. Each of the three source nodes generates traffic at the constant bit rate (CBR) of x bps with a packet length of 1500 bytes and the relay nodes work on the decode-and-forward (DF) mode. We change the value of "x" to reflect the scenarios of different node traffic loads. We adopt Rayleigh fading model in the simulations.

We compare CAC-MAC protocol (CAC-MAC) with legacy IEEE 802.11 protocol (802.11) in terms of network throughput and packet delay.

Figure 9 reveals the network throughput varying with source nodes traffic load, that is, the value of x. As each node traffic load increases, the network throughputs adopting CAC-MAC protocol and IEEE 802.11 protocol both increase up to saturation; however, the CAC-MAC protocol always significantly outperforms IEEE 802.11 protocol.

Figure 10 depicts the relation between packet delay and source nodes traffic load. It is evident that data packets in CAC-MAC protocol experience significantly less delay than in legacy IEEE 802.11 protocol as the source nodes traffic load increases.

As demonstrated in Figures 9 and 10, CAC-MAC protocol can achieve a much higher network performance than the legacy IEEE 802.11 protocol in terms of network throughput and packet delay. These improvements primarily stem from the novel cross-layer adaptive approach to design the cooperative MAC, which involves interaction of both MAC layer and physical layer.

6. Conclusions

In this paper, we propose a cross-layer adaptive cooperative MAC (CAC-MAC) protocol for IEEE 802.11 DCF-based wireless ad hoc networks, which consists of a realistic cooperative framework to exploit both MAC layer and PHY layer information. In CAC-MAC protocol, each data transmission is based on two planes: control plane and data plane. The control plane is to determine the data transmission mode, in which the main issues include relay selection and the cross-layer adaptive data transmission algorithm. The data plane is in charge of transmitting, receiving, or forwarding data packets according to the transmission mode. Simulation results based on NS2 show that our proposal can significantly improve network throughput and reduce packet delay compared with legacy IEEE 802.11 protocol.

Acknowledgments

The authors would like to thank the Editor Shukui Zhang and the anonymous reviewers for their constructive comments and valuable suggestions. This work was supported by the National Natural Science Foundation of China (no. 61002032) and the Doctoral Fund of Ministry of Education of China (no. 20094307110004).

References

[1] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—part I: system description," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1927–1938, 2003.

[2] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity—part II: implementation aspects and performance

analysis," *IEEE Transactions on Communications*, vol. 51, no. 11, pp. 1939–1948, 2003.

[3] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[4] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.

[5] T. E. Hunter and A. Nosratinia, "Diversity through coded cooperation," *IEEE Transactions on Wireless Communications*, vol. 5, no. 2, pp. 283–289, 2006.

[6] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman, "A simple cooperative diversity method based on network path selection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, pp. 659–672, 2006.

[7] H. Zhu and G. Cao, "rDCF: a relay-enabled medium access control protocol for wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1201–1214, 2006.

[8] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. S. Panwar, "CoopMAC: a cooperative MAC for wireless LANs," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 2, pp. 340–353, 2007.

[9] T. Korakis, Z. Tao, Y. Slutskiy, and S. Panwart, "A cooperative MAC protocol for ad hoc wireless networks," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 532–536, March 2007.

[10] F. Liu, T. Korakis, Z. Tao, and S. Panwar, "A MAC-PHY cross-layer protocol for wireless ad-hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '08)*, pp. 1792–1797, Las Vegas, Nev, USA, April 2008.

[11] J. W. Mark and W. Zhuang, *Wireless Communications and Networking*, Prentice-Hall, Upper Saddle River, NJ, USA, 2003.

[12] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535–547, 2000.

[13] H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement," in *Proceedings of the IEEE INFOCOM*, pp. 599–607, New York, NY, USA, June 2002.

[14] P. Chatzimisios, A. C. Boucouvalas, and V. Vitsas, "IEEE 802.11 packet delay—a finite retry limit analysis," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '03)*, pp. 950–954, December 2003.

[15] Network Simulator ns-2, http://www.isi.edu/nsnam/ns/.

[16] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 modeling and simulation in NS-2," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '07)*, pp. 159–168, October 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 161347, 9 pages
doi:10.1155/2012/161347

Research Article

A Mobile Agent Routing Algorithm in Dual-Channel Wireless Sensor Network

Kui Liu, Sanyang Liu, and Hailin Feng

Department of Applied Mathematics, Xidian University, Xi'an 710071, China

Correspondence should be addressed to Kui Liu, liukui003@gmail.com

Received 1 December 2011; Accepted 27 February 2012

Academic Editor: Shukui Zhang

A mobile agent routing algorithm (MARA) is presented in this paper, and then based on the dual-channel communication model, the two-layer network combination optimization strategy is also proposed. Since this strategy deals with the collision between packets and the multicast suppression in channel competitive process well, the blocking probability of network and the error rate of packet transmission can be cut down by utilizing this strategy. Furthermore, a restore rule for the failure optimal route is presented too. By using this rule, the optimal route can restore quickly in the local of fail nodes and most of the information of original optimal route can be reserved. Compared with other existing algorithms like ant colony optimization-based dynamic energy-efficient mobile agent routing algorithm (ADEEMA) and mobile agent-based wireless sensor network (MAWSN), simulation results show that MARA performs better in improving the success rate of packet transmission and cutting down the delay of communication. The success rate of packet transmission is improved 15%. Simultaneously, this algorithm can keep the ant agents away from the nodes with less residual energy in searching for the optimal route. This way can make the energy of each nodes on the optimal route overall decline and hence improve the lifetime of network.

1. Introduction

Sensor devices have already become more smaller in size, lower power, and lower cost. With the developments of microelectromechanical systems, wireless communications, and digital electronics, such extremely small devices integrate sensing data processing and communication capabilities. Large numbers of sensor nodes can be deployed in the areas of interest such as disaster places and use self-organization and collaborative methods to form a sensor network. The key factors that influence the performance of wireless sensor network are as follows. (1) *The packet collision between control packet and data packet.* In the process of channel competitive, how to significantly reduce the amount of packets conflict in shared channel is a research hot spot, when two sensor nodes attempt to use the shared channel transiting packets at the same time. The packet conflict between control packet and data packet is the prime reason which causes communication conflict in channel competitive, so the rate of them is the key factors that influence the performance of wireless sensor network. (2) *The redundancy information.* Since the sensor nodes in a wireless sensor network are deployed in an arbitrary manner, an important task of wireless sensor network is to monitor and collect the relevant data in a set of targets. So a full sensing coverage like in literatures [1, 2] is one of the main requirements, which demands that every location in the target should be covered by at least one sensor. Moreover, to cope with the problem of faulty sensor nodes and guarantee network functionality, duplicate coverage of the same region is an appealing solution through using appropriate sensors redundancy strategies. Sensors redundancy can guarantee that there exists at least one communication path between any pair of sensor nodes and meet the requirement of fault tolerant. But these redundancy sensor nodes will produce a large amount of redundancy data that needs to be processed. These redundancy dates stream need more transportation bandwidth to communication, this can lead to network communication blocking. Cormio and Li et al. proposed an approach to avoid collisions between data packets and control packets by using separate

channels for different kinds of packets in literatures [3–6]. This strategy can significantly reduce the amount of communication collisions and hence the energy consumed. But the local transmissions mode presented by these paper cannot balance the energy consumption of each nodes in the proposed route. This local transmission mode can lead to excessive use of some sensor nodes and lead to the premature death of these nodes. Fok, Li, and Chen et al. presented a mobile agent model in literatures [7–13]. Base on this model, they can successfully complete information fusion in sensor node and avoid transmitting large number of data, hence, reduce energy consumption of network. In wireless sensors network, processor node is the leader of data query processing and is the recipient of trade data stream, which can complete the effective collection of the correct data by means of moving the mobile agents from processor node to all the target nodes, and other nodes do not need to direct transmits data to process node. This strategy of collecting data has two advantages: (1) It can reduce the requirement of bandwidth and lowered the network load effectively. (2) According to network load, determine dynamically the way of data processing methods and the way of mobile agent migration strategies. So this strategy can be used to balance network load and improve network performance. Based on the ant colony optimization in literature [14], a dynamic energy-efficient mobile agent routing algorithm is presented in literature [12], the route chosen by this strategy can consider both the energy consumption and the node's residual energy. However, since this algorithm adopted the single channel wireless sensors network, the collisions between control packet and data packets cannot be avoided, and the error rate of packet transmission may be high. To solve the problems in these algorithms, a new mobile agent routing algorithm (MARA) based on the dual channel wireless sensors network is presented in this paper. This algorithm proposes a combination optimization route strategy, when there is enough idle resource in the data plane, the service blocked in the control plane can use these idle resources to transmit control packet in synchronous manner, so the blocking probability of network can be cut down by this means and the effective utilization of these idle resource in dual channel will be realized. Simultaneously, a restore rule for the failure optimal route is presented to adapt to the topology changes in dynamic sensor network as in literatures [15, 16]. When the network topology is changed, a new optimal route can be found fast by using this strategy and most information of the original optimal route can be reserved. So both the delay of communication and the energy consumption for restoring the optimal route can be cut down by this means.

The remainder of this paper is organized as follows: in Section 2, the related work is summarized; in Section 3, we describe the models in our work; in Section 4, we present the mobile agent combination optimization routing algorithm in detail; in Section 5, we present the strategy of rapid restoring optimal route on the basis of keeping most information of the original optimal route; in Section 6, we provide experimental results; finally, we conclude this paper in Section 7.

2. Related Work

The factors that influence the performance of wireless sensor network has been widely discussed in many papers. According to the cause of these factors, the factors that influence the performance of network are classified into three categories: the packet collision between control packet and data packet, the redundancy information, and topology control of network.

In a single channel ad hoc network as in literatures [1, 2], one channel is shared by a number of communication nodes, located in close proximity. The throughput of such a network depends largely upon the performance of the multiple access control (MAC) protocol in use, which controls and coordinates the access of the nodes to the shared channel. In order to increase the throughput, many MAC schemes require nodes to sense the common channel before packet transmission. However, collisions which arise when more than one packet is received at a node at the same time are still possible. The authors of [5] propose a collision avoidance scheme, they propose the dual busy tone multiple access protocol, they use the RTS packets to initiate channel request, two out-of-band busy tones are then used to protect the RTS packets and the data packets, respectively. In this strategy, the whole channel is split into two subchannels for message transmission and control packet transmission. A ready node sends its request to the target node on the request channel, successful requests will be acknowledged by the node passed through before the data packet is transmitted. This way can successfully put down the collision of network.

In literatures [4, 6], it was proved that the packet conflict between control packet and data packet is the prime reason which causes communication conflict in the process of channel competing. These communication conflict and interference seriously decrease the packet delivery ration of multihop flows. So the rate of them is the key factors that lead to the poor throughput performance of wireless sensor network. The authors of literature [6] present a novel effective random medium access control protocol. This new protocol uses an out-of-band busy tone and two communication channels, one for control frames and the other for data frames, and can give a comprehensive solution to the aforementioned problems in single channel ad hoc network. This protocol can improve the throughput by up to 20% for one-hop flows and by up to 5 times for multihop flows under heavy traffic.

Sensors redundancy can guarantee connection of network and meet the requirement of fault tolerant. But these redundancy sensor nodes will produce a large amount of redundancy information that needs to be processed. So, how to complete data aggregation in order to put down the amount of redundancy information has been well studied in recent years. In network aggregation means, computing and transmitting partially aggregated data rather than transmitting all the raw data to reduce energy consumption. There are a vast amount of extant works on data aggregation in the literatures [7–13].

Mobile agent model was proposed for completing data aggregation and putting down the amount of redundancy

information in literature [12], the mobile agent is a special kind of software that propagates over the network either periodically or on demand (when required by the applications). It performs data processing autonomously while migrating from node to node. As described in [12], many inherent advantages of the mobile agent architecture make it more suitable for wireless sensor network, and it is found to be particularly useful for data fusion tasks in distributed wireless sensor network. To solve the problem of the overwhelming data traffic, Fok et al. [7] proposed mobile agent-based distributed sensor network for scalable and energy efficient data aggregation too. By transmitting mobile agent to sensor nodes, a large amount of sensory data can be reduced or transformed into a small amount of data by eliminating the redundancy.

How to adapt to the topology changes is a research hot point in the learning of the dynamic wireless sensor network. Precious sensor node battery power can be saved by topology control measures that can dynamically adjust the radio transmission range of individual sensor nodes to balance energy efficiency while maintaining adequate network connectivity [15, 16]. In order to adapt to the topology changes in dynamic sensor network, a restore rule for the failure optimal route is presented in literature [15]. A new optimal route can be found fast by this strategy and most information of the original optimal route can be reserved. Base on this restore rule, we propose a new restore strategy, this strategy is that, when there are some nodes failed in the optimal route, we first draw a circle such that its diameter is denoted by the line between the failure node's father node and child node, then we can find some new nodes around the failure node to repair this failed optimal route in this circle, thus we can get a new optimal route near the old one. So both the delay of communication and the energy consumption for restoring the optimal route can be cut down by this means.

3. The Model of MARA Algorithm

3.1. Mobile Agent's Attributes and the Communication Energy Consumption Model. In this paper, mobile agent has the properties as follows. (1) Mobile agent has to store the position information of the processing node, the target node and the nodes visited by this mobile agent. (2) Mobile agent has the function to complete data fusion on each target node. (3) Mobile agent has the function to complete data acquisition and processing on each target node. The capacity of data taken by mobile agent is constantly and the data precision will be increased when the mobile agent travels along the optimal route which can reach all the target nodes in network. At the same time, we use the radio model as discussed in [15] to analyze energy consumption of sensor nodes. For transferring k-bits data message between the father node and the child node, the transmitting and receiving energy costs are given by (1) and (2), respectively, d in (1) is the distance between the two nodes:

Transmitting energy costs model:

$$E_T(k,d) = \begin{cases} E_1 \times k + a \times k \times d^2 & d \le d_0, \\ E_1 \times k + b \times k \times d^4 & d > d_0. \end{cases} \quad (1)$$

Receiving energy costs model:

$$E_R(k) = E_1 k, \quad (2)$$

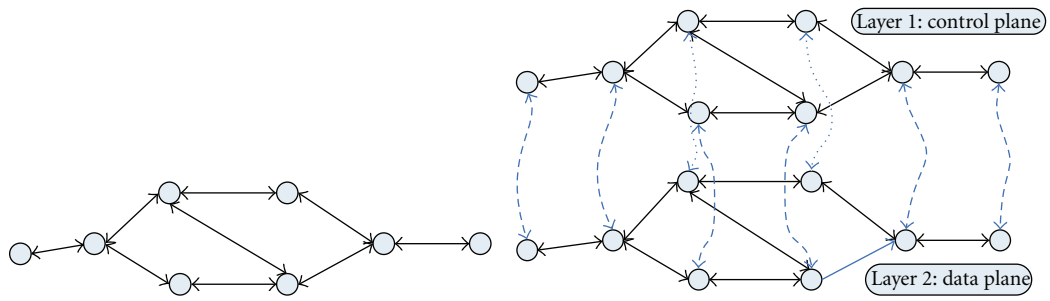
where $E_T(k,d)$ in (1) denotes the total dissipated energy in the transmitter and $E_R(k)$ in (2) represents the energy cost incurred in the receiver. The parameters E_1 in (1) and (2) is the per bit energy dissipation for transmission and reception. In order to maintain an acceptable signal-to-noise ratio and transfer data messages reliably, we use both the free-space propagation model and the two-ray ground propagation model to approximate the path loss sustained due to wireless channel transmission. When d is less than or equal to the threshold transmission distance d_0, the free-space model is employed, and let transmit amplifier parameters corresponding to this model is equal to a. When d is greater than the threshold transmission distance d_0, the two-ray model is applied for this case and let transmit amplifier parameters corresponding to this model is equal to b.

3.2. The Two-Layered Graph Network Model. In this section, to solve the channel converter problem in wireless sensor network, the two-layered graph model is constructed based on graph theory. In this model, the number of available channels is always one. Each channel can be represented by a layer which has the same set of sensor nodes and links that are duplicated in each layer (Figure 1).

In this model, two layers, each of which corresponds to a single channel, are generated. The channel converter can be represented by the inter layer links between the same node in different layer. By utilizing this model, because each node and link deals with only one channel, so the channel converter problem is simplified into a routing problem over the two-layered graph model. The two layers are one for transmitting control packet and the other for transmitting data packet. By using the separate channel layers, we can avoid collisions between control packets and data packets. The channel layer used to transmit control packet is referred to as the control plane, and the other one is referred to as the data plane, when there is enough idle resource in data plane, a combination optimal strategy in control plane and data plane is proposed. The service blocked in control plane can use this idle resource to transmit control packet in synchronous manner. So the blocking probability of network can be cut down and the success ratio of packet transmissions can be improved by using this strategy.

3.3. Neighbor Information Table and the Optimal Route Evaluation Standard. In this paper, a wireless sensor network model similar to those used in literatures [9–12] is applied, with the following properties.

(1) All sensor nodes are immobile and have a unique ID.



(a) Physical topology (b) Layered-graph with two channels

Figure 1: Two-layered graph model.

(2) All sensor nodes are energy constrained with a uniform initial energy accusation.

(3) All sensor nodes have two communication channels, one for transmitting control packets and the other for transmitting data packets.

(4) All sensor nodes have two omni-directional antennas, one for transmitting packets and the other for receiving packets.

Each sensor nodes should build its own neighbor information table. Every neighbor information table has its own memorizer to store the information as follows.

(1) The ID of the local sensor node and its neighbor sensor nodes.

(2) The residual energy of the local sensor node and its neighbor sensor nodes.

(3) The position information of the local sensor node and its neighbor sensor nodes.

(4) The density of ant pheromones between the local sensor node and its neighbor sensor nodes.

This paper also proposes a new optimal route evaluation standard which can be used to evaluate the performance of the optimal route. This evaluation standard is considerate to both the overhead on the route and the residual energy of the nodes. This evaluation standard can keep the optimal route away from the nodes with less residual energy and make the energy of each nodes on the optimal route overall decline and hence improve the lifetime of wireless sensor network. This evaluation standard is defined as

$$D = \sqrt{\lambda_1 \sigma^2 + \left(\frac{\lambda_2 \times En_1}{\lambda_3 \times En_2 + \lambda_4 \times En_3}\right)^2}. \quad (3)$$

The less value of D, the better route is. In expression (3), $\lambda_1, \lambda_2, \lambda_3$, and λ_4 are partialness bias parameters, and they satisfy $\lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0, \lambda_4 > 0$, and $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$. En_1 denotes the total energy consumption in the route, En_2 is the average residual energy of all the nodes in the route, En_3 is the minimum residual energy of all the nodes in the route. σ is the standard deviation of the residual energy which is that of all the nodes in the route.

4. The Algorithm of MARA

In dual-channel wireless sensor network, to put down the probability of collision between data packet and control packet, an isolation transmission strategy is proposed in this paper. This strategy can avoid collision between the different packets by transmitting control packet in control plane and transmitting data packet in data plane. The optimal route between the source node and the target node can be found through using the Mode 1 of the mobile agent routing algorithm to search in control plane. In data plane, the data packets can be sent across the optimal route which was found by using Mode 1 of the mobile agent routing algorithm. When there is enough idle resource in data plane, the blocking transport traffic in control plane can be sent down to search for the optimal route in data plane. This way can put down the blocking probability of network and the Mode 2 of combination optimization strategy can be put into effect.

Mode 1. The process of searching for the optimal route in control plane by using the mobile agent routing algorithm is given as below.

Phase 1. There are m species ants in the control plane and every species has n ants. If the initialization source node v gets the request from the set of target nodes $u = \{u_i | i = 1, 2, \ldots, m\}$, the ants are dispatched at regular intervals from v-node to search the target nodes. First v-node uses the position information of each target nodes to find the nearest target node. Assume that node u_i is the nearest target node to v-node. The ant is dispatched from v-node to search for the node u_i has to storage information as follows: (1) the position information of node u_i; (2) the position information of node v; (3) the ID number and the residual energy of the nodes have been visited by the ants.

Phase 2. When the ant k arrived at the i-node, the information of the i-node's neighbor nodes in the neighbor information table will be modified as the following rules: (1) if the target node u_i in the neighbor information table, only the target node u_i can be reserved in this neighbor information table; (2) if the node belongs to the neighbor information table of the i-node has been visited by the

ant k, this node should be eliminated from the neighbor information table; (3) after rule one, suppose the target node u_i is not in this neighbor information table, let d_i be the distance between the i-node and the target node u_i. If the distance between the target node u_i and the node belongs to this neighbor information table is larger than d_i, then this node should be eliminated from the neighbor information table. Base on the fake random proportion rule, the ant k in i-node will choose j-node belongs to the modified neighbor information table of i-node as the next hop node. When the ant k arrived at the j-node, the information of the ant k storage should be modified and the local pheromone should be modified as follows formula:

$$\hbar_{ij} \longleftarrow (1-\xi)\hbar_{ij} + \xi\hbar_0. \tag{4}$$

In this expression, ξ and $\hbar_0$ are two parameters, and ξ satisfies $0 < \xi < 1$, $\hbar_0$ is the initialization pheromone.

Phase 3. When the ant k arrived at the target node u_i, first, the ant k should judge the position information of the target node u_i whether is the same as initialization source v-node's. If the position information of the target node u_i is the same as initialization source v-node's, then turn to step 5; otherwise, the ant k will delete the target node u_i from the set of target nodes. If the set of target nodes have been modified, and it is not an empty set, then, the ant k will find the nearest target node to the target node u_i by using the position information of every target node. Suppose the target node u_j is the nearest target node to the target node u_i, we assign the position information of the node u_i to the v-node and assign the position information of the node u_j to the node u_i, and turned to Phase 2; otherwise turned to Phase 4.

Phase 4. Put the initialization source v-node into the set of the target nodes, and let the initialization source v-node as the target node u_i; let the last visited target node u_i as the source v-node and turned to Phase 2.

Phase 5. The route optimal degree can be calculated according to formula (3). When the whole of a species ants arrive in the target nodes v, the route which has the smallest route optimal degree should be put into the set of the optimization routes, then the global pheromone will be modified as follows formula:

$$\hbar_{ij} \longleftarrow (1-\rho)\hbar_{ij} + \rho\Delta\hbar_{ij}, \quad \forall (i,j) \in T_{vu}, \tag{5}$$

where $\Delta\hbar_{ij} = 1/C_{vu}$ is the increment of the pheromone; the parameter C_{vu} is the overhead of the optimization route; the parameter ρ is the evaporation rate of pheromone; the parameter T_{vu} denotes the optimization route. Then a new ant belongs to the another species will be dispatched to find the optimization route again. When the ants have been dispatched, the route with the smallest route optimal degree in the set of the optimal routes can choused, as the route which the mobile agent travel along to complete data collection and fusion.

Mode 2. The process of searching for the optimal route in two-layers plane by using the combination optimization strategy is described as follows.

Phase 1. Ant agents search the optimal route for each traffic synchronously in control plane. When there is enough network resources in the control plane, the ant agents will search the optimal route for this traffic in control plane like Mode 1, otherwise, proceed to next step.

Phase 2. First, the ant agents in the i-node conduct a survey to make sure whether there is enough bandwidth resources in data plane between the i-node and the nodes lay in the i-node's neighbor information table which the ant agent did not visit before. If there are not enough bandwidth resources, the ant agents ceased implementing this searching route business, otherwise, proceed to next step.

Phase 3. The ant agents find the next hop node j to transmit inquiry information between i-node and j-node in data plane like Mode 1. When the inquiry information arrived at j-node, the ant agents should conduct a survey to make sure whether there is enough bandwidth resources in the control plane between the j-node and the nodes lay in the j-node's neighbor information table which the ant agents did not visit before. If there are not enough bandwidth resources, the ant agents continue to search the next hop node in data plane. Otherwise, the ant agents return to the control plane and search the optimal route in control plane like Mode 1.

5. The Strategy of Rapid Restoring Failed Optimal Route

Let v-node be the source node and let u-node be the target node. In the free space model, the energy consuming for transmitting a unit of information between v-node and u-node is denoted by E_n. Let $v \rightarrow w \rightarrow u$ be a route from v-node to u-node which passes through w-node, if the energy consumed in this route for transmitting a unit of information is less than E_n, then $d_{vw}^2 + d_{wu}^2 < (d_{vu})^2$, by cosine theorem, the angle $\angle vwu$ is an obtuse angel, hence, the w-node in the circle which diameter is vu, denoted by the infection circle between v-node and u-node. If the energy consumed for transmitting a unit of information is equal to E_n, then $d_{vw}^2 + d_{wu}^2 = (d_{vu})^2$, by Pythagorean theorem, the angle $\angle vwu$ is a right angel, hence, the node w is on the infection circle. The node w is said to be the infection shortcut node if the energy consumed for transmitting a unit of information is less than or equal to E_n in route $v \rightarrow w \rightarrow u$. The diameter vu divide the infection circle into two semicircle, so the routes from v-node to u-node which pass through the infection shortcut nodes in any semicircle have the following properties.

Theorem 1. *Let $vw_1w_2 \ldots w_hu$ be a route travel along the infection shortcut nodes in the semi-circle from the source v-node to the target u-node, then, for transmitting a unit of information, the energy consumed by this route is less than or equal to E_n.*

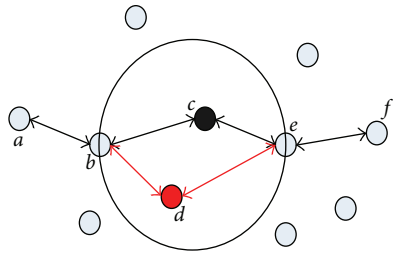


Figure 2: The local route restoration.

Proof. Since the nodes $w_1, w_2, \ldots, w_h$ lie in the semi-circle, then $\triangle vw_1w_2$, $\triangle vw_2w_3, \ldots, \triangle vw_{(h-1)}w_h$ are obtuse triangles, $\triangle vw_hu$ is a right triangle. Since $\triangle vw_1w_2$ is an obtuse triangle, by cosine theorem, then $d^2_{vw_1} + d^2_{w_1w_2} < d^2_{vw_2}$. Since $\triangle vw_2w_3$ is an obtuse triangle too, similarly, by cosine theorem, we can get formula: $d^2_{v,w_2} + d^2_{w_2,w_3} < d^2_{v,w_3}$, that is, $d^2_{v,w_1} + d^2_{w_1,w_2} + d^2_{w_2,w_3} < d^2_{v,w_3}$. Furthermore, $d^2_{v,w_1} + d^2_{w_1,w_2} + \cdots + d^2_{w_{(h-1)},w_h} < d^2_{v,w_h}$. Since $\triangle vw_hu$ is a right triangle, by Pythagorean theorem, we can get formula: $d^2_{v,w_h} + d^2_{w_h,u} = d^2_{vu}$. Furthermore, $d^2_{v,w_1} + d^2_{w_1,w_2} + \cdots + d^2_{w_{(h-1)},w_h} + d^2_{w_h,u} < d^2_{v,u}$, and hence, the energy consumption for transmitting a unit of information is less than or equal to E_n.

Now turn to the case of two-ray model. We compare this with the case of the free-space model and claim that Theorem 1 still holds good under these circumstances. In the free-space model, the distance between nodes is large than d_0, suppose w-node is a node in the circle whose diameter is vu, so the angle $\angle vwu$ is an obtuse angel, by cosine theorem, we get the formula as follows: $d^2_{v,w} + d^2_{w,u} < d^2_{v,u}$, so we can get formula: $d^4_{v,w} + d^4_{w,u} < (d^2_{v,w} + d^2_{w,u})^2 < (d^2_{v,u})^2$. Furthermore, $d^4_{v,w} + d^4_{w,u} < d^4_{v,u}$.

Hence, under the two-ray model, w-node is also a infection shortcut node in the infection circle whose diameter is vu. Suppose $vw_1w_2 \ldots w_hu$ be a route travel along the infection shortcut nodes in the semicircle from the source v-node to the target u-node, by cosine theorem and the Pythagorean theorem, then $d^2_{v,w_1} + d^2_{w_1,w_2} + \cdots + d^2_{w_{(h-1)},w_h} + d^2_{w_h,u} < d^2_{v,u}$. Further more,

$$\begin{aligned} &\left(d^2_{v,w_1} + d^2_{w_1,w_2} + \cdots + d^2_{w_{(h-1)},w_h} + d^2_{w_h,u}\right)^2 < \left(d^2_{v,u}\right)^2, \\ &d^4_{v,w_1} + d^4_{w_1,w_2} + \cdots + d^4_{w_h,u} \\ &< \left(d^2_{v,w_1} + d^2_{w_1,w_2} + \cdots + d^2_{w_h,u}\right)^2 < d^4_{v,u}. \end{aligned} \tag{6}$$

So Theorem 1 still holds good under the two-ray model. □

Theorem 1 provides a rapid restoration strategy for the failed optimal route, The detail is that, when there are some nodes failed in the optimal route. we first draw a circle such that its diameter is denoted by the line between the failure node's father node and child node, then we apply the MARA algorithm to restore this section failed optimal route in this circle. Finally, we can find some new infection shortcut nodes around the failure node to repair this failed optimal route on the basis of keeping most information of the original optimal route. Figure 2 shows a section of the optimal route from the source node to the target node, the c-node in this optimal route cannot work due to some reasons. We draw a circle which diameter is denoted by the line between the failure node's father b-node and child e-node. Then, we can find some infection shortcut nodes in this circle to restore this section of the optimal route. Since d-node in this circle, b-node, and e-node in the neighbor list of the d-node's, we can replace the failure route $b \rightarrow c \rightarrow e$ by $b \rightarrow d \rightarrow e$. This strategy can help us to complete the local route restoration and keep most information of the original optimal route.

6. Experimental Results

In this section, we perform the experiments with the network simulator platform *OMNeT++* version 4.1. The network used in experiments is randomly generated. The programming language is C#. We implement these experiments to evaluate the performance of MARA and then compare the performance of MARA with other algorithms like ADEEMA in literature [12] and MAWSN in literature [9]. In our experiments, we use a network with 250-node random deployed in a square area of 200 m × 200 m. The source node located at ($x = 0$, $y = 0$) and the target node located at ($x = 185$, $y = 185$). The initial energy of sensor node is 1 J, the emitting power of sensor node is 15 mW, the received power and the idle power of sensor node are 8 mW. The bandwidth of the total channel is assumed to be 2 Mbps, the bandwidth of the control channel is assumed to be 0.4 Mbps, The bandwidth of the data channel is assumed to be 1.6 Mbps. Let the parameters in formula (3) equal to $\lambda_1 = \lambda_2 = 0.3$, $\lambda_3 = \lambda_4 = 0.2$. λ_1, λ_2, λ_3, and λ_4 denote the adjustable weight of different variable. We assume that the data packet is 400 bytes, the request frame, the remove frame, and the acknowledgement frame are 40 bytes.

Figure 3 shows the relationships of different algorithms' success rate of packet transmission. According to Figure 3, we can conclude that, when the load of network is heavy, the production rate of data packet have significant influence upon the success rate of packet transmission. The reason is that the collision between the data packet and the control packet will aggravate constantly with the network load going up. However, since the MARA algorithm adopts the strategy of isolation transmission, which can avoid the collision between the data packet and the control packet during the course of network load going up and hence improve the success rate of the packet transmission.

Figure 4 shows the relationships of different algorithms' average communication delay. The delay of consumption is estimated by using the average value of the absolute differences between two frames. One is the time consumption for transmitting packets from source node to target node in a light load network, the other is the time consumption for transmitting packets use different algorithms separately under different load conditions. According to Figure 4, we can conclude that, when the load of network is heavy, the average communication delay of MARA is the smallest one. The reason is that the strategy of combination optimal is adopted by MARA, this way can make full use of the idle

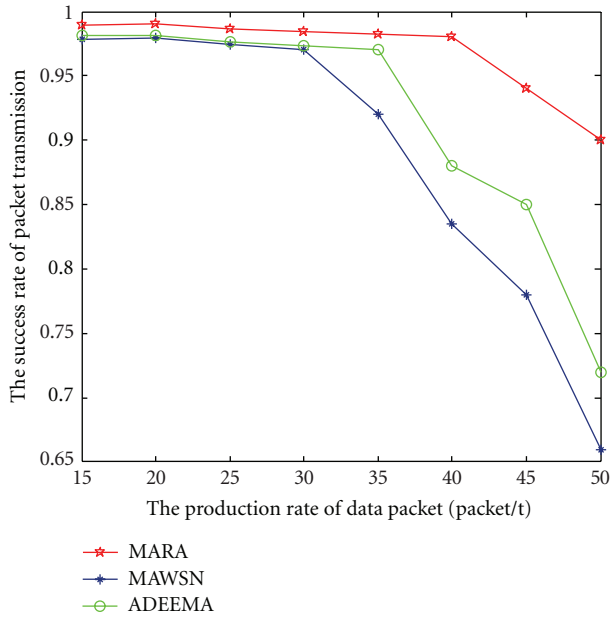


Figure 3: The success rate of packet transmission.

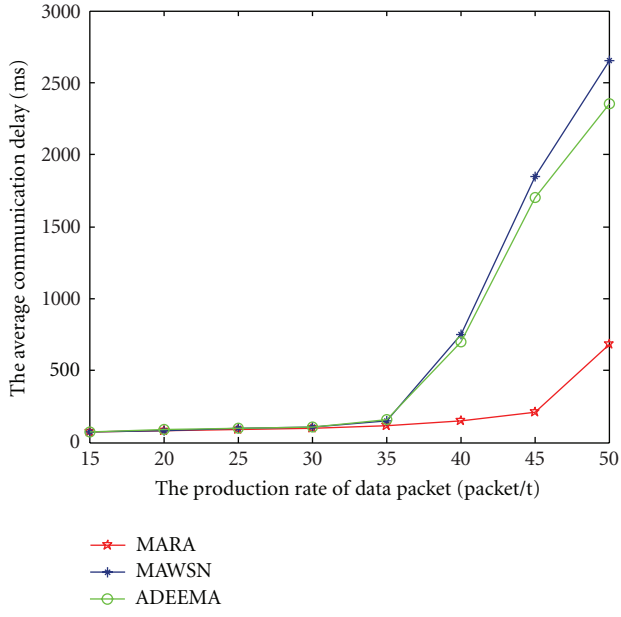


Figure 4: The average delay of communication.

channel resource in different layers. When there is enough idle resource in the data plane, the service blocked in the control plane can use these idle resource to transmit control packet in synchronous manner. So this way can put down the delay of communication efficiently.

To further verify the performance of three algorithms, the following experiment is done. Fixed the position of the source node at ($x = 0$, $y = 0$), then continual readjust the position of the target node. The MARA, MAWSN, and ADEEMA are separately used to search for the optimal

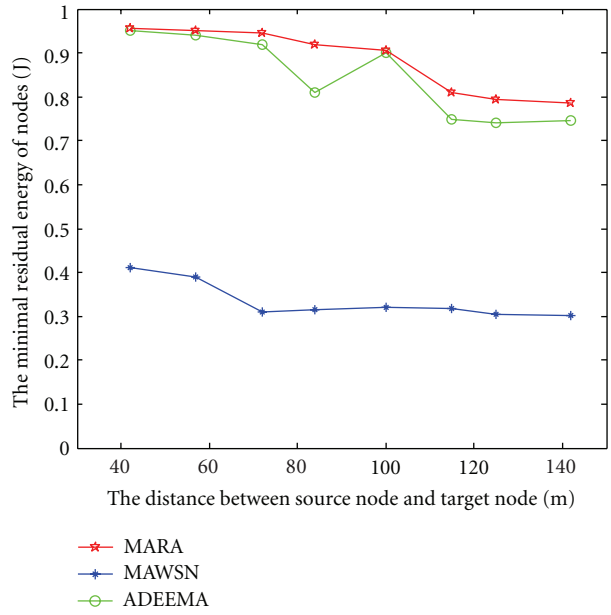


Figure 5: The minimal residual energy of the nodes.

routes from the source node to the target node in network. When the target node located at different position, we transmit a unit of information equal to 400 bytes along the optimal routes which were found by different algorithms. Figure 5 shows the relationships of the minimal residual energy of the nodes in different optimal routes. According to Figure 5, it can be seen that the minimal residual energy of the nodes in optimal route which was found by MARA is higher than the minimal residual energy of the node in optimal routes which were found by MAWSN and ADEEMA. This is because we proposes a new optimal route evaluation standard and use this rule to evaluate the performance of the optimal route. This evaluation standard can keep the ant away from the nodes with less residual energy in the process of searching route, so the minimal residual energy of the nodes in optimal route is the largest. The MAWSN algorithm based on the greed strategy is proposed, this greed strategy does not consider the residual energy of nodes in choosing the next hot node. So the minimal residual energy of the nodes in optimal route which was found by MAWSN is the smallest one.

Figure 6 shows the relationships of total energy cost of the different optimal routes. According to Figure 6, we can conclude that the total energy consumption of the optimal route which was found by MARA is smaller than the total energy consumption of the optimal routes which were found by ADEEMA and MAWSN, this stems from two respect reasons commonly. Firstly, this is because MARA algorithm adopts the strategy of combination optimal, this way can improve the success rate of packet transmission, hence put down the total energy consumption of the optimal route. Secondly, MARA algorithm uses the inquire information to search for the optimal route and lets the mobile agent travel along the optimal route to complete data acquisition and

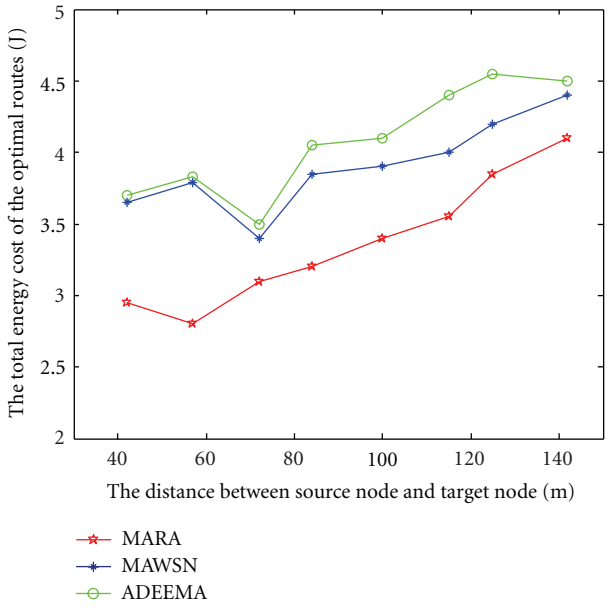


Figure 6: The total energy cost of the optimal routes.

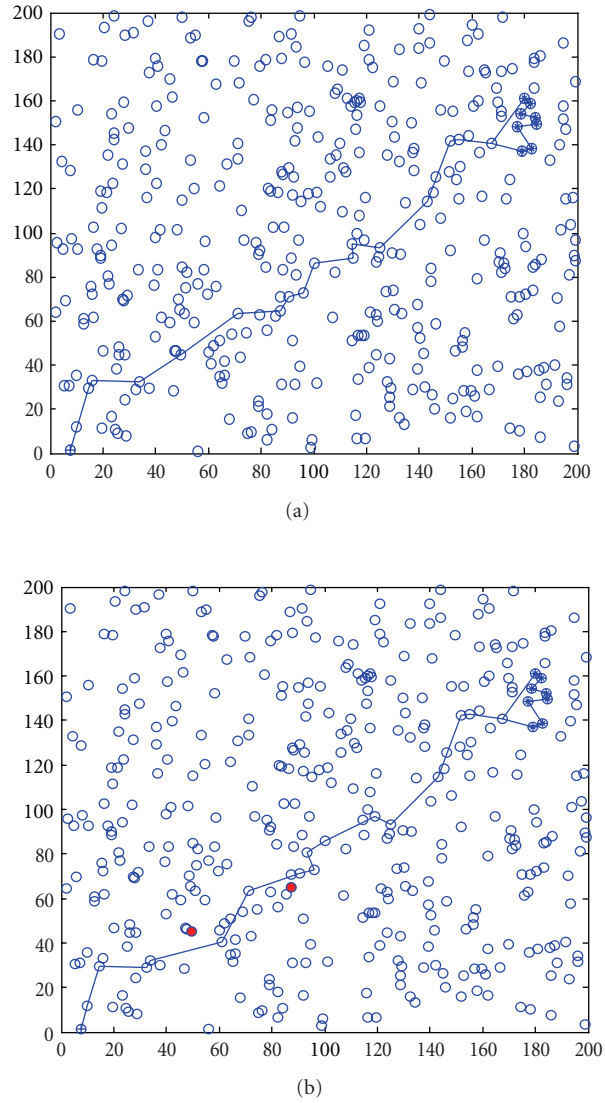


Figure 7: (a) The original optimization route. (b) The restoring optimization route.

processing. This strategy can put down the workload of data transmission, hence reduce the total energy cost. However, MAWSN and ADEEMA algorithm use the mobile agent to search for the optimal route, mobile agent contain more data quantity than inquire information, so these algorithms improve the amount of data transmission and hence improve the total energy cost.

How to adapt to the topology changes is a research hot point in learning of the dynamic wireless sensor network, the simplest way is to restart the algorithm to search for the other optimal route in network. But this strategy does not make full use of the information of original optimal route, while the new optimal route will be in some sense related to the original one. A new rapid restoring strategy is presented in Section 4, this strategy can reserve enough original information to speed up the process of searching for the new optimal route. This rule can search some shortcut nodes around the failure node to repair this failed route, hence the new optimal route can be obtained quickly, and most information of original optimal route can be reserved. We can see from the following experiment that our strategy can find a new optimal route quickly when the topology changes. Fixed the position of the source node at $(x = 8, y = 0)$, the number of target node is eight, fixed the position of the target nodes at $(x = 180, y = 137)$, $(x = 183, y = 139)$, $(x = 178, y = 149)$, $(x = 181, y = 158)$, $(x = 187, y = 151)$, $(x = 185, y = 153)$, $(x = 182, y = 163)$, $(x = 185, y = 160)$. In Figure 7, we prove the rapid restoring strategy is available, the Figure 7(a) is the optimal route between the source node and the target nodes which was found by MARA, when two red nodes fail, a new optimal route as shown in Figure 7(b) can be obtain by this rule. According to Figure 7, we can find that the two optimal routes are almost identical except the area where the nodes fail. Because this rule only restores in the area where topology changes, we can reserve enough information of original route. Thus, we can get a new optimal route near the old one. Both the delay of communication and the energy consumption for restoring the optimal route can be cut down by this means.

MARA is a novel evolutionary algorithm based on ant colony strategy which derived from the foraging behavior of real ant. This strategy let sensor nodes periodically release inquiry information which has the property of ants. The node received inquiry information will updat its local information opportunely. So, MARA is a scalability algorithm which is of strong suitability. MARA has polynomial computational complexity, its computational complexity is $O(m * n^3)$. The computational complexity of MAWSN is $O(m * n^3 + m * n)$ and ADEEMA is $O(m * n^3 + m * \lg n)$, where m is the numbers of cycle, n is the number of

node. It can been seen MARA has the lowest computational complexity. The result of the above-mentioned experiment is in accordance with this analysis.

7. Conclusions

A mobile agent routing algorithm is presented in this paper to solve the problems of collision between packets and multicast suppression in channel competitive process. First, with a two-layer graph model, the channel converter problem in dual-channel wireless sensor network can be simplified into a routing problem over the two-layer graph, so we can search for routes in the control plane and transport traffic in the data plane synchronously. Then, the control plane and the data plane are integrated into a two-layer network, and searching route for each traffic in the two-layer network opportunely. This algorithm can make full use of these idle resource in different layers, when there is enough idle resource in the data plane, the service blocked in the control plane can use these idle resource to transmit control packet in synchronous manner, so the blocking probability of network can be cut down by this way. This strategy deals with the collision between packets and multicast suppression in channel competitive process well, so the error rate of packet transmission can be cut down by this strategy. Second, a data fusion strategy based on the mobile agent model is also presented. This strategy can reduce the amount of transmitted data, hence reduce the energy consumption of wireless sensor network. Last, a restore rule for the failure optimal route is developed to adopt to the topology changes in dynamic sensor network. The optimal route can restore quickly in the local area of failure sensor nodes, and most of the information of original optimal route can be reserved by this rule. So both the delay of communication and the energy consumption to restore the optimal route can be cut down by this way. Although the assumption of node has two communication channel improving the cost of sensor about 6%, this way can improve the success rate of packet transmission by 15% and cut down the delay of communication evident. Hence, these benefits are worth to improve the cost of sensor.

Acknowledgment

This project is supported by the National Natural Science Foundation of China (Grants nos. 60874085, 60974082), the Fundamental Research Funds for the Central Universities (Grant no. JY10000970013), the Foundation of State Key Laboratory of ISN.

References

[1] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Localized sensor area coverage with low communication overhead," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 661–672, 2008.

[2] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: a survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.

[3] C. Cormio and K. R. Chowdhury, "A survey on MAC protocols for cognitive radio networks," *Ad Hoc Networks*, vol. 7, no. 7, pp. 1315–1329, 2009.

[4] L. Li, W. Jiang, L. Sun, and X. Fan, "Receiver-based cross-layer forwarding protocol for mobile sensor networks," *Computer Research and Development*, vol. 46, no. 1, pp. 120–128, 2009.

[5] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 349–365, 2003.

[6] H. Zhai, J. Wang, and Y. Fang, "DUCHA: a new dual-channel MAC protocol for multihop ad hoc networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 11, pp. 3224–3233, 2006.

[7] C. L. Fok, G. C. Roman, and C. Lu, "Agilla: a mobile agent middleware for self-adaptive wireless sensor networks," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4, no. 3, article 16, pp. 4–29, 2009.

[8] Z. Y. Li and H. S. Shi, "A data aggregation algorithm based on grid and mobile agent in wireless sensor networks," *Chinese Journal of Sensor and Actuators*, vol. 21, pp. 624–628, 2008.

[9] M. Chen, T. Kwon, Y. Yong, and V. C. M. Leung, "Mobile agent based wireless sensor networks," *Journal of Computers*, vol. 1, pp. 14–21, 2006.

[10] E. F. Nakamura, A. A. F. Loureiro, and A. C. Frery, "Information fusion for wireless sensor networks: methods, models, and classifications," *ACM Computing Surveys*, vol. 39, no. 3, Article ID 1267073, 55 pages, 2007.

[11] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 2, article 7, 40 pages, 2008.

[12] W. Zheng, S. Y. Liu, and X. L. Kou, "Dynamic mobile agent routing algorithm in sensor network," *Control and Decision*, vol. 25, no. 7, pp. 1035–1039, 2010.

[13] Z. Li and H. Shi, "A data-aggregation algorithm based on minimum steiner tree in wireless sensor networks (WSN)," *Journal of Northwestern Polytechnical University*, vol. 27, no. 4, pp. 558–564, 2009.

[14] D. Marco and S. Thomas, *Ant Colony Optimization*, MIT Press, Cambridge, Mass, USA, 2006.

[15] W. Zheng, S. Liu, and X. Kou, "A route restoration algorithm for sensor network via ant colony optimization," *Journal of Xi'an Jiaotong University*, vol. 25, pp. 1035–1039, 2010.

[16] R. W. Ha, P. H. Ho, X. S. Shen, and J. Zhang, "Sleep scheduling for wireless sensor networks via network flow model," *Computer Communications*, vol. 29, no. 13-14, pp. 2469–2481, 2006.