Wireless Communications and Mobile Computing

Communications and Networking for Mobile Sink in Wireless Sensor Networks

Lead Guest Editor: Ki-Il Kim
Guest Editors: Shuhui Yang and Euisin Lee

Communications and Networking for Mobile Sink in Wireless Sensor Networks

Wireless Communications and Mobile Computing

Communications and Networking for Mobile Sink in Wireless Sensor Networks

Lead Guest Editor: Ki-Il Kim
Guest Editors: Shuhui Yang and Euisin Lee

Chief Editor

Editorial Board

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9845850, 2 pages
https://doi.org/10.1155/2021/9845850

Editorial

Communications and Networking for Mobile Sink in Wireless Sensor Networks

Ki-Il Kim,[1] **Shuhui Yang,**[2] **and Euisin Lee**[3]

[1]*Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, Republic of Korea*
[2]*Department of Computer Science, Purdue University Northwest, Hammond, IN 46323, USA*
[3]*School of Information and Communication Engineering, Chungbuk National University, Cheongju, Republic of Korea*

Correspondence should be addressed to Ki-Il Kim; kikim@cnu.ac.kr

Received 19 July 2021; Accepted 19 July 2021; Published 8 November 2021

Wireless sensor networks (WSNs) have opened many new possibilities for emerging applications in event tracking and surveillance. However, when compared to the great research interest in this topic, only a limited number of frameworks have been deployed in the real world due to multihop and long hop delivery over many constraints on both nodes and networks. Currently, mobile sink nodes can be a feasible solution for data collection in an energy-efficient way to reduce the total energy consumption with a small number of hops. Unlike static sink approaches, the mobile sink approach requires a new paradigm to collect data such that the sensed data is correctly delivered to mobile nodes. In this situation, a packet may be delivered to a previous location, or a node needs to keep a packet until a mobile sink moves to a nearby place. In addition to these cases, the movement of mobile sink nodes may lead to many research challenges caused by their frequent path changes.

Despite these challenges, recently, there has been a large amount of research introducing mobile robots or unmanned aerial vehicles as mobile sinks. With the help of these systems, many research challenges can be simplified—however, the energy efficiency problem with respect to their communications remains unsolved. Hence, there is a tremendous need for researchers and engineers to have a comprehensive knowledge of the latest advances in mobile sink technology. Based on this demand, this special issue contributes to the advances in open technical problems and challenges in communication and networking for mobile sink in WSN, presenting original research articles that tackle the problem of efficient integration of novel solutions with existing mobile sink technologies.

While considering our objectives, the editors believe that this special issue provides a collection of articles on networking techniques in mobile sinks. We have selected seven valuable papers by evaluating several aspects such as relevance to the special issue and novelty of solutions. The topic of these papers is roughly categorized into the following areas: clustering, routing, OFDM, data distribution, and development of testbed.

In the paper entitled "Optimal Clustering in Wireless Sensor Networks for the Internet of Things Based on Memetic Algorithm: memeWSN," M. Ahmad et al. proposed a memetic algorithm (MemA) to decrease the probability of early convergence by utilizing local exploration techniques. In the proposed scheme, a cluster header in the cluster header set is dynamically selected as early as possible to reduce the convergence time. The proposed technique outperforms the existing scheme in the aspects of control message overhead, cluster count, reconstruction rate, and cluster lifetime. J. Park et al. presented a clustering as well as trajectory optimization scheme to improve energy efficiency in the second paper. The authors propose an iterative algorithm to minimize the amount of energy consumed by components of wireless sensor networks. The proposed scheme takes into account the density of sensors and residual battery of sensors when deciding a suitable number of clusters and cluster headers in the first phase. In the phase, the trajectory is optimized by formulating the suitable trajectories of multiple mobile sinks to minimize the amount of energy consumed by mobile sinks.

In addition to the clustering scheme, C. Kim et al. addressed a multipath routing protocol for mobile sinks. The proposed protocol dynamically constructs multipaths along the moving path of a sink with the help of agents. In addition, path shortening and path extending schemes are applied to reduce the impact of the multipath reconstruction interval and neighbor list update interval. Furthermore, S. Li and S. Kang presented new signal constellation pairs for mapping active subcarriers of the zero-padded trimode orthogonal frequency division multiplexing with index modulation (ZTM-OFDM-IM) systems and evaluated the performance in bit error rate.

The paper is about sensor clouds system, entitled "OEDDBOS: An Efficient Data Distributing Strategy with Energy Saving in Sensor-Cloud Systems." The authors addressed energy consumption for data dissemination issues and presented data distribution with delay demand. To achieve the research objectives, continuous monitoring for the variation of the channel quality as well as the decision for stopping time is introduced to maximize the expected reward of energy efficiency. The final paper is for a testbed system in Internet of Things (IoT). In this work, the authors present an IoT testbed system which can be used to run the experimentation of diverse IoT technologies. The system consists of a set of robust hardware components and software modules. The evaluation results for Bluetooth Low Energy (BLE), Zigbee, and 6LoWPAN technologies have been given. The last paper is for another deployment issued, entitled "Research on the Difficulty of Mobile Node Deployment's Self-play in Wireless Ad Hoc Networks Based on Deep Reinforcement Learning." H. Wang et al. addressed the fast convergence of deep reinforcement learning for node deployment in wireless networks. Dynamic updating learning rate and the method of selecting the latest model to generate sample data are proposed for AlphaZero algorithm.

Conflicts of Interest

I declare that none of the guest editors have a conflict of interest.

Acknowledgments

As guest editors, we appreciate all authors, reviewers, and editorial members for their invaluable contribution. Without their hard work and dedication, it would not have been possible to select these high-qualified papers within the given time limit of this special issue.

Ki-Il Kim
Euisin Lee
Shuhui Yang

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4361650, 13 pages
https://doi.org/10.1155/2021/4361650

Research Article

Research on the Difficulty of Mobile Node Deployment's Self-Play in Wireless Ad Hoc Networks Based on Deep Reinforcement Learning

Huitao Wang,[1] Ruopeng Yang,[1] Changsheng Yin,[1] Xiaofei Zou,[1] and Xuefeng Wang[2]

[1] *College of Information and Communication, National University of Defense Technology, No. 45 JieFang Park Road, Wuhan Hubei, China*
[2] *College of Army Logistics, No. 20 North 1st Road, University Town, Shapingba District, Chongqing, China*

Correspondence should be addressed to Huitao Wang; wangjane@163.com

Received 16 January 2020; Revised 1 February 2021; Accepted 24 February 2021; Published 9 March 2021

Academic Editor: KI-IL Kim

Deep reinforcement learning is one kind of machine learning algorithms which uses the maximum cumulative reward to learn the optimal strategy. The difficulty is how to ensure the fast convergence of the model and generate a large number of sample data to promote the model optimization. Using the deep reinforcement learning framework of the AlphaZero algorithm, the deployment problem of wireless nodes in wireless ad hoc networks is equivalent to the game of Go. A deployment model of mobile nodes in wireless ad hoc networks based on the AlphaZero algorithm is designed. Because the application scenario of wireless ad hoc network does not have the characteristics of chessboard symmetry and invariability, it cannot expand the data sample set by rotating and changing the chessboard orientation. The strategy of dynamic updating learning rate and the method of selecting the latest model to generate sample data are used to solve the problem of fast model convergence.

1. Introduction

With the rapid development of artificial intelligence technology, intelligence has become an important direction of the development of various application fields, and machine learning has made significant progress in many fields, especially the success of AlphaGo and AlphaZero technology in Go human-computer game, making machine learning method to solve traditional problems become a new way [1].

Wireless ad hoc network is to build the communication network coverage within the application scenario area according to the communication support requirements of users and application scenarios and provide users with random access communication channels. The main work is to reasonably select the deployment location of mobile nodes and the connection relationship between mobile nodes [2]. Therefore, the deployment process of wireless ad hoc mobile nodes can be analogized to the game playing process of both sides of Go game, mobile game. The nodes and users can be regarded as the black and white pieces of Go, and the grid terrain area in the application scene can be regarded as the board of Go. On this basis, we explore the application of machine learning in wireless ad hoc network and build a deep reinforcement learning model of wireless ad hoc network based on the AlphaZero algorithm, so as to realize the intelligent deployment of mobile node location. The key and difficulty of the method are to generate a large number of high-quality sample data through the continuous self-play of the model. In the formal scenario application, the best deployment probability of mobile nodes can be predicted by sampling the sample data for supervised learning. The important foundation of machine learning comes from the sample data. Compared with the sample collection of chess game data, the data accumulation of wireless ad hoc network and other application fields is less, and the amount of data available is limited. After the model algorithm is built, the focus is how to generate a large number of high-quality and type rich data samples through the model of self-play to guide the model optimization and realize the best prediction of the deployment location and networking of mobile nodes in practical application.

2. Analysis of Traditional Algorithm Model

At present, for small and typical wireless ad hoc networks, it can be realized by mature network topology automatic planning technology; for large and heterogeneous mesh networks and wireless ad hoc networks under special application scenarios, it mainly uses network planning tool software to carry out static planning in advance and dynamically fine tune and optimize the adaptation in practical application.

2.1. Review of Traditional Algorithms. At present, the algorithm model of multiobjective heterogeneous wireless network mainly abstracts the connection relationship between mobile nodes and nodes into points and lines. Based on graph theory, it simulates, analyzes, and optimizes by setting approximate assumptions and constraints and designing various channel models, traffic models, wireless models, and link models. It mainly includes 4 categories: (a) Optimize design algorithms around network hierarchical structure, topological structure, etc., and design and improve algorithms for different types of network structures such as mesh, tree, and star. For example, for the problem of dense and dense network structures, a K-means-based algorithm is the proposed random graph topology generation algorithm and hierarchical structure topology generation algorithm; for the construction of wireless sensor network plane topology structure, a network topology optimization algorithm based on a Voronoi diagram is proposed; for multihop packet wireless network structure, the shortest route tree table based on node switching is adopted. The method updates and optimizes the network topology. Aiming at the problem of network structure loss, taking the node degree and connectivity as constraints proposes a hierarchical network topology planning algorithm. In addition, it also focuses on network node mobility and network connectivity, links one or more indicators such as reliability and network capacity, sets approximate assumptions and constraints, and analyzes and studies the network topology by constructing various channel models, traffic models, mobility models, and link models. Typical algorithms include the Minimum Spanning Tree Algorithm (MST), Shortest Path Algorithm (SPT), Delaunay Triangulation Algorithm (DT), and Voronoi Diagram Algorithm [3, 4]. (b) Construct a multiobjective optimization function that focuses on multiobjective optimization combination algorithms that focus on user communication requirements, network coverage, deployment costs, network service quality, and other specific communication requirements. For example, around network connectivity, network fault tolerance, network throughput, and other index requirements, respectively, construct a wireless mesh network topology control model based on the conflict domain; around the wireless mesh backbone network topology throughput, propose a minimum spanning tree and conflict load joint topology control algorithm; propose a network topology planning method based on probability statistics based on indicators such as network coverage and network connectivity; based on network index systems and weights, propose a network topology planning method based on performance and effectiveness evaluation [5]. (c) Optimize artificial intelligence methods and strategies, focusing on solving multiobjective heterogeneous wireless network planning problems to build algorithm models, mainly including optimized search space algorithms, random search algorithms, intelligent algorithms, and improvements and combination algorithms based on the above algorithm models. For example, based on the heuristic search algorithm model, the network nodes and links are designed with the goals of optimizing the path loss in the wireless link and optimizing node deployment; abstract the node deployment problem in the network topology as the K center point problem in geometric mathematics. Optimize the links between nodes by constructing an improved particle swarm algorithm model; optimize network connectivity and network coverage by constructing simulated annealing algorithm models and genetic algorithm models; optimize wireless mesh network nodes by constructing an ant colony algorithm model. Optimize deployment; use the tabu search algorithm model to design a global optimization combination strategy and taboo table for the deployment of wireless mesh network nodes to achieve the global optimization of the network topology. Through a greedy algorithm, simulated annealing algorithm, tabu table algorithm, heuristic search algorithm, ant colony algorithm, particle swarm optimization algorithm, genetic algorithm, etc., there is intelligent algorithm optimization, improvement, and reorganization [6]. (d) Draw lessons from the concepts of complex networks, super networks, and fields in physics, and explore and study cross-domain cross-combination algorithms around the importance of network nodes and edges, information, and interaction between nodes. For example, the concepts of field and hypergraph in physics are introduced into complex networks and hypernetworks, and the network topology is optimized through the process of information interaction between nodes. In addition, in the application of artificial intelligence methods, it is proposed to use deep learning to intelligently plan wireless ad hoc network topology [7, 8].

2.2. Shortage of Traditional Algorithm. The static preplanning method is often used to optimize the topology design algorithm. The model is set for the fixed scene; the index system is greatly constrained by the conditions and cannot be adjusted flexibly and dynamically according to the scene changes effectively. (a) The accuracy of the network cannot be quantitatively evaluated, especially for large-scale network applications. The dynamic adjustment adaptability of the model is not enough, and the planning time is too long. Based on the multiobjective combination modeling method, restricted by the constraints, it can only be implemented according to different communication means or groups. The number of indicators that can be concerned is limited, and it cannot give full consideration to all indicator systems. (b) The accuracy of networking cannot be quantitatively evaluated. In the index optimization decision-making, it is easy to lose one or the other, and the universality is not high. Based on genetic algorithm, artificial bee colony algorithm, particle swarm algorithm, and other artificial intelligence network topology optimization algorithms are mostly aimed at the convergence speed, accuracy, and robustness of the model, which are commonly used in simulation analysis and verification, and the real transformation application is still relatively small. (c) The research of network topology based on complex network theory cannot fully solve the problem of multinetwork interaction in wireless ad hoc networks and cannot fully represent the characteristics of network structure and effectively reflect the function of node type.

3. Model of Algorithm

Based on the analogy between wireless ad hoc network and board games such as Go [9, 10], and referring to the AlphaZero algorithm framework, a deep reinforcement learning algorithm for wireless ad hoc network in typical application scenarios with full visibility is constructed [11].

3.1. Algorithm Principle. According to the principle of the AlphaZero algorithm [12], a deep neural network model $(p, v) = f_\theta(s)$ with parameters θ and the evaluation index system of network system effect are constructed $z \in \{-1, 1\}$. Taking the user location and the location state distribution of mobile nodes as the input s, the deployment location probability p of mobile nodes and the evaluation v of network system effect of wireless ad hoc network are output. As shown in formula (1), MCTS search is used for heuristic search and optimization through the interaction of neural network and MCTS: on the one hand, MCTS is guided to perform heuristic search according to the maximum deployment location probability predicted by a neural network; on the other hand, the maximum location probability predicted by MCTS search reacts on the weight update of neural network and forecasts the current again [13]. The next best deployment location of mobile nodes in grid map is to maximize the similarity between the prediction probability of optional location neural network and the search probability of MCTS and to minimize the difference between the network deployment effect v_t and the deployment success z_t [14].

$$l = (z - v)^2 - \pi^2 \log p + c\|\theta\|^2. \tag{1}$$

Specifically, according to formula (2), the gradient descent method is used to update the parameters θ of the neural network in every t iterations. According to formula (3), the weights of the neural network are updated and the sample data set is optimized (s, p, v). In the formal deployment, the weights of the neural network are optimized by the sample set to guide the prediction of the maximum location probability of mobile nodes, and the wireless ad hoc network with high quality and satisfying requirements is gradually generated. Figure 1 is shown the mathematical model of deploying mobile nodes.

$$\Delta\rho \propto \frac{\partial v_\theta(s)}{\partial\theta}(z - v_\theta(s)), \tag{2}$$

$$\Delta\theta \propto \frac{\partial \log p_\rho(a_t \mid s_t)}{\partial\theta} z_t. \tag{3}$$

3.2. Structure of the Algorithm's Model. The deep reinforcement learning model focuses on designing from key submodels such as input object feature extraction, heuristic search "exploration-balance" mechanism, and reinforcement learning iterative feedback mechanism. On this basis, clarify the model training process to ensure the effective integration of various functional modules and the smooth flow of the algorithm process.

3.2.1. Input. Compared with the neural network structure in the AlphaZero algorithm, the neural network structure of the wireless ad hoc network model under the condition of full visibility takes the binary feature plane as the input, where the optional position is represented by 0, and the nonoptional position or occupied position is represented by 1. The number of planes is reduced to 3: the first plane represents the current position of the user, the second plane is the current deployed position of the mobile node, and the third plane is the constraint condition.

3.2.2. Neural Network Structure. Due to the board space of Go 19 × 19, the number of layers of AlphaZero's neural network reaches 40~80. Considering the computing power of a personal computer, the structure of neural network in this study is simplified compared with that in the Go algorithm. As shown in Figure 2, the neural network is designed as 5~6 layers, and the structure is basically the same as that of the AlphaZero algorithm.

3.2.3. Output. There are two output terminals: the value output terminal generates the maximum probability of mobile node deployment location through Softmax function, and the policy output terminal generates the evaluation value p of wireless ad hoc network effect through Tanh function v.

3.2.4. Model Convergence Mechanism. The function loss entropy maximizes the similarity between the deployment location probability p of the communication guarantee unit and the output probability π in the MCTS and minimizes the difference between the expected value v of the neural network in the successful network topology planning and the MCTS evaluation value. The specific method is as follows: (a) In the process of continuous trial and error and interaction in reinforcement learning, the optimal choice of the deployment position of the communication guarantee unit is optimized to obtain the largest cumulative reward. (b) Generate the maximum probability π of the deployment location of the communication guarantee unit and the evaluation value z of the network topology planning through the MCTS heuristic search algorithm, and act on the neural network to update the parameters. (c) Use the predicted maximum probability p generated by the neural network to guide the MCTS to select the location of the communication guarantee unit with the highest probability for deployment and to collect sample data for the network topology planning scheme that meets the conditions. (d) Construct a sample database based on MCTS

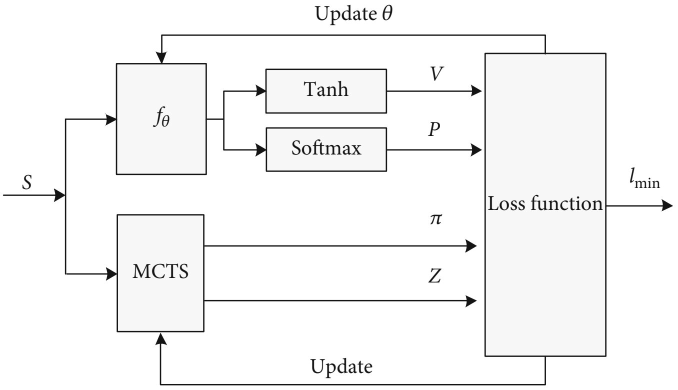


FIGURE 1: The mathematical model of deploying mobile nodes.

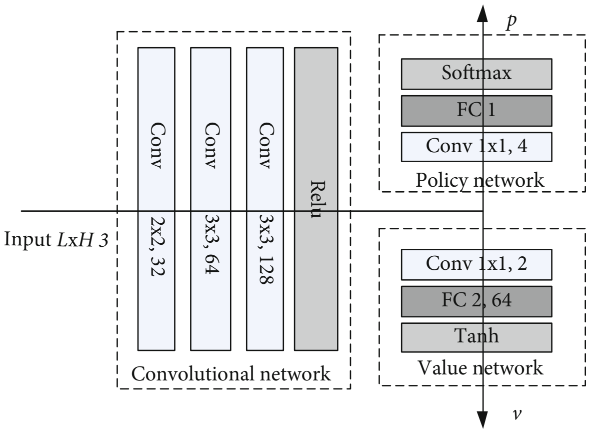


FIGURE 2: Structure of deep reinforcement learning model for wireless ad hoc networks.

sample data, and update the neural network parameters θ through iterative training of sample data.

3.3. Network Effect Evaluation System. The evaluation of Go and other chess is a zero-sum game of win or lose. The evaluation standard of wireless ad hoc network construction not only includes whether the network construction is successful or not but also involves the construction effect. It is necessary to turn the non-zero-sum game evaluation system constraint into a zero-sum game. As a preliminary exploration of intelligent wireless ad hoc network, this study only takes whether the network system can meet the minimum communication of users in the application scenario as the evaluation standard, that is, the communication distance of mobile node workshop must meet $D_{\min} \leq d_{ij} \leq D_{\max}$, among which are the minimum communication distance $D_{\min}$ and the maximum communication $D_{\max}$ distance that can ensure the communication quality and efficiency of wireless node workshop. For redundant channels, deployment costs, and so on, conditions can be set [15].

4. Analysis of Difficulties in Self-Play Training

The key point of self-play is to update the weights of neural network and generate the optimal data through continuous iteration. The key and difficult problems in the process of model training and optimization mainly include three aspects: first, how to optimize the model weight to ensure that the model can converge quickly and achieve effective solution; second, how to ensure that the model can be updated and optimized continuously and prevent overfitting in the process of model optimization; third, how to ensure that a large number of data samples with good quality and various types can be collected [16].

Considering the computing power of a personal computer, the analytic test method of difficult problems in this study is based on the size of 16×16 grid space, and the number of model self-play is set to 1500.

4.1. Parameter Optimization Strategy of Neural Network Based on Dynamic Learning Rate Updating

4.1.1. Setting Learning Rate. In the AlphaZero algorithm, the weight of neural network is updated by the method of random gradient descent in the process of self-play, so that the weight of the model is updated continuously and robust. The learning rate in the random gradient descent directly determines the performance of the model algorithm: the setting of the learning rate is too small, which is easy to cause the model to fall into the local minimum value and affect the convergence speed of the model, resulting in the overfitting phenomenon of the model; the setting of the learning rate is too large, which is easy to cause the model to oscillate near the extreme value, which makes the model unable to converge. Therefore, choosing the appropriate learning rate is the key to ensure the weight optimization and convergence of wireless ad hoc networks.

4.1.2. Method of Update Learning Rate. Learning rate updating methods mainly include the step-by-step reduction method, exponential decay method, and reciprocal decay method. The three methods have specific application scenarios in the field of machine learning [17]. The setting of learning rate of the AlphaZero algorithm adopts the step-by-step reduction method and sets the learning rate step by step according to the increase of training times [1]. Because the number of layers and structure of the neural network in this study is less than that of Go, the method of setting learning

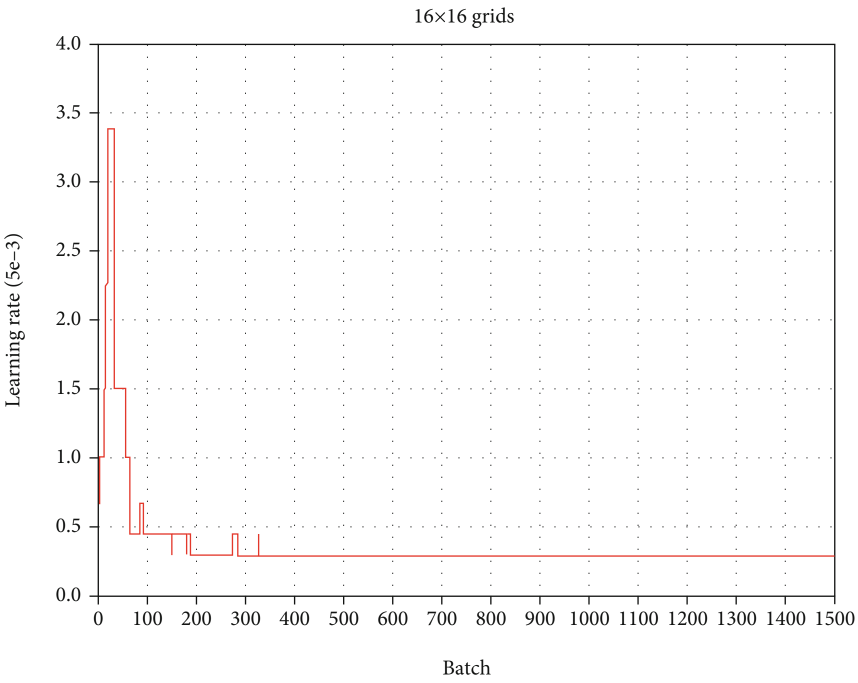


Figure 3: The change of model learning rate.

rate by the AlphaZero algorithm cannot be fully applied in the algorithm model of wireless ad hoc network. Through observation in the first 24 hours of the pretest phase, it is found that the model training has no effective results, and the model convergence is slow. On the basis of adopting the gradual reduction method and setting the initial learning rate to 0.005, increase the weighted value, and set the conditions for changing the weighted value to dynamically adjust the learning rate, as shown in Figure 3.

4.1.3. Method of Model Update. As the Go is a game between black and white chess players, the method to test the weight update of the model neural network is to generate a player according to the current latest model and the historical model, respectively, and judge the outcome of the game between the two parties. If the current latest model wins, the judgment model is updated; otherwise, the judgment model is not updated [17]. However, in wireless ad hoc networks, due to the lack of symmetry between mobile nodes and users, the above-mentioned Go evaluation method cannot be used to determine whether the model is updated. In this study, we consider designing an independent MCTS model as a third party. The players generated by the independent MCTS model do not make any changes except to increase the search depth when they fail in the game. After every 50 sampling training, the players of the model and independent MCTS model will deploy 10 mobile nodes, respectively. By comparing the success/failure ratio of the deployment, we can evaluate whether the model becomes better. If the success/failure ratio of the current sampling model is greater than that of the MCTS model, then the current model is considered to be better. At the same time, the number of searching steps of the MCTS model is increased by 1000. In the next 50 sampling training, the above steps will be continued for evaluation. As shown in Figure 4, 30 comparison results of 1500 local samples in this study can be judged that the model has been optimized 10 times.

4.2. Sample Data Generation Strategy Based on the Optimal Model

4.2.1. Sample Data Generation Analysis of the AlphaZero Algorithm. The model self-play process of wireless ad hoc network mobile nodes is to generate sample data through the model and optimize the model according to the data. The process of model optimization and sample data quality optimization is complementary and closely related. Therefore, on the basis of ensuring the convergence of the model algorithm, ensuring the model optimization can ensure the gradual improvement of the quality of data samples, focusing on the generation of data sample selection.

In the Go game, the AlphaGo algorithm and AlphaZero algorithm adopt two methods to generate sample data: the historical optimal model and the latest model. The method of generating sample data from the historical optimal model needs to add the test procedure of model updating and optimization in the process of model training [18]. For the reinforcement learning process of model optimization with the continuous iterative method, adding the test procedure at the same time will occupy part of computer resources and affect computer computing power and computing efficiency. Using the latest model to generate sample data cannot effectively ensure that the data sample data are generated by the optimal model [19].

4.2.2. Selection of Sample Data Generation. In this study, the sample data is generated from the latest model, and the model optimization inspection cycle is set at the same time.

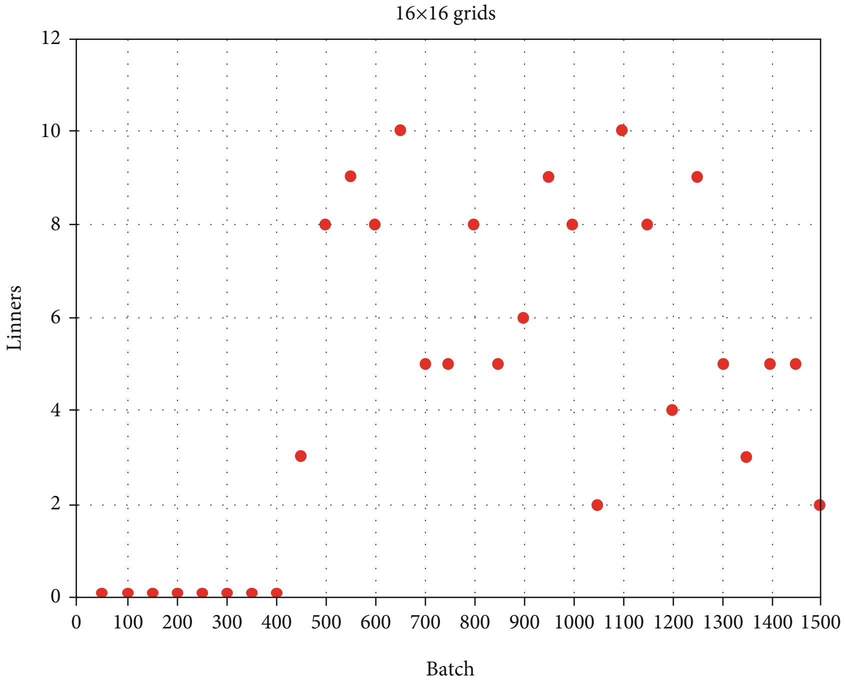


Figure 4: The game winning and losing of the current model and independent MCTS model.

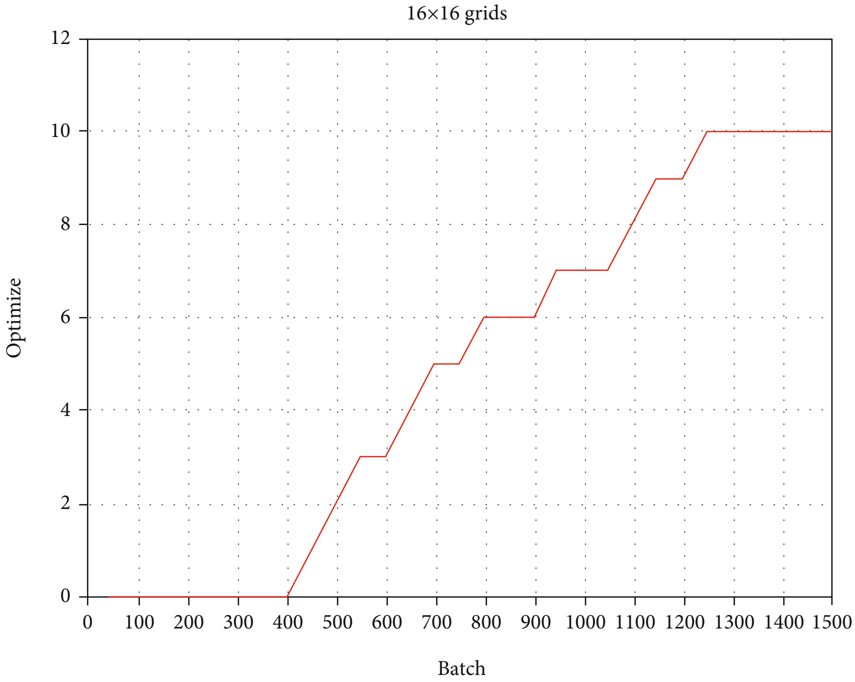


Figure 5: The result of model optimization.

After every 50 samplings, the model optimization is judged. The main considerations are as follows: first, according to the principle analysis of model update in the AlphaZero algorithm, the latest model is generally not worse than the historical optimal model, and the quality of sample data generation is guaranteed. Second, because of the high frequency of update and change of the latest model, the generated sample data is relatively independent, which can improve the coverage of sample data and improve the diversity of sample data and the speed of model convergence. The third is to check the optimization of the model on a regular basis. Instead of comparing the current latest model with the current optimal model every time, it can save training resources and time cost. Figure 5 shows the optimization of the model.

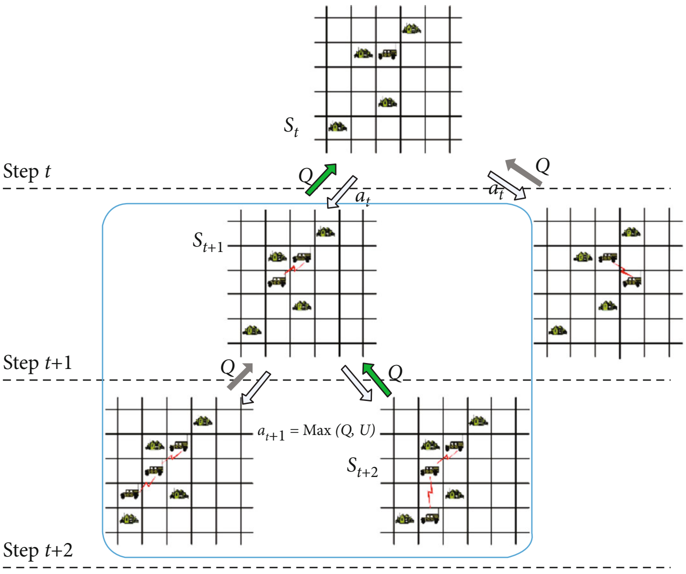


Figure 6: Schematic diagram of the realization process of "exploration-utilization" heuristic search.

4.3. Strategies to Improve the Quality of Sample Data Based on Heuristic Search

4.3.1. Sample Data Expansion Method. At present, the commonly used methods of expanding data set in machine learning mainly include the following: first, collecting more data from the data source; second, resampling the data to obtain more data; third, adopting technical processing to the original data, such as adding random noise to expand the data; fourth, generating new data artificially according to the distribution of data set, such as the AlphaZero series algorithm. According to the symmetry and invariability of Go board, the data sample set is expanded by rotating and changing the board orientation [1, 20].

4.3.2. Sample Data Expansion Analysis. Compared with the AlphaZero algorithm, there are two problems in sample data generation of mobile nodes in wireless ad hoc network: first, the terrain characteristics of the application scene of wireless ad hoc network do not have the image and turning characteristics of Go board, so the number of samples cannot be expanded through board turning. Second, compared with the attributes of Go black and white chess pieces, the attributes of mobile nodes and users in wireless ad hoc networks are different. Therefore, it is not like the game of Go, no matter whether it is black or white; as long as the current situation of chess players winning, the sample data can be collected. The deployment of mobile nodes in wireless ad hoc networks can only collect the sample data of successful deployment of current mobile nodes.

The method of mobile node deployment in wireless ad hoc network is to optimize the MTCS search method by adjusting to maximize the balance of search width and depth. The data sample collection strategy mainly includes three aspects:

(a) Set the appropriate search depth. The search depth in the example is designed as 400 steps to ensure the diversity of deployment location selection of mobile nodes in the current state

(b) Remove some positions that cannot be deployed at all, such as water surface, large obstacles, low-lying, and other positions, and reduce MTCS search space by eliminating these positions that cannot be deployed [21]

(c) Adding noise. By adding noise (Dirichlet noise) [22], expand the search depth, narrow the search scope of MCTS, solve the problem of depth and breadth balance in the search process, and improve the balance of data sample distribution

The mathematical expression of the algorithm is shown in formula (4). Among them is the deployment location selection of the communication security unit at step$t$, and s is the current input state, is the expected value of successful network topology planning, is the deployment location selection probability of the communication security unit, and selects the current deployment of the communication security unit during multiple simulations, the number of times the location is counted.

$$\begin{cases} a = \arg\max \left(Q(s,a) + U(s,a)\right), \\ U(s,a) \propto \dfrac{p(s,a)}{(I + N(s,a))}, \\ W(s,a) = W(s,a) + v, \\ N(s,a) = N(s,a) + I. \end{cases} \quad (4)$$

Figure 6 shows the implementation process of the "exploration-utilization" heuristic search principle in the MCTS in the deep reinforcement learning model of network topology planning. The combination of MCTS and neural network is adopted to consider both the current best

Table 1: The structure of deep reinforcement learning neural network under different grid sizes.

Structure	6 × 6 grid	8 × 8 grid	10 × 10 grid	16 × 16 grid
Input layer	(6,6,3)	(8,8,3)	(10,10,3)	(16,16,3)
Convolutional layer 1	(6,6,32)	(8,8,32)	(10,10,32)	(16,16,32)
Convolutional layer 2	(6,6,64)	(8,8,64)	(10,10,64)	(16,16,64)
Convolutional layer 3	(6,6,128)	(8,8,128)	(10,10,128)	(16,16,128)
Convolutional layer 4	(6,6,4)	(8,8,4)	(10,10,4)	(16,16,4)
Convolutional layer 5	(6,6,2)	(8,8,2)	(10,10,2)	(16,16,2)
Output layer	(36,2)	(64,2)	(100,2)	(256,2)

deployment location selection for network topology planning and consideration, the overall planning result of the network topology. On the one hand, MCTS is used to simulate and evaluate the deployment position of the communication guarantee unit in the next state through multiple iterations. In the case of successful network topology planning and prediction, the location with the most selected times in the simulation evaluation is selected for deployment. On the other hand, MCTS selects the possible deployment locations of other communication security units in a random probability manner, so as to explore and expand the possible deployment locations of communication security units.

5. The Result of Self-Play Training

According to the analysis of the above-mentioned model training optimization method, the model training optimization process is designed to aim at the network adjustment and reconstruction needs after the network is destroyed and interfered by the enemy under the condition of complete visibility. According to the model offline self-training strategy, whether the model is optimized, converged, and resulted in model generation, carry out analysis to test whether the overall architecture of deep reinforcement learning model construction and training process design is applicable.

5.1. Model Initialization. Deep reinforcement learning model training and optimization is a complex and time-consuming process. It is unrealistic for machine learning to directly construct a model training optimization process in a complex environment, or even completely impossible to achieve. Due to limited hardware conditions, the model training optimization constructed in this paper adopts typical application scenarios and appropriately simplifies the depth of the model to verify the model construction method, training process design, and model optimization effect.

(a) Apply background settings. Different from the traditional simulation method, the goal of deep reinforcement learning training is to achieve neural network parameter tuning. The key is to ensure that the model can quickly converge through training and to collect high-quality sample data to improve the fit of the model function. The test standard is the quality feedback of the network topology structure generated by the model, and the application scenarios should be selected with the goal of being able to test the functional effects of the model

Since deep reinforcement learning model training and optimization is a step-by-step iterative optimization process, considering the model training optimization effect and the actual network topology planning, deep reinforcement learning model training optimization is based on the network topology planning under full visibility conditions as the background, and the network is defeated by the enemy. The ability of network topology reconstruction and adjustment in the case of damage and interference is tested for application scenarios. Full visibility conditions refer to the assumptions for the use of tactical Internet organizations in this article. Specifically, the terrain environment requires that the terrain undulates slowly and the terrain fluctuations are within 20 M. Microwaves and ultrashort waves can be regarded as unobstructed, complete visibility propagation and electromagnetic. The environment requires transmission loss and external interference to be within the ideal range, and the communication distance under different communication methods is within the maximum communication distance of the equipment.

(b) Neural network structure setting. According to the overall consideration of model construction in Section 2, the optimization of deep reinforcement learning model training focuses on the effective implementation of the network topology planning model function structure and training process and focuses on the detailed design of input objects, neural network structure, and output functions

According to the neural network structure framework, taking into account the model training speed and optimization effect, the site selection space of the communication guarantee unit is not easy to be too large. Table 1 shows the four grid spaces of 6 × 6, 8 × 8, 10 × 10, and 16 × 16. In the design of the neural network structure in the deep reinforcement learning model, in order to improve the efficiency of the algorithm, the input object image is converted into a binary feature plane instead.

(c) Evaluation method setting. According to the network topology evaluation ideas in this paper, the evaluation

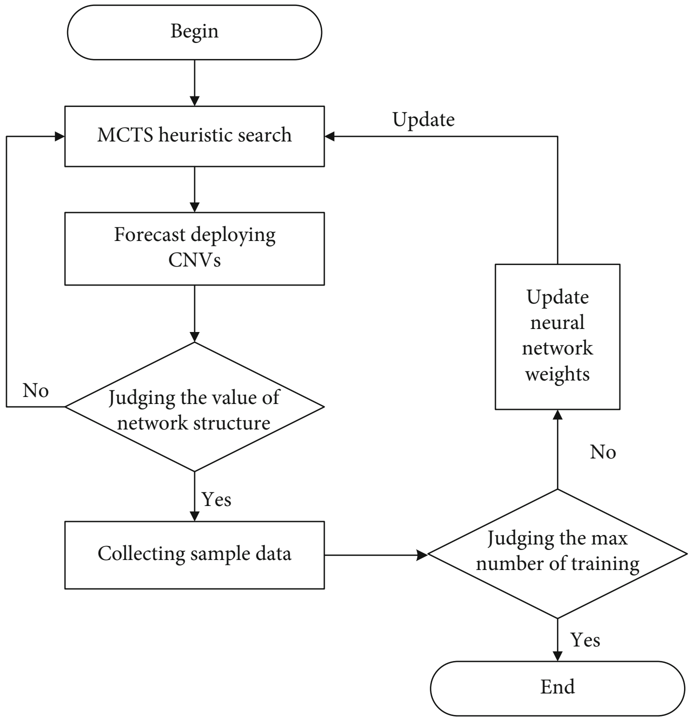

Begin
MCTS heuristic search
Update
Forecast deploying CNVs
Update neural network weights
No
Judging the value of network structure
No
Yes
Collecting sample data
Judging the max number of training
Yes
End

Figure 7: The deep reinforcement learning model training process in network topology planning.

method in the model training optimization is constructed according to three aspects: the connection relationship between network nodes, redundant links, and the connectivity of the whole network

Construction of the connection relationship between node pairs: in each step of the position layout of the communication guarantee unit, the shortest path is calculated according to the Dijkstra algorithm to determine whether the communication distance constraint is satisfied to determine the connectivity between the node pairs.

Whole network connectivity evaluation: according to the communication distance constraint between the communication guarantee units, under the action of the heuristic search algorithm, the connection relationship that does not meet the distance constraint is first removed and then judged according to whether the whole network constitutes a connected graph method.

Redundant link construction: on the basis of judging the connectivity of the entire network, the Prim algorithm is used to generate the minimum spanning tree of the network, and the number of edges between the minimum spanning tree and the connected graph of the entire network is judged.

5.2. Condition Setting

(a) Software and hardware support: CPU 2.3GH, Ubuntu16.04 operating system, Python 2.7 version, development and design program based on the deep reinforcement learning technology framework in AlphaZero

(b) Grid size setting: the network deep reinforcement learning model is trained under four grid spaces of 6×6, 8×8, 10×10, and 16×16. The number of training samples is 1000, and the number of simulations per step of MCTS is set to 400 times

(c) Communication distance constraint setting: in the case of 6×6 grids, the effective communication distance between type 1 users and type 2 users is 1 grid (the distance between type 1 users in this area is already within 1 to 4 grids. In the grid, communication can be achieved). In the case of 8×8, 10×10, and 16×16 grids, the effective communication distance between type 1 users and type 2 users is 1 to 2 grids, and it is effective between type 1 users. The communication distance is 1 to 4 grids

(d) Learning rate setting: through the adjustment results of the learning rate update method such as the gradual decrease method, exponential decay method, and reciprocal decay method adopted in the machine learning in the previous work, the gradual decrease method is selected as the learning rate for model training optimization. The initial learning rate is set to 0.005, and the weighting value change condition is set to realize the dynamic change of the learning rate

5.3. Model Training Optimization Analysis. Model training optimization analysis is carried out in two dimensions, horizontal and vertical, horizontal analysis of model parameters and performance changes under the same grid size, and longitudinal analysis of model performance parameter changes under different grid sizes.

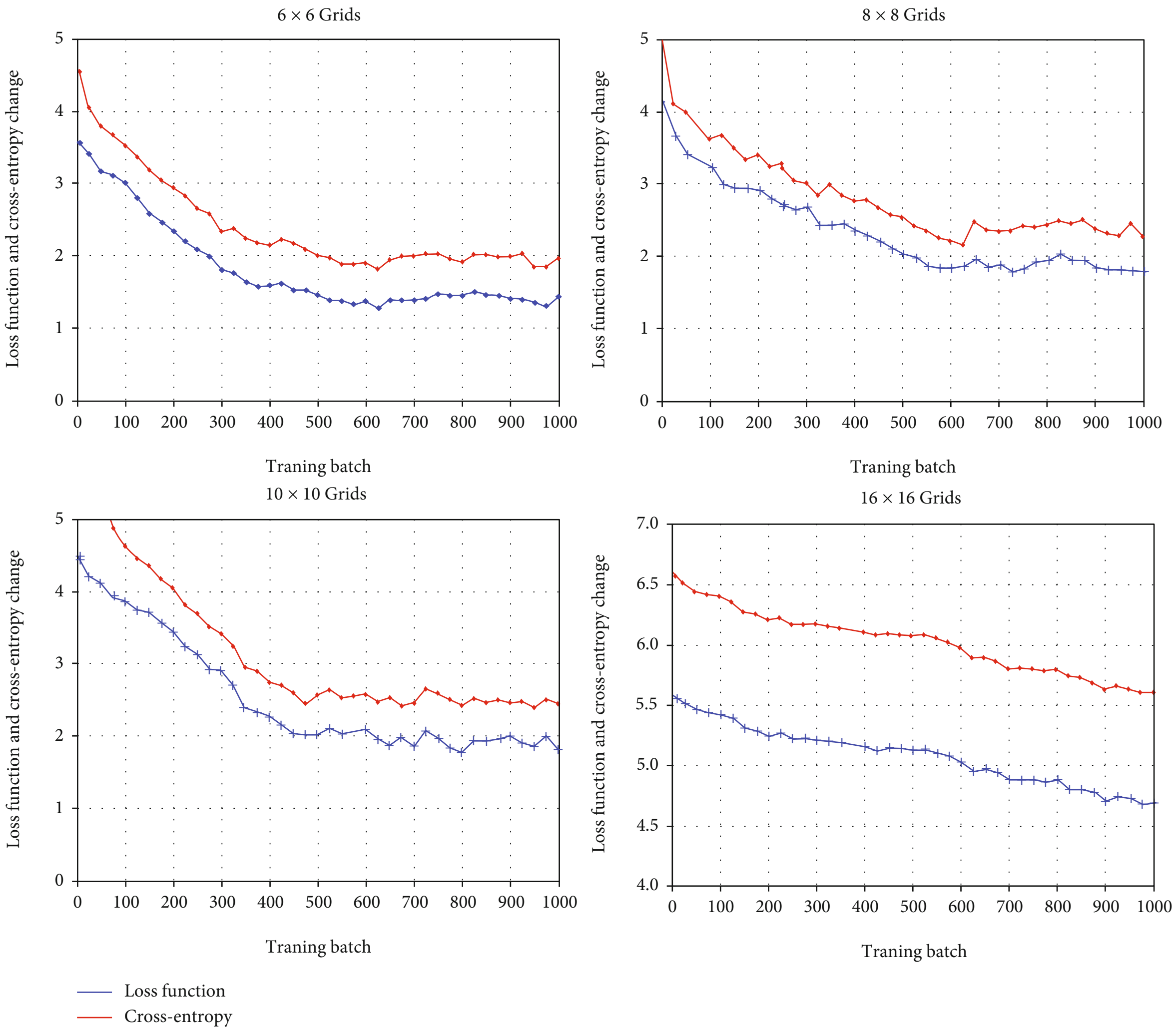


Figure 8: The changes of objective function in deep reinforcement learning model training.

According to the analysis of the model training process, it takes about 24 hours, 60 hours, 72 hours, and 310 hours to complete the training in the four grid sizes of 6 × 6, 8 × 8, 10 × 10, and 16 × 16. It is not difficult to see that as the grid size increases, the time cost of model training optimization increases significantly.

5.3.1. Construction of Training Process. The network topology planning model training optimization based on deep reinforcement learning requires predesigning the connection relationship between the output and the input of each submodule and the flow relationship between the submodules. Figure 7 is a model training process design based on deep reinforcement learning network topology planning.

(a) Neural network model process: perceive input objects through a shared neural network and perform feature extraction. According to the sample data collected by the heuristic search algorithm, the value network and policy network weights are tuned, and the estimated value of the best deployment location of the communication guarantee unit and the network topology structure in the current state is predicted

(b) Heuristic search process: starting from scratch, randomly deploy communication guarantee units in the grid of the optional rasterized topographic map through heuristic search, and collect every step of the plan when a network topology planning scheme that meets the conditions appears, the deployment location of the communication security unit. Sample the collected plan sample data and input it into the neural network, and update the neural network weights and update the optimization model after reinforcement learning iterative training

(c) Reinforcement learning iterative feedback mechanism: use the function loss entropy method to update the selection probability of the deployment location of the communication guarantee unit, and maximize the similarity between the network topology evaluation of the heuristic search algorithm and the value network output value. Within the constraints of the

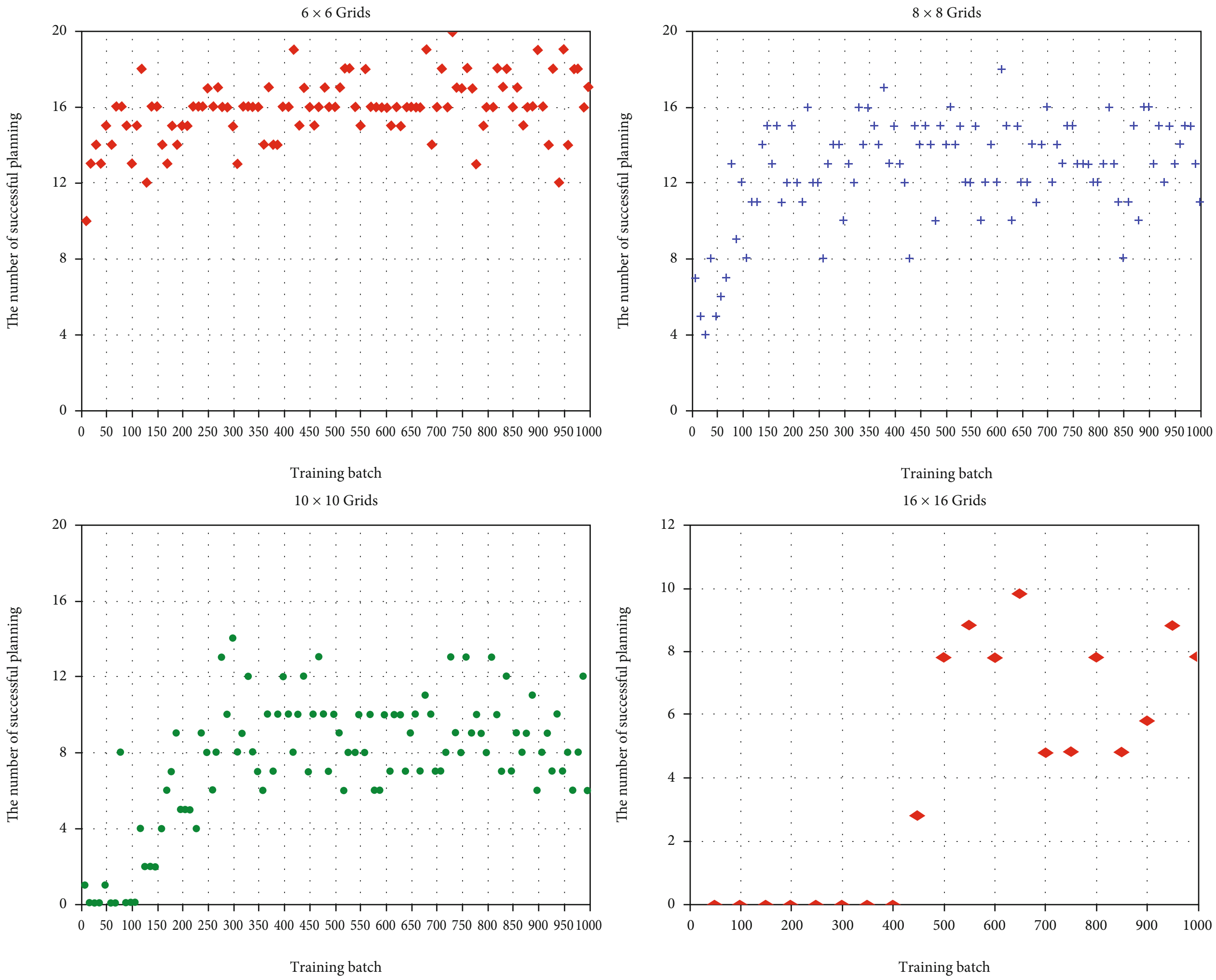


Figure 9: The distribution of successful times of network topology planning under four grid sizes.

maximum number of training times, the heuristic search algorithm process and the neural network model process are continuously repeated to ensure that the results of the strategy network and value network are gradually close to the expected value

Through the nesting and looping of the above submodules, the weights of the neural network are updated and iterated step by step, so as to continuously improve the accuracy and scientificity of the neural network function fit and prediction results.

5.3.2. Model Training. The change of model loss function reflects the construction effect of reinforcement learning iterative feedback mechanism. The change of loss function and cross-entropy reflects the interaction of neural network and heuristic search algorithm. The change of model optimization convergence can be seen through the change of loss function and cross-entropy. Figure 8 shows the changes in the model loss function and the cross-entropy in the strategy network in the four grid sizes in the case of 1000 training samples. The specific analysis is as follows.

Overall analysis: under the four grid sizes, although the loss function and strategy network cross-entropy fluctuate in sample training batches, the overall trend of change is gradually decreasing. It shows that the model training process is designed reasonably and the method is feasible.

Loss function analysis: in 6 × 6, 8 × 8, and 10 × 10 grid sizes, after model sampling training is 500 times, 700 times, and 800 times, the loss function basically remains at about 2.0, 2.4, and 2.5., which shows that the basic training of the model is completed under the above three grid sizes. However, after the model has been sampled and trained 1000 times under the 16 × 16 grid size, the loss function still continues to change and has not stabilized. It can be judged that the model can converge quickly and show a trend of gradual improvement under small grid sizes; as the number of grids increases, model convergence and parameter tuning take longer, and the training effect is not obvious.

Strategy function cross-entropy analysis: the strategy function cross-entropy directly reflects the difference between the actual probability and the expected probability in the training of the communication guarantee unit. Similar to the change of the loss function, the cross-entropy of the function gradually decreases and tends to stabilize under the first three grid sizes, indicating that as the sample sampling batch increases, the neural network predicts the probability of the deployment location of the communication guarantee unit from random. Approximately equal-probability gradually showed a trend of uneven distribution step by step, and the result prediction gradually showed a trend, indicating that the model gradually tends to converge, and the design of the interaction mechanism between neural network and heuristic search in network topology planning is feasible.

5.3.3. Design Aspects of Evaluation Indicators and Evaluation Methods. In this study, we adopted the method of regularly testing the effect of model network topology planning. The effect of model network topology planning was analyzed every 20 training sample batches. Analyze and evaluate the effect of network topology planning to test the effectiveness of the proposed evaluation method. Figure 9 shows that the model is tested for 20 network topology planning results after every 20 sampling training during the model training process. The specific analysis is as follows.

Evaluation index analysis: the network topology planning results under the four grid sizes meet the average network connectivity times of 17/20, 12/20, 10/20, and 6/20. It shows that the model can train the optimization model through the whole network connectivity index and realizes the improvement of the prediction accuracy probability of the communication guarantee unit site selection. It shows that the index decomposition method is adopted and the method of using the whole network connectivity index as the model evaluation index is feasible.

Evaluation method analysis: the effect of the evaluation method in machine learning cannot be directly observed, but the horizontal comparison and analysis of the network topology planning results under the four grid sizes show that the evaluation method is basically feasible and available. At the same time, as the number of grids increases, the number of times that the network topology planning result meets full connectivity gradually decreases, which indicates that the efficiency of constructing the connectivity relationship between node pairs in the evaluation method increases. When the model has a large solution space, the realization of the function requires further improvement methods.

6. Conclusion

Compared with traditional planning methods, the AI deep learning method does not solidify knowledge in an algorithm model but achieves the abstraction and understanding of knowledge through self-learning, reduces the interference of human factors, and improves scientificity. The application of artificial intelligence technology is based on large-scale sample data. In this study, by migrating the AlphaZero algorithm framework, a new mobile node deployment algorithm model under the condition of complete intervisibility is constructed as the specific practice exploration of artificial intelligence in the typical application of mobile wireless ad hoc network. The difficulty and key work are how to generate rich and comprehensive sample data through self-play, so the design of the self-play process is the core and foundation. Referring to the main methods of the AlphaZero algorithm and comparing the different points in the mapping between Go and wireless ad hoc network, this study solves the difficult problems that affect the training sample data generation, such as model optimization, data collection, and model convergence, realizes the migration application of AlphaZero technology in wireless ad hoc network under the condition of full visibility, and provides the next step for the model exploration and application in complex terrain. The basis of the research is given.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] D. Silver, T. Hubert, J. Schrittwieser et al., “Mastering chess and shogi by self-play with a general reinforcement learning algorithm,” 2017, https://arxiv.org/abs/1712.01815.

[2] P. Santi, “Topology control in wireless ad hoc and sensor networks,” *ACM Computing Surveys*, vol. 37, no. 2, pp. 164–194, 2005.

[3] E. Amaldi, A. Capone, M. Cesana, I. Filippini, and F. Malucelli, “Optimization models and methods for planning wireless mesh networks,” *Computer Networks the International Journal of Computer & Telecommunications Networking*, vol. 52, no. 11, pp. 2159–2171, 2008.

[4] H. Kim, E. C. Park, S. K. Noh, and S. B. Hong, “Angular MST-based topology control for multi-hop wireless ad hoc networks,” *ETRI Journal*, vol. 30, no. 2, pp. 341–343, 2008.

[5] A. Noack, P. B. Bok, and S. Kruck, “Evaluating the impact of transmission power on QoS in wireless mesh networks,” in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, Lahaina, HI, USA, 2011.

[6] M. E. Newman, S. H. Strogatz, and D. J. Watts, “Random graphs with arbitrary degree distributions and their applications,” *Physical review E*, vol. 64, no. 2, 2001.

[7] S. Sakamoto, E. Kulla, T. Oda, M. Ikeda, L. Barolli, and F. Xhafa, “A comparison study of simulated annealing and genetic algorithm for node placement problem in wireless mesh networks,” *Journal of Mobile Multimedia*, vol. 9, no. 2, pp. 101–110, 2013.

[8] N. N. G. Le HD, N. H. Dinh, N. D. Le, and V. T. Le, “Optimizing gateway placement in wireless mesh networks based on ACO algorithm,” *International Journal of Computer & Communication Engineering*, vol. 2, no. 2, pp. 45–53, 2013.

[9] O. E. David and N. S. Netanyah, “End-to-end deep neural network for automatic learning in chess,” in *International Conference on Artificial Neural Networks*, pp. 88–96, Cham, 2016.

[10] C. Clark and A. J. Storkey, “Training deep convolutional neural networks to play Go,” in *International conference on machine learning*, vol. 37, pp. 1766–1774, 2015.

[11] X. Zou, R. Yang, C. Yin, Z. Nie, and H. Wang, “Deploying tactical communication node vehicles with AlphaZero algorithm,” *IET Communications*, vol. 14, 2019.

[12] D. Silver, J. Schrittwieser, K. Simonyan et al., “Mastering the game of Go without human knowledge,” *Nature*, vol. 550, pp. 354–359, 2017.

[13] D. Silver, A. Huang, C. J. Maddison et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 2016.

[14] R. Coulom, "Efficient selectivity and backup operators in Monte-Carlo tree search," in *International conference on computers and games*, pp. 72–83, Berlin, Heidelberg, 2006.

[15] C. H. Liu, S. Y. Kuo, D. T. Lee, C. S. Lin, J. H. Weng, and S. Y. Yuan, "Obstacle-avoiding rectilinear Steiner tree construction: a Steiner-point-based algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 7, pp. 1050–1060, 2012.

[16] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[17] D. Silver, L. Newnham, D. Barker, S. Weller, and J. McFall, "Concurrent reinforcement learning from customer interactions," in *International conference on machine learning*, vol. 28, pp. 924–932, Atlanta, GA, USA, 2013.

[18] C. Finn, P. Christiano, P. Abbeel, and S. Levine, "A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models [EB/OL]," 2016, https://arxiv.org/abs/1611.03852.

[19] V. Mnih, A. P. Badia, M. Mirza et al., "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, vol. 48, pp. 1928–1937, New York, NY, USA, 2016.

[20] S. Loffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, vol. 37, pp. 448–456, Lille, France, 2015.

[21] D. Perez, P. Rohlfshagen, and S. M. Lucas, "Monte-Carlo tree search for the physical travelling salesman problem," in *European Conference on the Applications of Evolutionary Computation*, pp. 255–264, Berlin, Heidelberg, 2012.

[22] A. Doerr, N. D. Ratliff, J. Bohg, M. Toussaint, and S. Schaal, "Direct loss minimizaton inverse optimal control," *Molecular Ecology*, vol. 23, no. 10, pp. 2602–2618, 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8875950, 14 pages
https://doi.org/10.1155/2021/8875950

Research Article

Optimal Clustering in Wireless Sensor Networks for the Internet of Things Based on Memetic Algorithm: memeWSN

Masood Ahmad,[1] **Babar Shah**,[2] **Abrar Ullah,**[3] **Fernando Moreira**,[4] **Omar Alfandi**,[2] **Gohar Ali**,[5] **and Abdul Hameed**[6]

[1]*Department of Computer Science, Abdul Wali Khan University, Pakistan*
[2]*College of Technological Innovation, Zayed University, Abu Dhabi, UAE*
[3]*Heriot-Watt University, UK*
[4]*Head of Science and Technology Department, Universidade Portucalense, Portugal*
[5]*Department of Information Systems and Technology, Sur University College, Oman*
[6]*Department of Computing and Technology, Iqra University, Islamabad, Pakistan*

Correspondence should be addressed to Babar Shah; babar.shah@zu.ac.ae

Received 3 July 2020; Revised 2 December 2020; Accepted 16 December 2020; Published 6 January 2021

Academic Editor: Ki-Il Kim

In wireless sensor networks for the Internet of Things (WSN-IoT), the topology deviates very frequently because of the node mobility. The topology maintenance overhead is high in flat-based WSN-IoTs. WSN clustering is suggested to not only reduce the message overhead in WSN-IoT but also control the congestion and easy topology repairs. The partition of wireless mobile nodes (WMNs) into clusters is a multiobjective optimization problem in large-size WSN. Different evolutionary algorithms (EAs) are applied to divide the WSN-IoT into clusters but suffer from early convergence. In this paper, we propose WSN clustering based on the memetic algorithm (MemA) to decrease the probability of early convergence by utilizing local exploration techniques. Optimum clusters in WSN-IoT can be obtained using MemA to dynamically balance the load among clusters. The objective of this research is to find a cluster head set (CH-set) as early as possible once needed. The WMNs with high weight value are selected in lieu of new inhabitants in the subsequent generation. A crossover mechanism is applied to produce new-fangled chromosomes as soon as the two maternities have been nominated. The local search procedure is initiated to enhance the worth of individuals. The suggested method is matched with state-of-the-art methods like MobAC (Singh and Lohani, 2019), EPSO-C (Pathak, 2020), and PBC-CP (Vimalarani, et al. 2016). The proposed technique outperforms the state of the art clustering methods regarding control messages overhead, cluster count, reaffiliation rate, and cluster lifetime.

1. Introduction

Wireless sensor network-enabled Internet of Things (WSN-IoT) is the set of WMNs capable to share data with their neighbors. WSN-IoT can be used for managing different applications such as rescue, flood monitoring, monitoring a border between two states, managing a disaster, and communication in the battlefield among others. The clustering techniques commence awesome once the size of WSN-IoT turns into a massive network in comparison to flat WSN irrespective of routing structure implemented [1]. The scalability problem in flat WSN is a serious concern when a big total of mobile knots are moving in different directions. When the number of WMNs in WSN-IoT using a flat routing arrangement is x, then the complexity of proactive routing structure will be $O(x^2)$ [2]. While the number of WMNs in WSN-IoT grows, the routing overhead also increases with the ratio of the square the number of WMNs. The reactive routing algorithms similarly result in route setup delay once the number of WMNs in a WSN-IoT increases. Furthermore, the flooding route request packets fear may similarly rise which further slowdowns the network. Hence, to accomplish elementary performance assurance in sized WSN-IoT, a hierarchal structure is compulsory [3]. The common execution of a hierarchal strategy is the clustering organization. Cluster formation is a challenging task in designing the cluster-based routing schemes since the choice of optimal CH-set is an NP-hard problem [4].

Planning a clustering structure to route QoS information is the main requirement of the WSN-IoT study. Clustering is the main prototype, and its significance can be listed in two ways. Primarily, WSN-IoT management can be accomplished meritoriously thru clustering structure. An ordinary WSN-IoT contains above a hundred or even a thousand WMNs. In flat WSN-IoT configuration, unneeded packets are transported from source to sink nodes [5]. The scalability problem may possibly rise with flat-based WSN when we want to increase the number of WMNs in WSN and may saturate the network. The WMNs in WSN-IoT may move, and controlling the scalability is extra stimulating in contrast to static networks. Hence, cluster-based routing can be used for the effective management of WSN-IoT. Moreover, clustering will help answer queries such as topology control, building a virtual network, and intrusion discovery [6].

One of the arrogant design concerns of a cluster-based routing algorithm is finding an ideal CH-set that is supposed to shelter the whole WSN-IoT network area. Each time a WMN is associated with a cluster, however, it is not compulsory that all clusters have a CH. Since the presence of a CH in a cluster has the advantage of managing the WSN-IoT easily, the majority of the existing work assumes the presence of CHs in each cluster. The construction of clusters may be performed in a manner that should reduce the message overhead that occurred during the cluster formation phase. If not, the proposed cluster-based routing scheme will consume more energy as compared to the flat routing protocol. Discovering optimal CH-set may be carried out using optimization algorithms, i.e., evolutionary genetic algorithms (GA), neural networks, and particle swarm optimization (PSO) [7]. The particle swarm optimization (PSO) scheme is not suitable because PSO falls easily to optima when the space is high dimensional. PSO also has a low convergence rate in the iterative process. The bee colony optimization lacks the use of secondary information. It also needs novel fitness checks on new algorithm parameters. A higher number of objective function evaluation is essential in BCO. The algorithm may perform slowly in serial processing. The genetic algorithm is very slow and is not suitable for WSN where the resources are limited.

In this article, the WSN-IoT is disseminated to multiple subgroups known as clusters with an evolutionary MemA. The research question of the clustering issue is assumed as a graph plus the suitability model is verified as per the need of WSN-IoT. In the suggested memeWSN, the algorithm chooses a CH-set. The CH-set indicates a chromosome (optimum elucidation). The worth of the optimal solution is enhanced using a confined search technique. Moreover, the CH-set outcome is estimated using a fitness function (FF). The paternities nominated are intended for the replica on the basis of fitness value (FV). Mutation in addition to crossover can be used to bring diversity in the population and to generate different offsprings. The new-fangled chromosomes are produced as well as verified till the ideal CH-set originate. The effectiveness of the clustering procedure is improved using MemA as exposed in simulation results. The experimental results prove that the advised technique devises prominent performance while matching with state of the art clustering schemes.

The remaining article is structured as follows: in Section 2, we argue the relevant research presented in recent years. Section 3 presents our proposed system designed on the basis of MemA. In Section 4, the formulation of WSN-IoT using MemA is presented. In Section 5, we conclude the article, and the future research directions are stated.

2. Literature Review

The main goal of the proposed method is to reduce the energy depletion during the design of the cluster. In [8], a clustering procedure has been proposed which works in a distributed way termed distributed CH scheduling (DCHS) algorithm to enhance the lifetime of WSN. The key parameter of node mobility and a large number of neighbors are not considered in the scheme as well as in the selection of CH. Furthermore, the important parameters such as communication load, reputation, and trust are passed over as well in the paper. The authors in [9] selected CHs randomly in the first round where the criteria in the second round for CH selection is based on residual energy. However, the result of the arbitrary choice of CH-set in the 1st phase causes a derange partition. The movement and degree of MNs for the duration of the CH-set selection is also snubbed.

The mobility of sensor nodes is key to consider during the selection of CHs to increase the lifetime of WSN-IoT. The node motion may be in random directions or may practice a different movement configuration. In leader-based group routing, the WMNs are assembled into a group under the umbrella of a leader [10]. Each group in the network assumes a different element to reduce the requests initiated for different resources. In this way, the throughput will increase, and routing overhead will reduce. The separate consignment of resources to each group is made to minimize the influence of group dissimilarities on the efficiency of WSN-IoT. In [11], the mobility of WMNs is considered major criteria to select the CHs. The distinguished feature of a WMN is its lower mobility and most suitable node to perform the CH role. A weighted cluster-based scheme is assumed to accomplish the determination of CHs and the association of their member nodes. The WMNs with minimum motion are the top contestants for the CH role. The proposed algorithm may not execute fine while the nodes in WSN are moving with fast speed. The WMNs with little speed will detach eventually, and the cluster formation method will initiate repeatedly which will decrease the lifetime of WSN. Correspondingly, the low mobility WMN selected as CH on the other hand with a dissimilar track than its neighbors may result in connectivity interruption to their sisters. The CHs are selected without considering the node degrees. We suggest consideration of some or all parameters such as residual energy, node neighbors, and comparative mobility during the cluster formation.

The mobility of WMNs is predicted accurately to handle the topology changes due to the high mobility proposed in [12]. The scheme forms a cluster and maintains it based on predicting the future mobility of nodes instead of a fixed geographical partition. In contrast to other WMNs in MANET, the WMNs with low speed is the ideal candidate for the cluster head role. The movement pattern of long time neighbor WMNs is used to calculate the movement ratio of a WMN. The WMNs with high probability demonstrate that the WMN is moving alongside its neighbors at equal speed and pattern or is moving with slight motion. It illustrates that a WMN will assist its neighbors for an extended duration and is a better nominee to become a cluster head. The objective of the scheme is as follows: due to mobility, the clusters that are highly resistant to topological changes are formed with the right prediction of the WMN mobility. The prediction of mobility is made on the basis of the WMN location relative to their neighbors. The location is checked over different intervals, and therefore, no special hardware is required for the purpose. The accurate calculation is possible if we consider and check the correlation of WMNs moving in the vicinity. The WMNs with relative mobility and high degree are ideal candidates to form stable clusters. The performance of the protocol may degrade when the mobility model is random and may increase control messages overhead. It is difficult to predict the future mobility of WMNs in many scenarios. Neighbor's quality is ignored. Local changes in topology may initiate the reclustering procedure but due to dynamic topology, the WMNs may not join the same cluster in the next round.

In the dynamic genetic algorithm-based clustering, the load balance issue is initially modeled to a dynamic optimization problem [13]. Using various diverse dynamic GAs established for dynamic optimization is suggested to resolve the problem of balance cluster formation in a network. The capability of a CH set is evaluated on the load balance matrix, and each individual represents a clustering structure. To assist the population deal with variations in topology plus recommend closely related better solutions, several approaches are suggested. To handle the ecological dynamics, some multipopulation approaches, memory, immigrants, and the combination of all these are combined into SGA. The authors talk over several structures to tackle DLBP; however, it is not clear in the paper how the proposed schemes will be applied. The pseudocode stated in the paper is very basic. The proposal assumes the degree of a node to become a cluster head, and the performance may be unsatisfactory in several setups. If the WMNs are assigned the cluster head role on the basis of high mobility, it may perform poorly than flat routing protocols. The reclustering is initiated when the topology changes; it results worst if the topology does not change because the battery of the cluster head will quickly drain.

The authors of [14] proposed an optimization algorithm which finds the optimal clusters in multiobjective fashion to efficiently manage the resources of the network. With optimal clusters in MANET, the energy efficiency will increase by efficiently organizing the resources and the CHs are assigned the task of intracluster and intercluster communication. As an alternative to allocating a weight value to all parameters, it deals straight with the DOP so as to find the Pareto optimal solutions. Less cluster heads will reduce the hop count besides packet delay in a cluster-based routing protocol. By minimizing the number of clusters, we can reduce the routing cost of a packet. The degree of WMNs is not taken into consideration during the cluster head selection process. The ideal candidates for the role of CHs are nodes having high broadcast power, and the CHs may be selected from one part of the network. In this scenario, the energy consumed on cluster formation will be wasted. The precomputation of WMN energy dissipation may not be practical; this parameter may be considered as the remaining energy of a node. The key parameters, i.e., node speed and mobility model, are not measured in the simulation.

Mobility Aware Energy Efficient Clustering for Wireless Sensor Network (MobAC) is presented in [15]. The paper suggests a cluster-based routing algorithm to enhance the lifetime of WSN. The node distance, its lingering vitality, and mobility are assumed during the CH selection process. To further reduce the energy consumption, the paths that are energy-efficient, stable, and short are selected for packet forwarding. The degree and residual energy of nodes is not considered during the CH selection process. The CHs may be selected from one part of the network, and the reaffiliation rate may increase.

A Proficient Bee Colony-Clustering Protocol to Prolong Lifetime of Wireless Sensor Networks (PBC-CP) is discussed in [16]. The nodes selected as CHs have more burden as compared to ordinary nodes in WSN. The CHs are selected based on the distance of a node from the base station, its degree, and energy. The energy-efficient paths are used to transmit data to other clusters and base stations. The mobility metric plays an important role in the selection of CHs. The PBC-CP ignores the mobility of nodes during the CH selection. The nodes selected as CHs with different mobility than their neighbors may cause reclustering more frequently. Calling the reclustering procedure more repeatedly may result in unstable clusters, and the network lifetime may decrease.

An Enhanced PSO-Based Clustering Energy Optimization Algorithm for Wireless Sensor Network (EPSO-C) uses particle swarm optimization to select the CHs and minimize the energy consumption [17]. In this approach, the particles are optimized to get the most optimal CHs. The CHs are selected based on their location, and the CH-set will cover the whole network. The location of nodes is used to select the CHs in WSN. The static nodes are assumed, but in the current and future networks, most of the nodes will be mobile or even the nodes will fly. The suggested method may fail if the nodes deployed in the network move from one location to another. The node degree and energy are also ignored in this scheme.

The strict nature of WMNs in WSN w.r.t energy and communication range make it challenging to communicate the identifying information on time with minimum delay. The routing problems in addition to small network time may also rise because of the WMN limited energy. The WMNs near the base station carry a substantial weight of information transmitted on behalf of other WMNs located nearby. The main goal of this proposal is to select the CHs from optimal sites, and the gateway WMNs will be chosen from all clusters based on their position. The presence of gateway nodes will reduce the burden on CHs. The gateway WMNs are responsible to transmit information intercluster and intracluster. This way, the load on the CHs will be balanced. The exposure of CHs will rise since the gateway WMNs transmit information with adjacent clusters on behalf of CHs.

To overcome the issues found in the literature, the CHs in WSN-IoT are selected based on multiple parameters such as WMN current energy, degree, and mobility. The current energy consideration increases the cluster lifetime. The balanced clusters are obtained by considering the WMN degree. To reduce the riffle effect of reclustering, the mobility of WMNs is considered during the cluster formation. The MemA is also applied to obtain the optimal CHs. The issue of premature convergence will not arise due to the local search technique embedded in the MemA.

3. WSN-IoT Clustering Problem Formulation

To model the matter of allocating the WSN-IoT into clusters, k clusters and N number of WMNs in WSN-IoT are assumed. In the proposed memeWSN, the model is first demonstrated followed by the formulation of the dynamic and optimum clustering in WSN-IoT. The difficulty of stable cluster development of n WMNs is essentially resolved by discovering a set of WMNs (CH-set) which shows the division of n WMNs into k nonoverlapping clusters (C1, C2, ⋯, Ck). This can be achieved by modeling the WSN-IoT as a graph $G(V, E)$, where V represents the vertices of mobile nodes and E represents the edge of the communication links between the WMNs in the graph.

A set of vertices (CHs) is selected optimally based on the FV of WMNs for the CHs role. The CH set with higher weighting values which covers most part of the WSN-IoT has a higher fitness value.

The following equation (1) calculates the weighting factor (WF) value of a WMN i.

$$W_{\text{node}_i} = WEnr_{\text{Node}_i} + WNgh_{\text{Node}_i} + WMob_{\text{Node}_i}, \tag{1}$$

Here, $WEnr_{\text{Node}_i}$is the WF concerning the residual power of a WMN i calculated as:

$$\text{AE}_{\text{Node}} = \frac{\sum_{i=1}^{n} R.Enr_{\text{Node}_i}}{n}. \tag{2}$$

$R.Enr_{\text{Node}_i}$ is the remaining power of a WMN i, n is the total WMNs in WSN-IoT, plus AE_{Node} is the average power of WMNs at present time. The WF with regard to energy may be extracted as:

$$\begin{gathered}
\text{if } R.Enr_{\text{Node}_i} > \text{AE}_{\text{Node}} \text{ then}\\
WEnr_{\text{Node}_i} = 1\\
\textit{else} \text{ if } R.Enr_{\text{Node}_i} \approx \text{AE}_{\text{Node}} \text{then}\\
WEnr_{\text{Node}_i} = 0\\
\textit{else}\\
WEnr_{\text{Node}_i} = -1
\end{gathered} \tag{3}$$

The WF regarding energy will be 0 if the remaining energy of WMN "$R.Enr_{\text{Node}}$" is nearly equal to the average energy (≈) "$\text{AE}_{\text{WSN-IoT}}$" of WSN-IoT. $WNgh_{\text{Node}_i}$ represents the weight factor with respect to neighbors of a WMN. The weight value $WNgh_{\text{Node}_i}$ can be assigned based on the inner and outer degree of WMNs for instance:

$$ANgh_{\text{Node}} = \frac{\sum_{i=1}^{n} (\text{OuterDeg} + \text{InnerDeg})_{\text{Node}_i}}{n}, \tag{4}$$

where OuterDeg and InnerDeg are the inner and outer degree of WMN i, $ANgh_{\text{Node}}$ is the typical neighbors in WSN-IoT. The WF w.r.t WMN neighbors can be allocated in the subsequent para as:

$$\begin{gathered}
\text{if } (\text{OuterDeg} + \text{InnerDeg})_{\text{Node}_i} > ANgh_{\text{Node}} > \text{then}\\
WNgh_{\text{Node}_i} = 1\\
\text{else if} (\text{OuterDeg} + \text{InnerDeg})_{\text{Node}_i} \approx ANgh_{\text{Node}} \text{ then}\\
WNgh_{\text{Node}_i} = 0\\
\text{else}\\
WNgh_{\text{Node}_i} = -1
\end{gathered} \tag{5}$$

Similarly, the WF of WMN regarding mobility $WMob_{\text{Node}_i}$ can be computed as follows keeping in mind that WMN is having relative movements; otherwise, the static WMNs are the best nominees for the role of CH.

$$\begin{gathered}
\text{if } (\text{mobility} \approx \text{relative or static}) \text{then}\\
WMob_{\text{Node}_i} = 1\\
\text{else}\\
WMob_{\text{Node}_i} = -1
\end{gathered} \tag{6}$$

k is the number of clusters would be calculated before selecting the CH-set through the equivalence in equation (7) for calculating the value of k:

$$k = \left\lceil \frac{\sum_{i=1}^{n} (\text{OuterDeg} + \text{InnerDeg})_{\text{Node}_i}}{n} \right\rceil + 1. \tag{7}$$

In the above equation (7), the total cluster in WSN-IoT is k, the total WMNs in WSN-IoT is represented by n, and the $(\text{OuterDeg} + \text{InnerDeg})_{\text{Node}_i}$ represents the neighbor information of node i.

The cluster head set covers WMNs as a minimum 3-hop away from another cluster head. Once the WFs of the whole WSN-IoT WMNs are calculated, the equation designated in equation (8) will be used to calculate the correctness of the

CH-set.

$$\text{Minimize } \operatorname{Fun}(W, \mathrm{AFV}) = \sum_{i=1}^{n}\sum_{j=1}^{k} R\mathrm{Weight}_{ij}\left(W_{\mathrm{node}_i} - \mathrm{AFV}_{\mathrm{node}_j}\right)^2,$$
$$\text{Subject to} \sum_{j=1}^{k}\left(R\mathrm{Weight}_{\mathrm{node}_j} = 1\right) = k \text{ for}(j = 1, \cdots, k),$$
$$R\mathrm{Weight}_{\mathrm{node}_j} = 1 \text{ or } 0, \text{for}(i = 1, \cdots, n \text{ and } j = 1, \cdots, k). \tag{8}$$

The function stated using equation (8) is a minimization function where n is the total WMNs in WSN-IoT, k number of known or unknown (the number of clusters may not be known in advance) clusters will be designed, W_{node_i} $(i = 1,2,3, \cdots, n)$ is the weight of node$_i$, and AFV$_j$ is the average fitness value of WMN to accomplish the CH role. Equation (9) below will compute the AFV$_j$ of a WMN i.

$$\mathrm{AFV}_{\mathrm{node}_j} = \frac{\sum_{j=1}^{k} R\mathrm{Weight}_{ij} \times W_{\mathrm{node}_j}}{k}. \tag{9}$$

The total WSN-IoT clusters designed are k solutions, R Weight$_{ij}$ is the relationship weight of Node$_i$ and cluster j, when the WMN i is allotted to the cluster j, and the value of RWeight$_{ij}$ will be 1 or 0 otherwise.

Once the entire FVs are calculated, the likelihood of choice P_{node_i} in lieu of every single CH can be calculated using:

$$P_{\mathrm{node}_i} = \frac{W_{\mathrm{node}_i}}{\sum_{j=1}^{k} W_{\mathrm{node}_j}}, \tag{10}$$

where k is equivalent to aggregate CHs and W_{node_j} is the WF of WMN j; local search is applied for optimal CH set in the region of node$_j$ as:

$$\mathrm{node}_j(x+1) = \mathrm{node}_j(x) + \alpha_{ij} \times z. \tag{11}$$

α_{ij} is the association of WMN i with cluster j, and z is the randomized variable accepting value in the range [-1, 1] to calculate the FV by equation (11). Touching the outer range of the target WMN is not permitted in the next inhabitants.

The WMNs are assigned to the CH-set vector on the basis of the FF described in equation (11). The neighbor WMNs join the nearby CHs to configure clusters. After the cluster formation, many WMNs change their state to sleep mode where other nodes in the cluster do schedule wake up to save energy. Consequently, periodic alterations will occur in the network topology. The objective of this research is finding a CH-set as early as possible once memeWSN encounters modifications in topology.

4. Memetic Algorithm for WSN Cluster Formation: memeWSN

The proposed MemA-based clustering known as memeWSN for WSN-IoT is demonstrated in this section. The CH-set is represented by a chromosome and is initially selected randomly. The CH-set is checked by a FF to get more optimum results. Thus, the proper individuals are elected for the succeeding peers to generate a new-fangled solution. The WMNs for the reconstruction of chromosomes (CH-set) are nominated from the populace analogous to the conventional genetic algorithm (GA). Subsequently, the choice of binary maternities in lieu of new-fangled populace, crossover in addition to the mutation will be used to make novel offsprings. The novel populace can be enhanced by calling a function designed for local search. In MemA, the indigenous search is applied to proficiently discover a less optimal solution and continue for global optimal.

The proposed memeWSN initiates the aforementioned job through computing each WMN WFs. WMNs having greater weights are designated for the early populace. The FVs of the populace are computed, and WMNs to turn into the CHs are designated. Furthermore, the possibility of WMN election is considered, and local search is used. In the same way, the FV of WMN to stay alive in the populace is computed. The notations used in Algorithm 1 are shown and defined in Table 1. The procedure for the formation of stable clusters using the evolutionary MemA is presented in Algorithm 1.

4.1. Genetic Representation. The classical evolutionary schemes for the example GA fails to explore numerous results of the domain area due to its inborn feature of early convergence. A MemA applies local search to reach its last stop deprived of preventing local maxima. The main stair in memA is coding the chromosome.

The set of cluster heads S_{CHi} of WMNs are erratically nominated from all the WSN-IoT as CHs where $i = 1, 2, \cdots, k$. Every elucidation to the issue is the set of S_{CHi} obtained throughout the selection of CH nodes. In this way, the cluster headset produced using a random permutation of WMN-IDs and the chromosome can be represented by a random set of cluster head IDs. A chromosome is the combination of WMN-IDs without repetition. Furthermore, the gene of a chromosome can be represented by a single node ID. In a network of 8 WMNs, their IDs in the WSN-IoT would be in the range 1 to 8. In such a situation, the chromosome will represent a random combination of WMN-IDs, e.g., 54673128. Once the process of the chromosome is completed, the subsequent job is to extract CH-set. The new nodes/genes are replaced/added to the cluster headset (chromosome) based on their weights. At each step of adding WMNs to CH-set, the weights of CHs will update. The WMN bearing greater weight is substituted using lower weight WMN. The discarded node from the cluster headset is no more considered to be a CH in the existing round. Subsequent to encoding, the fitness of the solution is judged using a fitness function.

```
Procedure memWSN_cluster
Input:$v[n]$, $n$, $k$
Output: WSN-IoT cluster head set
Initialize all variables
Total Deg = 0, $i = 1$.
        while$(i \le n)$do
            TotalDeg = TotalDeg + Deg$_{\text{i}}$
    end while
    $A = \text{TotalDeg}/n$
      $k = A, j = 1(14)$
      while$(j \le k)$do \\ initializes a random cluster head set permutation
a. $\text{CHs}[j] = \text{r and}(v[n])$
b. $j++$;
      end while
      pop = CHs$[k]$
      $p$_new = Local_Improver_WSN_IoT(v[n], pop, $k$)
      $p_{\text{new}}$ = call function replace($p$_new)
          $.v$ [$p$_new] = calculate fitness($p$_new)
      {if($f.v[P\text{\_new}] < f.v[\text{pop}]$)
      CHs = $p$_new}
      Return CHs
  end procedure memWSN_cluster
```

Algorithm 1: Psuedo code of memeWSN.

Table 1: Notations in Algorithm 1.

Symbol	Definition
New – CHs	New CHs
p_new	New population
pop	Population
Deg_i	WMN i degree
fv	FV of CH-set
$v[n]$	WMN ID array
TD	Total of WMN degree
A	Average of WMN degree
k	Total clusters
m	Total WMNs

4.1.1. Objective Function. Manipulating the FVs of WMNs considered for the CH role in the current round using standard deviation can tell us about the quality of a chromosome. The lower the FV, the best is the solution. The FV of a CH-set will be computed as:

$$F(\text{chr}) = \sum_{i=1}^{n}\sum_{j=1}^{k} R\text{Weight}_{ij}\left(W_{\text{node}_i} - \text{AFV}_j\right)^2. \tag{12}$$

In our research, the main objective is to find the set of cluster head nodes in such a way to balance clusters by assigning the cluster head role to those nodes having the max degree, low mobility, and away from other cluster heads. The selection of chromosomes for the next population is carried out after the fitness evaluation. With decent choice, better quality children are conceded to the succeeding inhabitants for reproduction. The chromosome is nominated based on its fitness value. In this research, the mode of selection adopted is a pairwise tournament where nomination will happen without substitution. The tournament size (TS) is fixed, i.e., TS = 02. To prevent premature convergence, a local search procedure is called which mines better quality chromosomes from the population as in Algorithm 2.

4.1.2. Local Improver. In this subsection, an indigenous search is used during the selection and evaluation of a chromosome. The weights of the genes in the population are calculated. The genes with greater weights are explored in the residents. The population is examined for the genes having greater weights and swapped with low-weight genes. Algorithm 2 improves the solution locally.

The significant memetic operators are crossover (which is the offspring generated from two predecessor chromosomes) and mutation (which is children produced from one chromosome by altering a gene). The features of the newly created chromosomes are inherited from all fragments of her maternities. The newly created chromosomes are located in the populace once crossover and mutation are performed. The chromosomes with good features are swapped by low-quality chromosomes.

The newly created inhabitants can be used for the additional execution of the procedure. The procedure is repeated till the ending criterion hits. The top solution is resumed after the final repetition of the memeWSN procedure.

5. Experiment Evaluation

To evaluate the efficiency of the proposed memeWSN clustering scheme, numerous simulation experimentations have been carried out in EstiNet 9.0. In a 1000 m × 1000 m square

```
Procedure Local_Improver_WSN-IoT
Input: vector[WMNs], CH-set, k
Output: CH-set
Initialize variables $i = x = 1$, $n =$ vector length
  While($i \le n$)do
    weight [i] = $\mathrm{WMN}_i$
  end while
  while($x \le n$)do
    for($y = 1 ; y \le k ; y{+}{+}$)
      if (weight[$x$] > weight[$y$]) then
        CH − set[$y$] = vector[$x$]
      end if
    end for
  end while
  return CH-set
  end Local_Improver_WSN-IoT
```

Algorithm 2: Pseudocode of procedure local-improver for WSN-IoT.

```
Procedure Replace_WSN-IoT
Input: populace, CH-set, k
Output: CH-set, populace
Initialize variables $j = p = 1$
  while($j \le k$)do
    New_CH − set[$j$] = rand (vector[WMNs])
  end while
  New_pop[$j$] = new_CH − set[$j$]
  while($p \le k$)do\* mutation and crossover */
        Temp = populace[$p$]
        populace[$p$] = New_populace[$j$]
        New_populace[$j$] = Temp
  end while
  $z = k/2$
  New_populace[$z$] = populace[$z$]
  return new_populace, CH-set
  end procedure replace_WSN-IoT
```

Algorithm 3: Pseudo code of procedure-replace for WSN-IoT.

simulation area, the WMNs (up to 500) are deployed in a random way. The simulation parameters are adopted from [17]. The speed of WMNs varies from 1 km/h to 80 km/h. Every WMN is equipped with a default omnidirectional antenna. The broadcast range of every antenna varies in the range between 100 m and 300 m. Each WMN has a storage (queue) for outgoing and incoming data and maintains the mobility statistics of their neighbors. The traffic sources are generated by a continuous bit rate mechanism. The maximum limit for the generation of traffic is set to twenty (20) packets/s. The simulations are executed for fifty (50) minutes. The average values of a hundred simulation rounds are demonstrated in the form of graphs. Table 2 shows the parameters used during the simulation as in [17].

The performance of the memeWSN algorithm is compared with PBC-CP [16] and EPSO-C [17] for the performance measurement of control overhead and computation. Similarly, MobAC [15] for the performance evaluation of cluster lifetime and reaffiliation. To assess the efficiency of the proposed memeWSN, the metrics discussed below are measured.

The number of clusters (NoC) or cluster count (CC): The WMNs are distributed to several simulated groups known as clusters in cluster-based WSN-IoTs. Here, the NoC or CC denotes the sum of virtual sets gained when executing the cluster foundation procedure. The fewer NoCs represent the stability of WSN-IoT clusters. The issues for instance channel access scheduling, reusing frequency, energy depletion, and latency may arise when the NoCs may increase

Re-affiliation rate (RR): The affiliation and deaffiliation of WMNs as CH or member throughout a certain interval of time represent reaffiliation rate. In cluster-based routing, a WMN links to the adjacent CH and vacates the former. A WMN may reaffiliate once its CH no more survives, and alternative WMN performs the role of a CH in the vicinity. The reaffiliation may also happen once the CH is not within the broadcast range of a WMN. The cluster lifetime will decrease when RR is high and vice versa

Control overhead (CO): The number of packet exchanges to maintain the statistics of topology variations denotes the CO. Several control message (CM) exchanges happen throughout the cluster construction stage. To measure the CO metric, the CMs received/sent at some interval are computed

5.1. NoC or CC. Several tests have been carried out to compute the CC metric once we increase the number of nodes from 50 to 500. The incremental step 50 is adopted to increase WMNs in individual simulation experiments. The random waypoint mobility model is practiced in this set of experiments. The speed of WMNs varies in the range 1 kilometer/h to 80 kilometers/h, i.e., 1 kilometer/-hour to 5 kilometers/hour (walking), 5 kilometers/hour to 20 kilometers/hour (running), and 20 kilometers/hour to 80 kilometers/hour (vehicle) drive. Similarly, the transmission range is 100–200 meters for various experiments. To measure the CC metric, the outcomes derived during

Table 2: Simulation parameters.

Parameters	Value	Parameters	Value
3d beam width	360 degrees	Simulation time	50 min
Pointing direction	90 degrees	Max (x)	1000 m
Angular speed	0 degrees/sec	Max (y)	1000 m
Link bandwidth	11 Mbps	Node space	
Frequency channel	3	Number of nodes	50-500
Bit error rate	0.0	Data packet size	1200 bytes
Frequency (MHz)	2400	Mobility speed	1, 5, 10 m/sec
Transmission power	15 dbm	Shadowing standard deviation	4.0
Carrier sense threshold	57 dbm	Close in reference distance	1.0 m
Antenna gain	1.0 dbi	System loss	1.0
Fading variation	10.0	Antenna height	1.5 m
Average building height	10 m	Ricean factor K	10.0 db
Street width	30 m	Max queue length	50 pkts
Path loss exponent	2.0	RTS threshold	3000 bytes

the simulation are presented in the form of a graph in Figures 1 and 2. The curves presented in Figures 1 and 2 show the NoCs for different size networks (i.e., up to 500 WMNs). The NoCs will increase with the increase in the number of WMNs as presented in Figure 1. The graph shows that memeWSN forms less number of clusters as compared to MobAC, PBC-CP, and EPSO-C. In memeWSN, the NoCs is computed based on node neighbors. The average degrees of WMNs are identified to compute the NoCs. Hence, the memeWSN has less NoCs, and the steadiness of clusters may also be ensured if node degree is in consideration. After memeWSN, PBC-CP performs glowing related to other protocols as presented in Figure 1. Analysing the curves demonstrated in Figure 1, we notice that EPSO-C performs pitiably, and the NoCs generated by MobAC is significantly fewer compared to EPSO-C. MobAC is emphasized on the steadiness of clusters by guessing the expected movement of WMNs throughout cluster creation. EPSO-C focuses on the optimum computations involved in the cluster creation process. The outcomes confirm that memeWSN leaves behind state-of-the-art cluster-based routing protocols w.r.t CC. PBC-CP performs well after memeWSN. To evaluate the performance of memeWSN in more detail, a sequence of simulation trials are conducted for another broadcast range, i.e., 200 meters. The outcomes of the succeeding trials are depicted in Figure 2. The arcs in Figure 2 indicate that the wireless broadcast ranges of WMNs affect the efficiency of cluster-based routing schemes. The CC metric drops greatly with high broadcast ranges. With a high broadcast range, the CH will cover a big region and the number of WMNs CH serve may grow hence results in less NoCs. The magnitude of backbone WSN-IoT will reduce when we rise the wireless broadcast span of WMNs. All protocols perform the same way as in Figure 1.

5.2. Cluster Lifetime. The effect of node speed on the lifetime of CHs is evaluated in this section. The WMNs with high motion may reduce the lifespan of CHs, deploying 100 WMNs in WSN-IoT to execute the simulation. The transmission range of each WMN is fixed to two hundred in the main experiment. The WMNs are moving at a speed varying from 1 kilometer/hour to 80 kilometers/hour where the mobility characteristics are adopted as in [17]. As shown in Figure 3, the proposed memeWSN clustering algorithm selects CHs for an extended period. Steady clusters are designed via a memeWSN clustering scheme because the comparative motion of WMNs and their neighbors is assumed for the duration of CHs choice. The proposed memeWSN outperforms PBC-CP, MobAC, and EPSO-C in terms of cluster steadiness. MobAC performs fine after memeWSN since it takes the future motion of WMNs throughout the cluster creation. The behavior of WMNs may not project genuinely in long term. In memeWSN, the cluster partners stay connected to the CH for an extended period. The comparative motion of WMNs is dignified throughout the CHs election, and a WMN with a higher degree and comparative movement is the top contender to be a CH.

The evaluation of bars presented in Figure 3 demonstrate that memeWSN create steady and long life clusters compared to MobAC. EPSO-C and PBC-CP choose unsteady CHs. The EPSO-C is better as compared to PBC-CP. Thus, PBC-CP selects unsteady CHs once the speed of WMNs increases. The movement of nodes is not taken into consideration throughout the cluster formation process. The wireless broadcast range of WMNs is amplified to 300 meters in the next test, and the outcomes are exposed in Figure 4. The chart demonstrates that the wireless broadcast range of WMNs influences considerably the efficiency of all clustering schemes assumed in this experiment comprising our recommended memeWSN scheme. More steady clusters are obtained when we increase the broadcast range of WMNs. The lifetime of the cluster and network may increase. The suggested memeWSN performs fine once we increase the transmission range. With more broadcast range, the WMNs reaffiliation drops and a CH will handle large-size WSN-IoT.

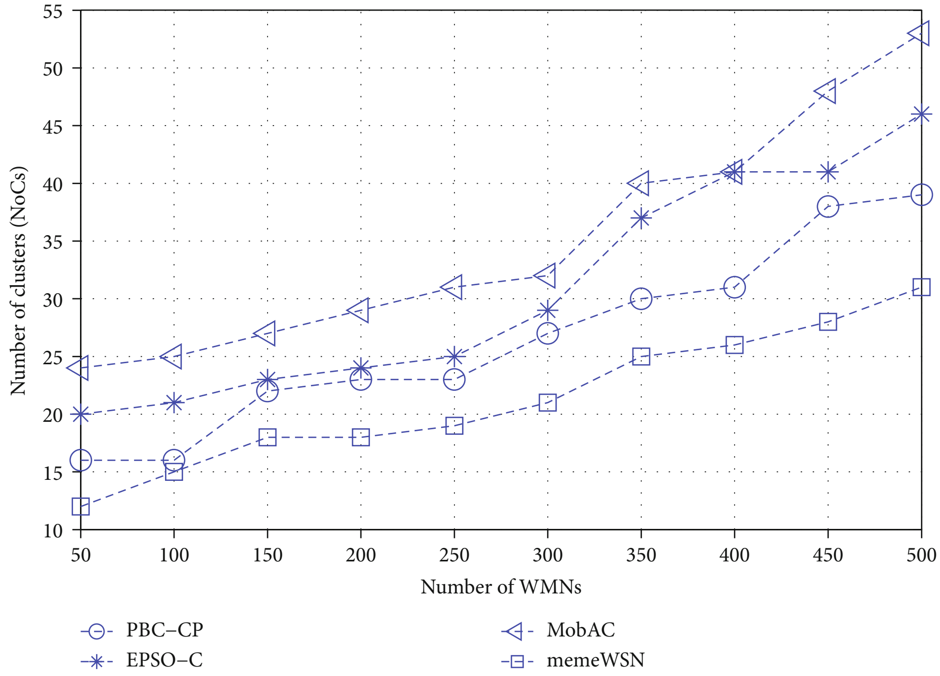


Figure 1: Average clusters vs. WSN-IoT size (WMN broadcast range 100 m).

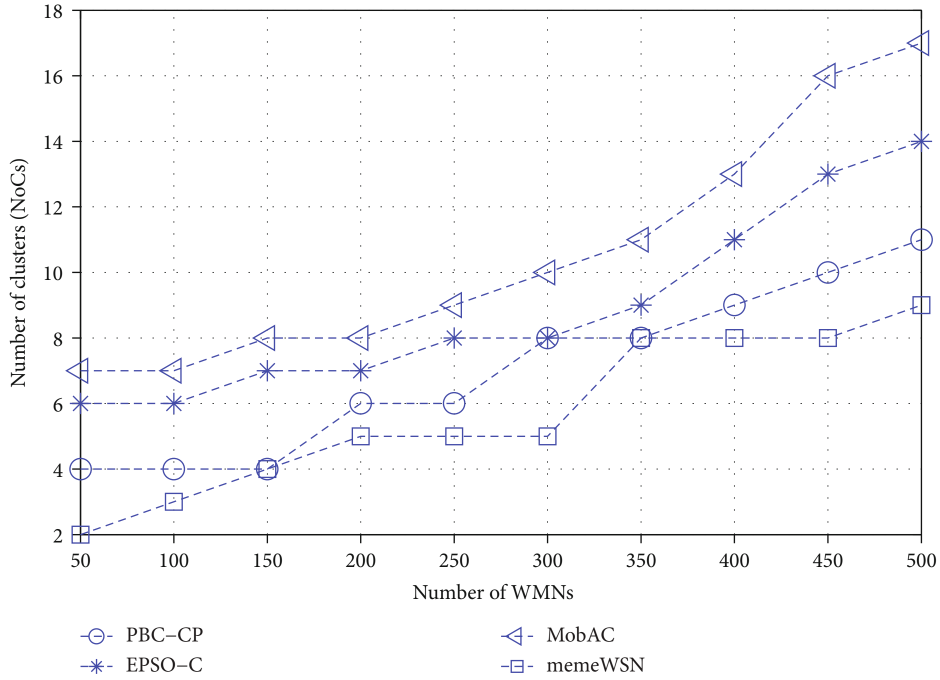


Figure 2: Average clusters vs. WSN-IoT size (WMN broadcast range 200 m).

5.3. Reaffiliation Rate (RR). To assess the efficiency of clustering schemes w.r.t reaffiliation rate for different WMN speed and wireless broadcast range, a series of simulation tests have been carried out in Figures 5 and 6. Reaffiliation may occur if a WMN leaves its present CH and join an alternative cluster or a CH move beyond the radio broadcast range of a member WMN. The WMN is unable to stay linked to its CH to any further extent. With the rise in the speed of WMNs, the reaffiliation is more often as WMNs leave their CHs more frequently. The radio broadcast range of all nodes is fixed to 200 meters. A total of 100 nodes are deployed randomly in a 1000 m × 1000 m simulation area initially.

The nodes are moving at a speed between 1 kilometer/-hour to 80 kilometers/hour. The outcomes are averaged for different experiments. The average reaffiliation rate for more than 100 dissimilar runs is presented in Figure 5. As depicted

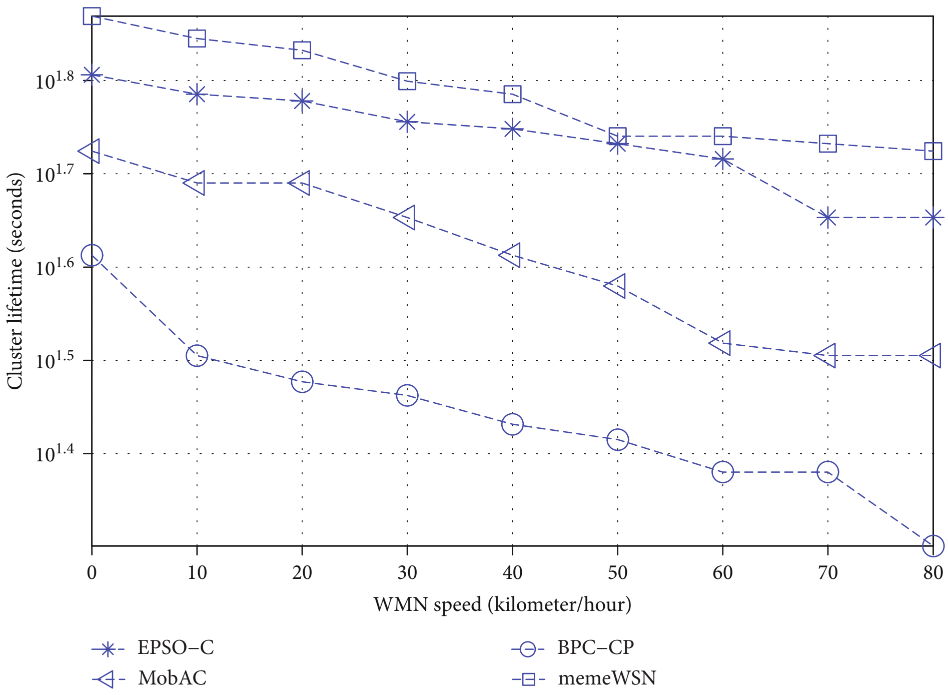


Figure 3: WMNs speed vs. cluster lifetime (WMN broadcast range 200 m).

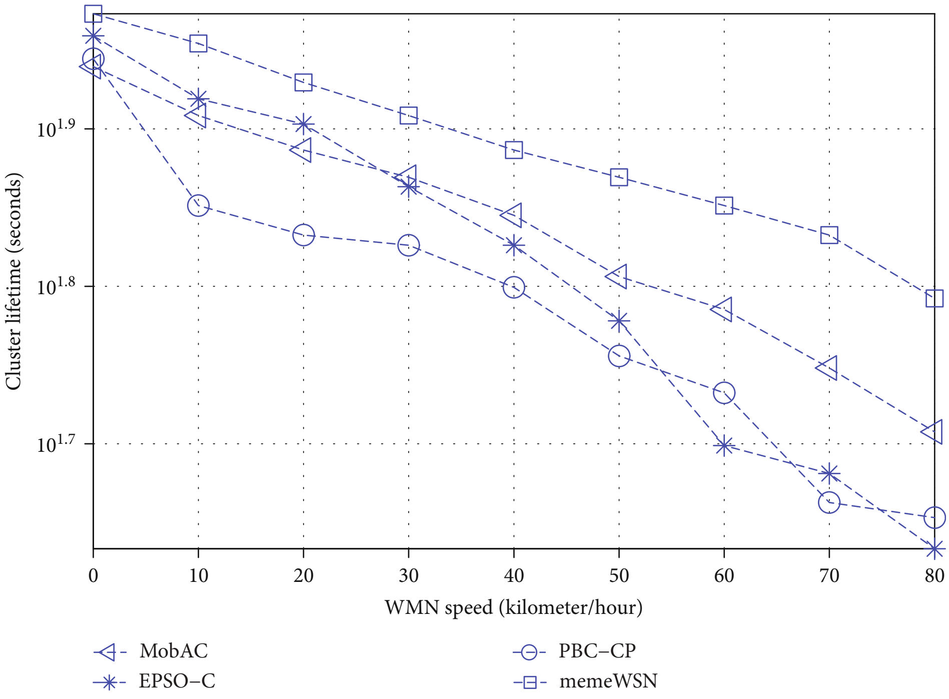


Figure 4: WMNs speed vs. cluster lifetime (WMN broadcast range 300 m).

in the graph, the rise in reaffiliation rate is noted in memeWSN, MobAC, PBC-CP, and EPSO-C when increasing the speed of WMNs. The efficiency of memeWSN is well as presented in the arcs because it has the lowermost RR as equated to other protocols under consideration. The lifetime of CHs is lengthy once the memeWSN cluster formation procedure is executed, and it indicates that the reaffiliation will be low as reclustering mechanism is initiated less repeatedly when the steady CHs are nominated. This is because the CHs are selected on the basis of WMN remaining energy, degree, and relative mobility. Therefore, the CHs lifetime will increase and the participants vacate the present cluster less frequently. The efficiency of MobAC is improved after memeWSN, since in MobAC, the WMN future mobility is taken

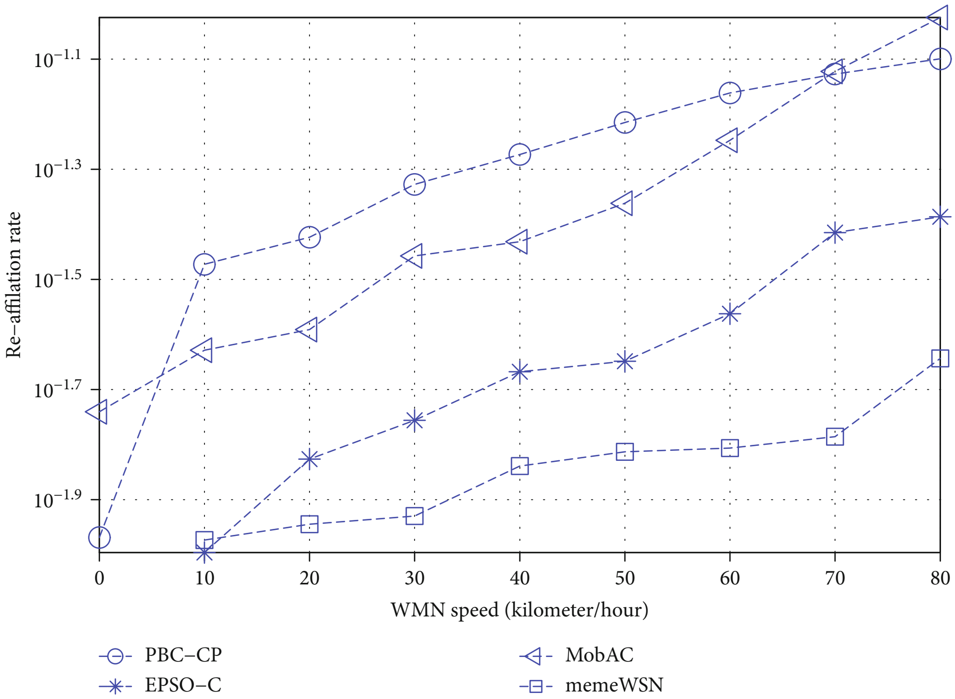


Figure 5: Reaffiliation rate vs. WMN speed.

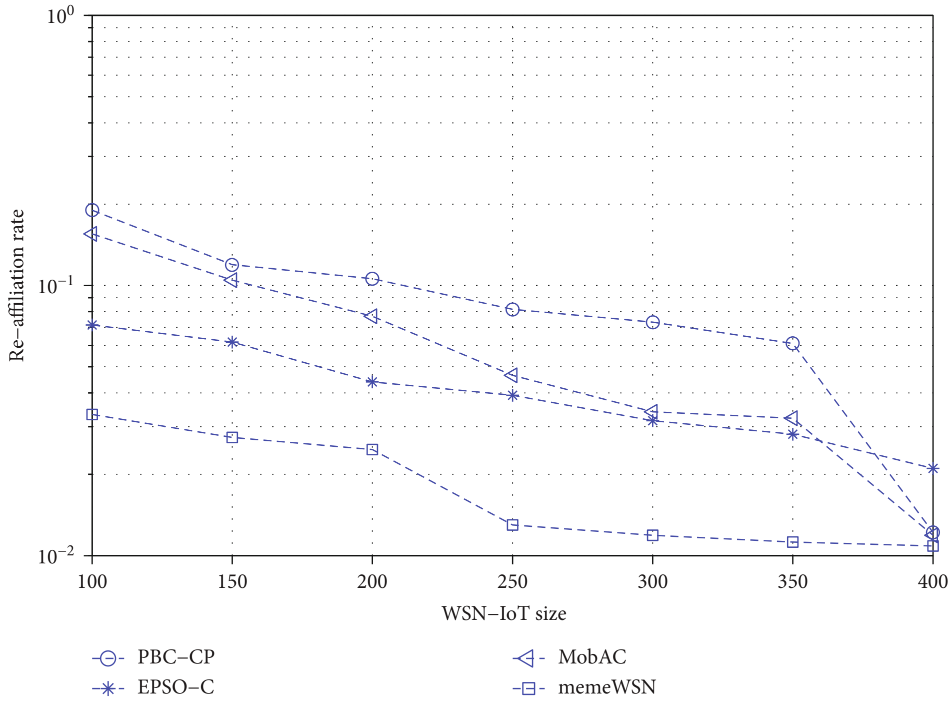


Figure 6: Reaffiliation rate vs. WMN wireless broadcast range.

into consideration throughout the CH election process. The efficiency of MobAC is better among all state of the art clustering algorithm under consideration because the neighbors of the selected CHs remain for a long time, and hence, the probability of reaffiliation decreases. The worst performance w.r.t reaffiliation is observed in PBC-CP compared to MobAC and EPSO-C. The mobility of nodes is not considered in PBC-CP during the cluster formation procedure. Hence, unstable clusters may result. The same simulation experimentations are repeated for a different size network, i.e., 50 WMNs to 500 WMNs. The results are presented in the form of a graph in Figure 6. Incremental step 50 was used

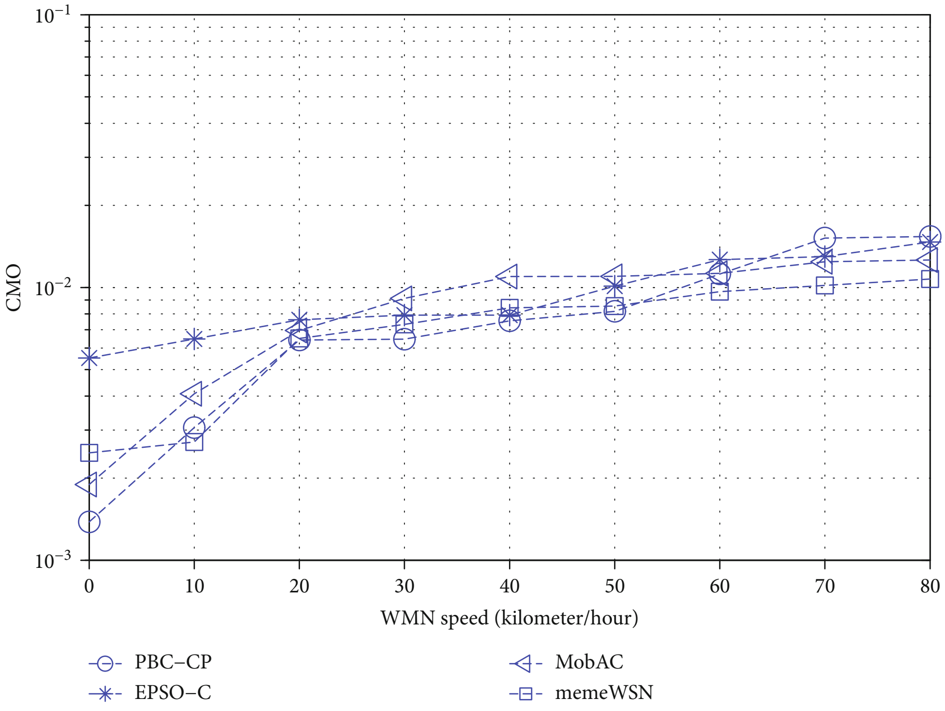


Figure 7: WSN-IoT WMN speed vs. CMO (broadcast range 200 meters).

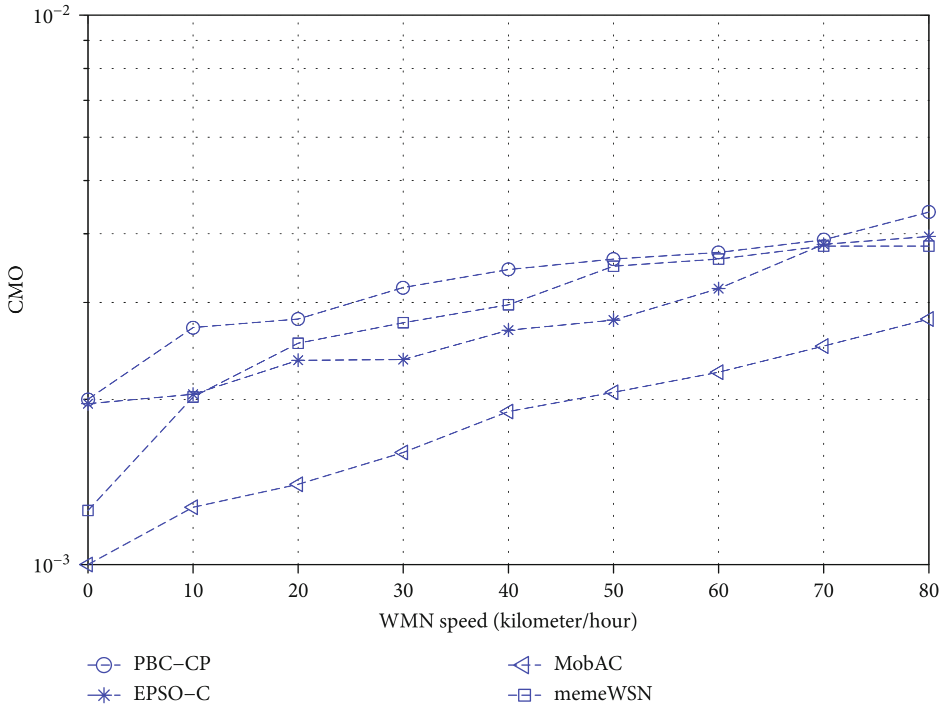


Figure 8: WSN-IoT WMN speed vs. CMO (broadcast range 300 m).

to evaluate the performance of memeWSN with a diverse number of WMNs in WSN-IoT. The graph presented in Figure 6 shows the decrease in reaffiliation rate when the WSN-IoT size becomes large. In a large-scale network, the CHs serve a large number of nodes, and the changes in topology are communicated less frequently. The simulation area was the same as in the previous experiment. The curve at the bottom of the diagram in Figure 6 shows the decrease in memeWSN reaffiliation rate when the size of the network increases in line with other cluster-based routing algorithms. In memeWSN, the CHs are selected on the basis of relative mobility, remaining energy, and node degree, and steady clusters are achieved. The reaffiliation rate will be low when the number of steady clusters increases.

5.4. Control Message Overhead (CMO). Numerous packets exchange is the essential part throughout the CH selection method in WSN-IoT. In this segment, the number of messages swapped during the course of the cluster creation and maintenance stage is noted and demonstrated. The efficiency of memeWSN is matched with MobAC, EPSO-C, and PBC-CP.

In this set of simulation tests, fifty WMNs are deployed in a random way. The simulation region $1000\,\mathrm{m} \times 1000\,\mathrm{m}$ is assumed. Random waypoint mobility was adopted in the simulation tests. The radio transmission range of WMNs is fixed to 200 meters. The WMNs are moving at a speed between 0-5 kilometers/h (walk), 5-20 kilometers/hour (running), and 20-80 kilometers/hour (car). The outcomes presented in the form of a line chart is depicted in Figure 7. The experiments are conducted for another transmission range of 300 meters, and the obtained results are presented in Figure 8. As exposed in the graph, the CMO of PBC-CP is low while the WMNs are moving with a gentle speed such as no more than 40 km/h plus becomes higher when the WMN movement is faster. GA hurts from a local maximum problem, and unsteady clusters may result once the WMN speed turns out to be high. The identical outcome can be seen in other protocols under consideration comprising memeWSN as the reclustering method is initiated frequently when the mobility is high. The CMO may decrease once we rise the radio broadcast range of WMNs. The high transmission range may drink extra energy throughout the WSN-IoT processes. The increase in WSN-IoT lifetime may not guarantee when the broadcast range of WMNs is high. More communication is required when finding optimal CHs, and the optimization techniques have great CMO values as shown in Figures 7 and 8. EPSO-C and MobAC offer consistent performance when the radio range of WMNs is 200 meters. When the broadcast range is high, EPSO-C and MobAC have the lowest CMO. Our proposed memeWSN algorithm requires more message exchanges for the duration of cluster creation. The parameters like WMN degree, its energy, and relative mobility are taken into consideration, and the accurate calculation may proliferate the number of control messages. The optimal CHs are computed on the cost of more overhead, but when the optimum CHs are identified, the reclustering method will initiate less frequently.

6. Conclusion and Future Work

In wireless sensor network-enabled Internet of Things, the resource limitation requires a handsome algorithm for cluster formation to route data with little resources in order to increase the lifetime of WSN-IoT. Optimization methods, for example, evolutionary algorithms, PSO, and linear programming, may be used to form stable and long life clusters. The stable and balanced clusters may be obtained when we consider parameters such as mobility, degree, and energy of WMNs. In this paper, the CH selection method is optimized using an evolutionary memetic algorithm. A minimization function is modeled to check the fitness of a solution. The CH-set represents a chromosome in the proposed scheme. The memetic algorithm has a built-in searching mechanism that prevents it from premature convergence. The proposed algorithm is validated through a series of simulation experiments. The simulation experiments demonstrate that the proposed algorithm outperforms the state-of-the-art clustering schemes in WSN.

In the future, a modified version of the memetic algorithm can be used to form balanced clusters in high-speed networks such as vehicular ad hoc networks and flying ad hoc networks, i.e., unmanned aerial vehicles.

Data Availability

Regarding data availability, no data were used to support this study. We discussed the results based on simulation which is already discussed in the paper in detail.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Zayed University RIF grant activity code R20129.

References

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

[2] E. M. Belding-Royer, “Hierarchical routing in ad-hoc mobile networks,” *Wireless Communications and Mobile Computing*, vol. 2, no. 5, p. 532, 2002.

[3] C. E. Perkins, *Ad-hoc Networking*, Addison-Wesley, 2001.

[4] S. Basagni and I. Chlamtac, “A generalized clustering algorithm for peer-to-peer networks,” in *Workshop on Algorithmic Aspects of Communication*, Bologna, Italy, July 1997.

[5] N. Shah, S. A. Abid, D. Qian, and W. Mehmood, “A survey of P2P content sharing in MANETs,” *Computers & Electrical Engineering*, vol. 57, pp. 55–68, 2017.

[6] A. H. Azni, R. Ahmad, K. Seman, N. H. Alwi, and Z. A. Noh, “Correlated topology control algorithm for survival network in MANETS,” in *Advanced Computer and Communication Engineering Technology. Lecture Notes in Electrical Engineering, vol 362*, H. Sulaiman, M. Othman, M. Othman, Y. Rahim, and N. Pee, Eds., pp. 93–102, Springer, Cham, 2016.

[7] K. Deb, K. Sindhya, and J. Hakanen, “Multi-objective optimization,” in *Decision Sciences: Theory and Practice*, pp. 145–184, CRC Press, 2016.

[8] G. Kannan and T. Sree Renga Raja, “Energy efficient distributed cluster head scheduling scheme for two tiered wireless sensor network,” *Egyptian Informatics Journal*, vol. 16, no. 2, pp. 167–174, 2015.

[9] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, “Residual energy-based cluster-head selection in WSNs for IoT application,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5132–5139, 2019.

[10] F. Ling, “Leader based group routing in disconnected mobile ad hoc networks with group mobility,” *Wireless Personal Communications*, vol. 71, no. 3, pp. 2003–2021, 2013.

[11] U. Venkanna and R. Leela Velusamy, “Distributed cluster head election in MANET by using AHP,” *Peer-to-Peer Networking and Applications*, vol. 9, no. 1, pp. 159–170, 2016.

[12] C. Konstantopoulos, D. Gavalas, and G. Pantziou, “Clustering in mobile ad hoc networks through neighborhood stability-based mobility prediction,” *Computer Networks*, vol. 52, pp. 1797–1824, 2008.

[13] H. Cheng, S. Yang, and J. Cao, “Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc networks,” *Expert Systems with Applications*, vol. 40, pp. 1381–1392, 2013.

[14] H. Ali, W. Shahzad, and F. A. Khan, “Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization,” *Applied Soft Computing*, vol. 12, no. 7, pp. 1913–1928, 2012.

[15] V. Singh and R. B. Lohani, “Mobility aware energy efficient clustering for wireless sensor network,” in *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–6, Coimbatore, India, Febuary 2019.

[16] A. Pathak, “A proficient bee colony-clustering protocol to prolong lifetime of wireless sensor networks,” *Journal of Computer Networks and Communications*, vol. 2020, Article ID 1236187, 9 pages, 2020.

[17] C. Vimalarani, R. Subramanian, and S. N. Sivanandam, “An enhanced PSO-based clustering energy optimization algorithm for wireless sensor network,” *The Scientific World Journal*, vol. 2016, Article ID 8658760, 11 pages, 2016.

[18] M. Ahmad, A. A. Ikram, R. Lela, I. Wahid, and R. Ulla, “Honey bee algorithm–based efficient cluster formation and optimization scheme in mobile ad hoc networks,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 6, Article ID 155014771771681, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8853662, 16 pages
https://doi.org/10.1155/2020/8853662

Research Article

Iterative Sensor Clustering and Mobile Sink Trajectory Optimization for Wireless Sensor Network with Nonuniform Density

Joohan Park, Soohyeong Kim, Jiseung Youn, Seyoung Ahn, and Sunghyun Cho

Department of Computer Science and Engineering, Hanyang University, Republic of Korea

Correspondence should be addressed to Sunghyun Cho; chopro@hanyang.ac.kr

Received 3 July 2020; Revised 20 July 2020; Accepted 8 October 2020; Published 20 October 2020

Academic Editor: Ki-Il Kim

Sensor clustering and trajectory optimization are a hot topic for last decade to improve energy efficiency of wireless sensor network (WSN). Most of existing studies assume that the sensor is uniformly deployed or all regions in the WSN coverage have the same level of interest. However, even in the same WSN, areas with high probability of disaster will have to form a "hotspot" with more sensors densely placed in order to be sensitive to environmental changes. The energy hole can be serious if sensor clustering and trajectory optimization are formulated without considering the hotspot. Therefore, we need to devise a sensor clustering and trajectory optimization algorithm considering the hotspots of WSN. In this paper, we propose an iterative algorithm to minimize the amount of energy consumed by components of WSN named ISCTO. The ISCTO algorithm consists of two phases. The first phase is a sensor clustering phase used to find the suitable number of clusters and cluster headers by considering the density of sensor and residual battery of sensors. The second phase is a trajectory optimization phase used to formulate suitable trajectory of multiple mobile sinks to minimize the amount of energy consumed by mobile sinks. The ISCTO algorithm performs two phases repeatedly until the amount of energy consumed by the WSN is not reduced. In addition, we show the performance of the proposed algorithm in terms of the total amount of energy consumed by sensors and mobile sinks.

1. Introduction

With the advent of the Internet of Things (IoT), the wireless sensor network (WSN) has attracted much attention as a key enabler of IoT technology. The WSN is used in a wide range of fields such as home IoT [1], smart agriculture [2], and smart city [3]. The WSN is a network consisting of a large number of battery-powered sensors deployed in a large area. To maximize the performance of the WSN, two challenges must be considered. One is the lifetime of the WSN. Since the sensor is powered by a battery, the lifetime of the sensor cannot last indefinitely. Therefore, in order to increase the lifetime of the sensor, numerous approaches such as energy-efficient routing path construction [4], duty-cycling management [5], and radio frequency energy harvesting [6] have been conducted [7]. The other problem is a coverage hole. The WSN is constructed in a very large Area of Interest (AoI) such as battlefields, radioactive power plants, and farms [8]. It is very important to place the sensor so that the coverage of the WSN is not empty in the AoI. The WSN needs to monitor the changes in the environment such as temperature, humidity, air pressure, and radiation occurring in the AoI. If there is a coverage hole in the AoI, there is a possibility that the change in the environment may not be correctly recognized. WSNs with important targets that should not be missed, such as severe radioactive leaks, can be more damaging if environmental changes are not properly monitored. Therefore, many researches have been conducted on sensor deployment to eliminate the coverage hole within a very large AoI [9].

However, it may not be appropriate to consider the entire coverage of a very wide WSN with the same level of interest. Where there is a relatively high probability of a disaster, such as fires, or radioactive spills, the area must be sensitive to environmental changes compared to those that are not. This means that sensors in areas with a high probability of disaster have to sense more frequently than other sensors. Sensors that sense frequently consume energy faster than other sensors. This causes an imbalance in energy consumption between sensors. More sensors need to be deployed to increase density to balance energy consumption between sensors across all coverage and thoroughly investigate environmental changes in the AoI [10].

However, an energy hole issue may occur where the sensor density is high [11]. Sensors transmit data to the sink through relaying between sensors to process the generated data [12]. Due to the geographical characteristics, a node close to the sink frequently communicates with a lot of data. This causes the sensor closer to the sink to discharge the battery faster than the far sensor, and this problem is called an energy hole issue. If the sensor density is high, the sensors located geographically close to each other generate a lot of data. A lot of data is relayed using a common intermediate sensor to reach the sink. As a result, the intermediate sensor which was in charge of relaying high-density sensors had to discharge the battery quickly.

The energy hole can be relaxed by using clustering and mobile sink [13]. It is possible to mitigate energy holes by sending data directly to the near cluster header by creating a large number of fine clusters. In addition, the energy hole can be mitigated by moving the sink near the cluster header to directly receive the collected data.

However, most sensor cluster and mobile sink trajectory planning studies do not take into account the possibility of "hotspot" in which sensors are densely located in areas of high interest. As mentioned above, a large number of fine clusters must be created in order to mitigate the energy hole in the hotspot. If the cluster header is selected without considering the density of each sensor, there is a probability that a sufficient number of fine clusters are not generated in the hotspot. If only a small number of clusters are formed in the hotspot, the number of member nodes responsible for the cluster header increases. This increases the energy consumption generated when the cluster header receives data sent by the member node and increases the probability that an energy hole will occur. Likewise, forming a trajectory of mobile sinks without considering hotspot can cause an imbalance in energy consumption between mobile sinks. The mobile sink in charge of the hotspot needs to visit a larger number of cluster headers than other mobile sinks, and energy consumption is also increased. Therefore, it is necessary to devise a method for forming clusters and trajectories of mobile sinks according to the hotspot.

Therefore, this paper proposes iterative sensor clustering and mobile sink trajectory optimization (ISCTO) method considering the hotspot of the WSN. The algorithm uses a density function using a k-nearest neighbor which can measure sensor density. We formulate the total energy consumption function of the components of the WSN. The energy consumption function is selected as an objective function of the optimization problem. To solve this problem, this algorithm separates the original optimization problem into clustering and trajectory optimization problems. In the clustering problem, the number of clusters and the cluster header and member nodes is configured to decrease the amount of energy consumed by the sensors. In the trajectory optimization, the locations of the cluster headers presented as the output in the clustering step are used to form a trajectory that minimizes the amount of energy consumed by all mobile sinks. The algorithm is repeated until energy efficiency no longer increases.

The main contributions of this paper are as follows:

(i) We propose a clustering method to increase the energy efficiency of the sensor considering the hotspot of the WSN. The clustering proceeds to select suitable cluster headers considering the density and residual battery of sensors

(ii) We propose a trajectory optimization algorithm to minimize the amount of energy consumed by all mobile sinks. In the proposed trajectory optimization method, we propose a sector boundary control algorithm to minimize the amount of energy consumed by all mobile sinks by decreasing the total trajectory length of mobile sinks

(iii) We present the performance of the proposed method through simulation. We show visually the process of sensor clustering and trajectory change. We also analyze the running time of our algorithm

The contents of this paper are as follows. In Section 2, we introduce related studies. We present the system model in Section 3. We also present the mathematical problem to be solved in Section 3. In Section 4, we introduce the proposed ISCTO algorithm to maximize the energy efficiency of the WSN. We show the performance of our proposed algorithm by simulation in Section 5. We conclude this paper in Section 6.

2. Related Works

In this section, we introduce some studies related to energy hole, sensor clustering, and mobile sink trajectory formulation in the WSN.

2.1. Energy Hole. There are numerous studies to mitigate energy hole issues in the WSN [14]. The authors of [15] propose an algorithm to balance the consumed energy among the whole network to avoid the energy hole issue. To balance a load distribution and energy consumption, the algorithm finds an optimal communication distance and uses a transmission strategy based on the amount of consumed energy. In [16], a transmission distance adjustment algorithm is proposed to minimize and balance the energy consumption of the whole sensors with a static sink. A short-trip forwarding strategy of an ant colony optimization and reference transmission distance is adopted to minimize and balance the energy consumption. The authors of [17] propose a routing path reconstructing algorithm when a dead node occurs in a network. In this paper, two protocols are introduced named on-hole children reconnection and on-hole alert. The authors of [18] analyze theoretically the lifetime of a WSN from initialization to the entire dead by estimating the energy consumption and traffic load of the sensors. In addition, the authors analyze the boundary of energy hole issues that happened. The authors of [19] propose a deployment model of a WSN consisting of heterogeneous and homogeneous sensors. The authors also give some guide to deploy the sensors by considering the energy hole issue to maximize the lifetime of the WSN on the several network topology scenarios. The authors of [20] propose two algorithms to find an optimal sojourn duration and find an optimal sojourn location of a mobile sink in a cluster. The optimal sojourn duration is determined by balancing the amount of consumed energy between cluster heads. By this algorithm, the energy hole can be alleviated. The second algorithm to find an optimal sojourn duration operates to balance the amount of consumed energy between member sensor nodes in a cluster to alleviate the coverage hole issue. In [14], a data gathering algorithm named EPEGASIS is proposed to alleviate the energy hole issue with the mobile sink. In the algorithm, a sensor calculates an optimal communication distance and then chooses a relay node from its neighboring sensors. In addition, the algorithm has a mechanism to protect the sensors from power outage by unbalanced energy consumption.

Unfortunately, the previous works may not operate efficiently in the WSN where the sensors are not deployed uniformly. When the sensors are deployed nonuniformly, the probability of the energy hole issue occurring depends on the density of the sensors. Therefore, this paper considers the hotspot location to mitigate energy hole issues to formulate energy-efficient sensor cluster and mobile sink trajectory.

2.2. Sensor Clustering. There are numerous studies to formulate an efficient sensor cluster in the WSN with a static sink [21] or mobile sink [24–22]. In [22], the wedge-based clustering and cluster merging algorithm is proposed in the static sink and the static sensor nodes of a WSN. In the algorithm, the cluster is merged when the residual energy of sensors in a wedge falls below a threshold. In addition, the authors propose a cluster header election mechanism to minimize the amount of needed energy to forward a packet to the sink. In [23], the cluster header selection mechanism is proposed in case of public safety network. To balance the load between sensors, energy and sensor density-based clustering algorithm is proposed in [21]. This paper proposes an algorithm to formulate clusters by dividing the whole network into equally sized subclusters. In addition, this paper proposed a multihop routing path-making protocol based on the clusters to maximize the lifetime of sensors.

The authors of [24] propose a clustering algorithm to maximize the lifetime of the sensors in mobile sink-based WSN. The algorithm uses the adaptive immune algorithm to make up a trajectory of the mobile sink. In addition, the algorithm determines the optimal number of cluster heads to minimize the amount of consumed energy by communication and control packet overhead. The authors of [25] propose an energy-efficient routing algorithm by considering the mobility of the sink node. The algorithm is based on the clustering. Therefore, the algorithm contains sectorizing of network and cluster head selection mechanism by considering the weight value of member sensor nodes. After cluster header selection, the member nodes construct a routing path by calculating the amount of consumed energy. To reduce the burden of sensors caused by location changes of the mobile sink, the author of [26] proposes a routing algorithm with a virtual multiring-shaped infrastructure. By this infrastructure, the burden can be reduced caused by advertising the new location of the mobile sinks.

However, most sensor clustering studies consider the situation where the sensors are relatively uniformly deployed. If sensor clustering is formulated without considering the hotspot where a large number of sensors are concentrated in the WSN, energy holes may occur. Therefore, this paper proposes an algorithm to find the appropriate cluster number and cluster header considering the density of sensors.

2.3. Mobile Sink Trajectory. There are numerous studies to formulate an energy-efficient mobile sink trajectory in the WSN [27–34]. In [27], an obstacle in the communication link between a sensor and a mobile sink is considered. This paper proposes a trajectory formulation mechanism by considering the obstacles in a cluster-based WSN. In [28], rechargeable sensor network is considered. The data gathering problem in this paper is formulated as a network utility maximization problem to maximize the amount of collected data by the mobile sink with a constraint about fairness. The authors of [29] propose a duty-cycling management and unmanned aerial vehicle (UAV) trajectory formulation algorithm to minimize the energy consumption of the whole sensor nodes in a WSN. In this paper, the problem is formulated as a mixed-integer nonconvex programming and the suboptimal solution is obtained by the successive convex optimization method. The artificial intelligence approach is applied in [30]. This paper proposes a data gathering mechanism in the WSN with obstacles. The mechanism has two rounds; the first round is clustering round by using ant colony optimization. A genetic algorithm is applied in the second round of the mechanism to construct an effective trajectory of the mobile sink. In [31], the UAV's trajectory formulation algorithm is proposed to maximize the lifetime of the network. In this algorithm, collection points similar with rendezvous point are selected before constructing the trajectory. After the collection points are determined, the mobile sink's trajectory is constructed by visiting all the collection points. In [32], a rendezvous-based data gathering protocol is proposed. This protocol consists of two steps. The first step is choosing the appropriate rendezvous point in the WSN. The second step is an energy-efficient mobile sink path planning step. The goal of the path planning step is making sure that as many sensors as possible are near the trajectory. The authors of [33] propose two trajectory planning algorithms based on rendezvous point. In this paper, two assumptions of rendezvous point papers are relaxed. One is the speed of data generated is the same between the sensors. The other is the sensors have an infinite buffer size. The authors of [34] propose a distributed UAV's trajectory determination algorithm to maximize the lifetime of a WSN based on game theory and enhanced ant colony optimization technology. By game theory, the mobile sink finds the best rendezvous point and the optimal trajectory is calculated by enhancing ant colony optimization.

However, most of the trajectory planning studies formulate the trajectory in the case where sensors uniformly deployed. If the sensor is placed nonuniformly, the cluster headers might be closely placed in a specific region. To avoid this situation, it is necessary to dynamically change the sector visited by each mobile sink. Therefore, this paper proposes an algorithm that dynamically changes the sector based on the location of the cluster header.

3. System Model

In this section, we describe the system model considered in this paper. We also provide the mathematical problem to be optimized considering the total amount of consumed energy in the target WSN. We present basic information about the kinds of nodes consisting of the WSN. In addition, we describe two kinds of wireless propagation models used in this paper. We also define many mathematical models about cluster, trajectory, and energy. Finally, we formulate a mathematical problem. Table 1 describes the notation used in this paper and the corresponding meaning.

TABLE 1: Notation and description.

Notation	Description
N	The number of sensors
M	The number of mobile sinks
$\mathscr{N}$	The set of sensors
$\mathscr{M}$	The set of mobile sinks
R	Radius of area of the WSN
ψ_n	Location profile of sensor n
$\psi_m(t)$	Location profile of mobile sink m in time t
$d_{n,n'}$	Distance between sensor n and sensor n'
$d_{m,n}(t)$	Distance between mobile sink m and sensor n in time t
$\mathscr{N}_n$	The set of nearest sensor from sensor n
$\phi_{n,n'}$	Binary variable to express the near sensor of n
ξ_n	Density value
$\mathrm{PL}^{\mathrm{AG}}_{m,n}(t)$	Ground-to-air propagation model from sensor n to mobile sink m in time t
$\mathrm{PL}^{\mathrm{GG}}_{n,n'}(t)$	Ground-to-ground propagation model from sensor n to sensor n'
L_{FS}	Free-space propagation model
η	Environmental factor of the propagation model
f	Center frequency
P^{TX}	Transmission power of sensor
P^{min}	Minimum received signal strength to decode
K	The number of clusters
$\mathscr{K}$	The set of clusters
$\delta_{k,n}$	Binary variable to express the cluster header of cluster k
$\mathscr{N}^{\mathrm{CH}}$	The set of cluster headers
$\gamma^{k,n}$	Binary variable to express the member node of cluster k
D_k	Summation of distance between the cluster header and all member nodes of cluster k
$\lambda^{m}_{n,n'}$	Binary variable to express the path of mobile sink m from cluster header n to cluster header n_0
L_m	Total trajectory length of mobile sink m
e^{move}	Unit energy consumed per unit distance
$\mathbb{E}^{\mathrm{move}}_m$	The amount of consuming energy of the mth mobile sink
$\mathbb{E}^{\mathrm{AG}}$	The amount of consuming energy of the cluster header
$\mathbb{E}^{\mathrm{GG}}_{n,n'}$	The amount of consuming energy of sensor node n' when its cluster header is n
α	Scaling factor
$\mathbb{L}_D$	A list of density value of sensors in an ascending order
$\mathbb{L}_E$	A list of residual battery of sensors in a descending order
$\gamma_{\mathbb{L}_D}(n)$	Rank of sensor n in $\mathbb{L}_D$
$\gamma_{\mathbb{L}_E}(n)$	Rank of sensor n in $\mathbb{L}_E$

3.1. System Model. We consider a single WSN consisting of N sensor nodes (SN) and M mobile sinks as the system model. Figure 1 is a picture depicting the system model considered in this paper. We define a set, N, as the set of SNs: $\{1, 2, \cdots, n, \cdots, N\} \in \mathscr{N}$. We also define a set, M, as the set of mobile sinks: $\{1, 2, \cdots, m, \cdots, M\} \in \mathscr{M}$. $\psi_n = (x_n, y_n, h_n)$ is defined as a location profile of the nth SN. It is assumed that all sensors are placed on the ground; that is, the height value, h_n, of all sensors is set as zero. $\psi_m(t) = (x^t_m, y^t_m, h_m)$ is defined as a location profile of the mth mobile sink in time t. We assume that the altitude does not need to be changed while the mobile sinks are running. We define the location profile (0, 0, 0) as the center of the WSN. The domain of the WSN is defined as a circle with the center at (0, 0, 0) and radius R. By using the location profile of the SNs, we can calculate the distance between the nth SN and n'th SN, $d_{n,n'}$, as follows:

$$d_{n,n'} = \sqrt{(x_n - x_{n'})^2 + (y_n - y_{n'})^2}. \quad (1)$$

In a similar way, we can calculate the distance between the mth mobile sink and nth SN in time t, $d_{m,n}(t)$, as follows:

$$d_{m,n}(t) = \sqrt{(x^t_m - x_n)^2 + (y^t_m - y_n)^2 + (h_m)^2}. \quad (2)$$

As mentioned above, it is necessary to grasp the density of each sensor for the efficient clustering and the trajectory optimization since we are considering hotspot SN deployment. We define the density of a SN by using the k-nearest neighbor function. $\mathscr{N}_n$ is defined as a set of the x number

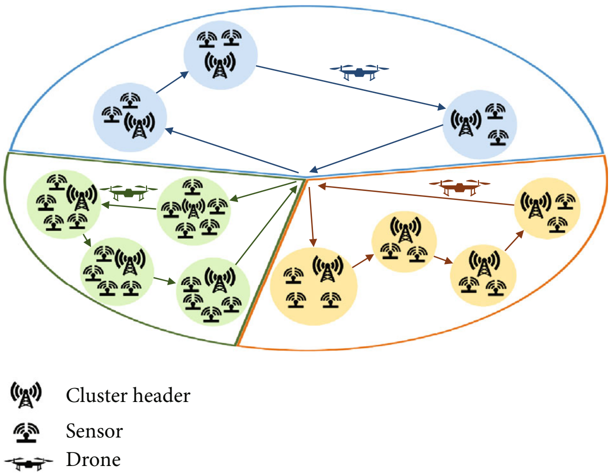


Figure 1: System model.

SNs closest to the nth SN. Using this set, we define a binary variable $\phi_{n,n'}$ as follows:

$$\phi_{n,n'} \begin{cases} 1, & \text{if } n' \in \mathcal{N}_n, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

We define the density values of the nth SN, $\mathcal{X}_{\mathrm{n}}$, as follows using $\phi_{n,n'}$ and Euclidean distance:

$$\mathcal{X}_n = \sum_{n' \in \mathcal{N}\{n\}} \phi_{n,n'} \cdot d_{n,n'}. \tag{4}$$

3.2. Wireless Channel Model. We assume that there are two types of communication in the considered system model. One is ground-to-ground propagation, where the member node SN sends data to its cluster header. The other is ground-to-air propagation, where the cluster header sends the collected data to the mobile sink. We use the ground-to-air propagation model already presented in [35, 36]. If the nth SN transmits the collected data to the mth mobile sink in time t, the considered ground-to-air propagation model, $\mathrm{PL}^{\mathrm{AG}}_{m,n}(t)$, is as follows:

$$\mathrm{PL}^{\mathrm{AG}}_{m,n}(t) = L_{\mathrm{FS}} + 20 \log \left(d_{m,n}(t)\right) + \eta, \tag{5}$$

where η is an additional attenuation factor determined by the environment. L_{FS} is a free-space path loss model. This model with center frequency f is defined as follows:

$$L_{\mathrm{FS}} = 20 \log (f) + 20 \log \left(\frac{4\pi}{c}\right), \tag{6}$$

where c is also environmental variable.

We also use the ground-to-ground propagation model already presented in [36, 37]. When the$n$th SN transmits a data to then'th SN, the propagation model,$\mathrm{PL}^{\mathrm{GG}}_{n,n'}$, is as follows:

$$\mathrm{PL}^{\mathrm{GG}}_{n,n'}(t) = -55.9 + 38 \log \left(d_{n,n'}\right) + \left(24.5 + \frac{1.5f}{925}\right) \log (f). \tag{7}$$

The SN is the only transmitter of the two kinds of communication; we only define P^{TX} as transmission power of the SN. In addition, we define $P^{\min}$ as the minimum received signal strength for a successful reception. Overall, we can define a condition for successful signal reception as follows:

$$\frac{P^{\mathrm{TX}}}{10^{PL/10}} \geq P^{\min}. \tag{8}$$

3.3. Cluster and Trajectory Model. We assume that there are the K numbers of clusters in the considered system model. Let us suppose that K is a set of the clusters $\{1, 2, \cdots, k, \cdots, K\} \in \mathcal{K}$ We define a binary variable, $\delta_{k,n}$, to express the cluster header of the kth cluster as follows:

$$\delta_{k,n} \begin{cases} 1, & \text{if } n \text{ is the cluster header of cluster } k, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

We also define a set of cluster headers of all cluster headers, $\mathcal{N}^{\mathrm{CH}}$. In a similar way, we define a binary variable, $\gamma_{k,n}$, to express the member nodes of the kth cluster as follows:

$$\gamma_{k,n} \begin{cases} 1, & \text{if } n \text{ is the member node of cluster } k, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

We can obtain the sum of the distance between the cluster header and the corresponding member SNs of the kth cluster, D_k, as follows:

$$D_k = \sum_{n \in \mathcal{N}^{\mathrm{CH}}} \sum_{n' \in \mathcal{N}^{\mathrm{CH}} \backslash \{n\}} \delta_{k,n} \cdot \gamma_{k,n'} \cdot d_{n,n'}. \tag{11}$$

In order to obtain the trajectory length for each mobile sink, we first need to define whether there is a trajectory path between cluster headers. If the mth mobile sink has a path from cluster header n to another cluster header n', $\lambda^m_{n,n'}$ is defined as one. Otherwise, it is defined as zero. Using $\lambda^m_{n,n'}$, we can get the total trajectory length, L_m, of the mth mobile sink as follows:

$$L_m = \sum_{n \in \mathcal{N}^{\mathrm{CH}}} \sum_{n' \in \mathcal{N}^{\mathrm{CH}} \backslash \{n\}} \lambda^m_{n,n'} \cdot d_{n,n'}. \tag{12}$$

3.4. Energy Model. In the proposed system model, energy consumption occurs in three cases. One is the energy consumption that the mobile sink generates as it moves along a given trajectory path. When the trajectory length of the mth mobile sink is given through (12), the amount of consuming energy of the mth mobile sink by moving, $\mathbb{E}^{\mathrm{move}}_m$, is defined as follows:

$$\mathbb{E}^{\mathrm{move}}_m = L_m \cdot e^{\mathrm{move}}. \tag{13}$$

e^{move} represents the unit energy consumed per unit distance. The other two energy consumption cases are caused by the SN. One of them appears when the cluster header sends the collected data to the mobile sink. We assume that the mobile sink comes to the exact location of the cluster header in order to receive data from the cluster header. Therefore, the transmission distance when the cluster header transmits the collected data to the mobile sink is unconditionally h_m. To increase the energy efficiency of the WSN, we assume that the cluster header sets the transmission power as the minimum strength for the mobile sink to successfully receive data. By these assumptions, the amount of energy consumed by the cluster header, $\mathbb{E}^{\mathrm{AG}}$, is defined as follows:

$$\mathbb{E}^{\mathrm{AG}} = P^{\min} \cdot 10^{PL^{\mathrm{AG}}/10}. \tag{14}$$

One remaining energy consumption case is when the member node transmits data to the cluster header. We assume that all member nodes send data directly to its cluster header without relaying to mitigate energy hole issues. The same as (14), it is assumed that the member node sets the minimum transmission power at which the cluster header can successfully receive a signal. By these assumptions, the amount of energy consumed by member node n, $\mathbb{E}^{\mathrm{GG}}_{n,n'}$, is as follows:

$$E^{\mathrm{GG}}_{n,n'} = P^{\min} \cdot 10^{\mathrm{PL}^{\mathrm{GG}}_{n,n'}/10}. \tag{15}$$

By utilizing (13)–(15), we can formulate a mathematical expression of the total amount of consumed energy in the WSN as follows:

$$\mathbb{E}_{\mathrm{tot}} = \sum_{m\in\mathcal{M}} \mathbb{E}^{\mathrm{move}}_m + K \times \mathbb{E}^{\mathrm{AG}} + \alpha \cdot \sum_{\mathrm{k}\in\mathcal{K}} \sum_{n,n'\in\mathcal{N}} \gamma_{k,n} \cdot \gamma_{k,n'} \cdot \mathbb{E}^{\mathrm{GG}}_{n,n'}. \tag{16}$$

α is a scaling factor to fairly consider the energy consumption according to the battery capacity of the SN and mobile sink. Obviously, the absolute energy consumption of mobile sinks is incomparably large for the consumption of SNs. However, mobile sinks have a large battery capacity compared to the SN, and charging is easy due to mobility. On the other hand, the SN has a very small battery and it is very difficult to charge or replace the battery. Therefore, it can be considered that the amount of battery consumed by communication has a large burden to the SN. Therefore, α is introduced to compensate for the absolute difference in energy consumption between the mobile sink and the SN.

$\mathbb{E}^{\mathrm{move}}_m$ and $\mathbb{E}^{\mathrm{AG}} + \mathbb{E}^{\mathrm{GG}}$ have a trade-off relationship. In order to reduce the energy consumption of the SN, the size of the cluster must be made finer to reduce the consumption caused by the communication between the member node and the cluster header. However, as the number of cluster headers increases, the number of points to be visited by the mobile sink increases. This causes an increase of L_m. On the other hand, in order to reduce the energy consumption of the mobile sink, minimizing the L_m by reducing the number of clusters as much as possible is advantageous. However, it increases the energy consumption of SNs. Therefore, we need to investigate this trade-off relationship and find the optimal cluster number and trajectory.

3.5. Problem Formulation. We intend to determine the number of clusters to minimize the amount of energy consumed by WSN components. In addition, we also intend to present the energy-efficient trajectory of each mobile sink. We also intend to determine the number of member nodes of all clusters to alleviate the energy hole problem of the cluster header. Therefore, the following mathematical optimization problem is presented by using (16):

$$\min_{K,\Delta,\Gamma,\Lambda} \mathbb{E}_{\mathrm{tot}} \tag{17a}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} \gamma_{k,n} = 1, \quad \forall n \in \mathcal{N}, \tag{17b}$$

$$\sum_{n\in\mathcal{N}} \delta_{k,n} = 1, \quad \forall n \in \mathcal{N}, \tag{17c}$$

$$\sum_{m\in\mathcal{M}} \lambda^m_{n,n'} \le 1, \quad \forall n \in \mathcal{N}^{\mathrm{CH}}, \tag{17d}$$

$$\sum_{n'\in\mathcal{N}^{\mathrm{CH}}\setminus\{n\}} \lambda^m_{n,n'} = 1, \quad \forall n \in \mathcal{N}^{\mathrm{CH}}, \tag{17e}$$

$$\sum_{n'\in\mathcal{N}^{\mathrm{CH}}\setminus\{n\}} \lambda^m_{n',n} = 1, \quad \forall n \in \mathcal{N}^{\mathrm{CH}}. \tag{17f}$$

Δ, Γ, and Λ are defined as the cluster header, member node profile, and trajectory path profile, respectively. Δ and Γ is a matrix of $K \times N$. Λ is a matrix of $M \times K^2$. Equation (17b) is a constraint that means that all SNs can belong to only one cluster. Equation (17c) is a constraint that every cluster can only have at most one cluster header. Constraint (17d) means that the path between the nth cluster header and the n'th cluster header can only be moved by one mobile sink. Constraints (17e) and (17f) allow the trajectory not to be scattered into multiple branches but to have only one path.

4. Proposed Method

This section provides a detailed description of the ISCTO algorithm to solve the presented problem efficiently. We reformulate the problem to solve (17a) easily. The reformed problem is divided into sensor clustering problem and trajectory optimization problem. To solve the sensor clustering problem, we define the cluster header election algorithm considering the density function defined in (4) and the amount of residual energy of the battery of the sensors. In the trajectory optimization process, we propose a trajectory path planning algorithm to minimize the amount of consumed energy by mobile sinks. Finally, we introduce an algorithm that combines the two proposed algorithms to reduce energy consumption by an iterative way.

4.1. Problem Decomposition. There is an obstacle that the domain of problem (17a) presented earlier is very wide to solve the problem directly. In order to determine the number of cluster headers, there are a total of N cases ranging from a minimum of 1 to a maximum of N. If there are k clusters, we need to examine N^k cases to find the appropriate k cluster headers among all sensors. Finally, $(k-1)^k$ paths should be considered for trajectory path planning. Consequently, the domain of (17a) is total $N^{k+1} \cdot (k-1)^k$. Therefore, we need to decompose the problem into two problems in order to solve the existing problems quickly and efficiently. One is a problem considering the energy consumption of the SN. The other is a problem of considering the energy consumption of the mobile sink. The decomposed problem about minimizing the amount of consumed energy by all SNs is as follows:

$$\min_{K,\Delta,\Gamma} K \times \mathbb{E}^{\mathrm{AG}} + \sum_{k\in\mathcal{K}} \sum_{n,n'\in\mathcal{N}} \gamma_{k,n} \cdot \gamma_{k,n'} \cdot \mathbb{E}^{\mathrm{GG}}_{n,n'} \tag{18a}$$

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} \gamma_{k,n} = 1, \quad \forall n \in \mathcal{N}, \tag{18b}$$

$$\sum_{n\in\mathcal{N}} \delta_{k,n} = 1, \quad \forall n \in \mathcal{N}. \tag{18c}$$

In a similar way, the decomposed problem of mobile sink energy consumption is as follows:

$$\min_{\Lambda} \sum_{m\in\mathcal{M}} \mathbb{E}^{\mathrm{move}}_{m} \tag{19a}$$

$$\text{s.t.} \quad \sum_{m\in\mathcal{M}} \lambda^{m}_{n,n'} \leq 1, \quad \forall n \in \mathcal{N}^{\mathrm{CH}}, \tag{19b}$$

$$\sum_{n'\in\mathcal{N}^{\mathrm{CH}}\setminus\{n\}} \lambda^{m}_{n,n'} = 1, \quad \forall n \in \mathcal{N}^{\mathrm{CH}}, \tag{19c}$$

$$\sum_{n'\in\mathcal{N}^{\mathrm{CH}}\setminus\{n\}} \lambda^{m}_{n',n} = 1, \quad \forall n \in \mathcal{N}^{\mathrm{CH}}. \tag{19d}$$

Despite going through the decomposition process, the domain is still very large to solve both problems directly. In particular, (19a) is a famous NP-hard problem that can be reduced to a traveling salesman problem [38]. Therefore, we first propose an algorithm to select the appropriate cluster header SN while gradually increasing the K value from the minimum value. Also, we present the energy-efficient trajectory path planning algorithm of mobile sinks using the cluster header SN given as the result of the aforementioned algorithm.

4.2. Sensor Clustering. As mentioned earlier, we need to devise an algorithm to solve problem (18a) easily because the domain is still very large to solve directly. We propose an algorithm to minimize the energy consumption of all SNs by forming sensor clusters and finding a suitable cluster header. Algorithm 1 shows the pseudocode of the sensor clustering algorithm.

```
Input  : $\Delta, \Gamma, K$
Output: $\Delta, \Gamma$
Make $\mathbb{L}_D, \mathbb{L}_E$
for $n \in \mathcal{N}$ do
    Obtain $r_{\mathbb{L}_D}(n), r_{\mathbb{L}_E}(n)$
end
if $K = M$ then
    Find $M$ number of $n$ with lowest $r_{\mathbb{L}_D}(n), r_{\mathbb{L}_E}(n)$
    for n with lowest $r_{\mathbb{L}_D}(n), r_{\mathbb{L}_E}(n)$ , and $k \in \mathcal{K}$ do
        $\delta_{k,n} = 1$
    end
end
if $K! = M$ then
    Find $k$ with most number of member nodes
    Find $n$ with lowest $r_{\mathbb{L}_D}(n), r_{\mathbb{L}_E}(n)$ in $k$
    $\delta_{k,n} = 1$
end
for $n \in \mathcal{N}$ do
    Update $\gamma$ with shortest $d_{n,n'}, n' \in \mathcal{N}_{CH}$
end
return $\Delta, \Gamma$
```

Algorithm 1. Density-battery coupled sensor cluster algorithm.

Before we explain the procedures of the algorithm, we need to figure out which SN is suitable to be a cluster header. We consider the density value defined by (4) and the amount of residual energy of all SNs in selecting a suitable cluster header. It can be said that the SN is suitable for becoming a cluster header in the following situations. In terms of density, it is better that a SN which can minimize D_k becomes the cluster header to reduce the amount of energy that member nodes consume by transmission. In terms of residual energy, a SN with larger residual energy is good for a cluster header so that data transmitted by a large number of member nodes can be successfully received without discharge. We define $\mathbb{L}_D$ as a list in which all SNs in N are sorted in an ascending order by density value, $\mathcal{X}_n$. In a similar way, we define $\mathbb{L}_E$ as a list of all SNs in N sorted in a descending order by residual battery. We define $\gamma_{\mathbb{L}_D}(n)$ and $\gamma_{\mathbb{L}_E}(n)$ as a rank using the order of a specific SN n in the two lists of a specific SN, L_D and L_E. We get the sum of the two ranks, $\gamma_{\mathbb{L}_D}(n) + \gamma_{\mathbb{L}_E}(n)$ of all SNs. A SN n with the smallest sum of ranks is a suitable SN for cluster header. The SN is selected as a cluster header.

The proposed algorithm works in the following order. If there is no cluster header already placed, that is, $K = 0$, the number of cluster headers of the same number as the mobile sink, M, is selected in the order of the lowest sum of the two ranks defined above. The reason for selecting the first M cluster header is to form the trajectory sector of each mobile sink, which is explained in detail in Section 4.3. If there are previously formed cluster headers, that is, $k \geq M$, an additional cluster header is selected. Unlike the initial M cluster header selection, the newly selected cluster header is selected of a SN with the lowest sum of ranks in the cluster with the most member nodes among the existing clusters. The reason of this procedure is to mitigate the energy hole issue. The header of a cluster with the most member nodes increases the battery consumption due to data reception compared to other headers. Therefore, the more the member nodes, the faster the header battery is discharged, so it is necessary to disassemble the cluster. If cluster headers are selected through the above-mentioned process, all remaining SNs form a cluster by belonging to the shortest $d_{n,n'}$ cluster header.

4.3. Trajectory Optimization. The trajectory optimization given by (19a) is a famous NP-hard problem, and we need to devise an algorithm for trajectory optimization. Fortunately, the ant colony optimization can be a good solution to find the minimum length of the trajectory. We form the mobile sink trajectory using the ant colony optimization. The remaining task is to determine the cluster headers that each mobile sink should visit. To solve this problem, a sectorization-based technique is proposed to determine the coverage of each mobile sink in the entire network. It is assumed that each mobile sink visits all cluster headers in its own sector. Also, we propose an algorithm that adjusts the sector to find the optimal trajectory when a new cluster header is added by the algorithm mentioned in Section 4.2.

Algorithm 2 shows the pseudocode of the trajectory optimization algorithm. If the first M cluster header is selected by the sensor clustering algorithm, one mobile sink is matched for each cluster header. At this time, the first sector of each mobile sink should be formed based on the location of the cluster headers. We make line segments so that both ends are the center of the WSN and the location of each cluster header n. In addition, we make line segments that bisect the angle formed by neighboring segments by the length of R starting from the center of the WSN. The arc formed by the segments of the length of R becomes the initial sector in charge of each mobile sink. s_m is the sector of mobile sink m, and θ_m is the central angle of arc-shape sector s_m.

When a new cluster header appears by the clustering algorithm, it is inevitably included in an area of one sector among previously set sectors. In this case, we define two cases named C1 and C2. First, C1 is a case in which the mobile sink in charge of the sector including the added cluster header visits the newly added cluster header. In this case, the trajectory length of mobile sinks in charge of other sectors does not change. L_m^{C1} is defined as the trajectory length of mobile sink m in case C1. L^{C1} is defined as the sum of the trajectory lengths of all mobile sinks calculated in case C1:

$$L^{C1} = \sum_{m \in \mathscr{M}} L_m^{C1}. \tag{20}$$

```
Input   : $\Delta, \Lambda, K$
Output: $\Lambda$
if $K = M$ then
    for $n \in \mathscr{N}_{CH}$ and $m \in \mathscr{M}$ do
        $\lambda_{0,n}^m = 1$ and $\lambda_{n,0}^m = 1$
        Forming $s_m$
        Calculating $\theta_m$
        $L_m = 2 \times \sqrt{x_n^2 + y_n^2}$
        end
end
if $n \in \mathscr{N}_{CH}$ has no $\lambda^m = 1 \forall m \in \mathscr{M}$ then
    Find $s_m$ included $\psi_n$
    Update $\Lambda^{C1}$
    for $m \in \mathscr{M}$ do
        Calculate $L_m^{C1}$
end
$L^{C1} = \sum_{m \in \mathscr{M}} L_m^{C1}$
    $s_{src} = s_m$
    $s_{dest} = m'$ which min $\{\theta_{m-1}, \theta_{m+1}\}$
    Find $n' \in s_{src}$ which has shortest Euclidean
        distance
    $s_{src}$ removes $n'$
    $s_{dest}$ includes $n'$
    Update $\Lambda^{C2}$
    for $m \in \mathscr{M}$ do
        Calculate $L_m^{C2}$
    end
    $L^{C2} = \sum_{m \in \mathscr{M}} L_m^{C2}$
    if $L^{C1} \geq L^{C2}$ then
        $\Lambda = \Lambda^{C2}$
    end
    if $L^{C1} < L^{C2}$ then
        $\Lambda = \Lambda^{C1}$
    end
end
return $\Lambda$
```

Algorithm 2. Arc-shape sector boundary adjustment.

C2 is defined as adjusting the sector boundary. At this time, the way to adjust the sector is as follows. A sector with a new cluster header is defined as a source sector, and a sector with a small central angle of arc among two neighboring sectors is defined as a destination sector. The boundary adjustment between sectors is a method of moving one cluster header from the source sector to the destination sector and adjusting the boundary of the sector accordingly. The cluster header to be moved to the destination sector is defined as the cluster header having the shortest Euclidean distance from the boundary of two neighboring sectors. After the sector boundary adjustment is completed, the trajectory length of all mobile sinks is calculated. Mobile sinks in charge of other sectors except for the source sector and destination sector do not have a change in trajectory length. L_m^{C2} is defined as the trajectory length of mobile sink m in case C2. The total trajectory length at this case is defined as

$$L^{C2} = \sum_{m \in \mathscr{M}} L_m^{C2}. \tag{21}$$

The values of L^{C1} and L^{C2} are compared to determine the final trajectory. In this case, since the case with a smaller value reduces the amount of energy consumed by the mobile sink, the case is presented as the output of the algorithm. The sector adjustment method is explained in Figure 2. Number 1 is when the first M cluster headers appeared. In case 2, the first sector is created based on the location of the first M cluster headers. In case 3, a new cluster header has appeared. 4-1 is the process of calculating L^{C1}, and 4-2 is the process of calculating L^{C2}. Finally, number 5 is the result of calculating the final trajectory.

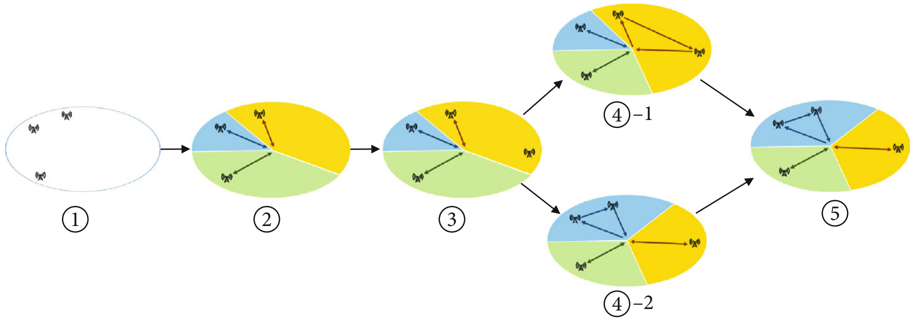


Figure 2: Procedure of modifying the territories of each sector.

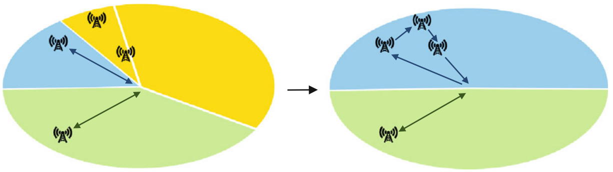


Figure 3: Special case of sector adjusting.

As a special case, there is a case where all cluster headers of the source sector are delivered to the destination sector as a result of the sector boundary adjustment. This special case is shown in Figure 3. In other words, it means the case where the mobile sink in charge of the source sector does not take off. This occurs when all the cluster headers of the source sector are included in the area that goes to the destination sector. Assuming that the trajectory length at this case is calculated as the output of the algorithm. If the next new cluster header appears, C1 and C2 are defined as follows. C1 is defined as the situation in which trajectory is designed without sector adjustment as before. C2 is defined as a case in which a cluster header is newly allocated to a mobile sink that has not taken off in the previous step. The sector boundary adjustment and trajectory are calculated according to this case instead of setting a neighboring sector as a destination sector.

4.4. ISCTO: Iterative Sensor Clustering and Trajectory Optimization. Using the algorithms proposed in Sections 4.2 and 4.3, we propose a sensor clustering and trajectory optimization algorithm to minimize the energy consumption, which is named ISCTO. As mentioned earlier, there is a trade-off relationship between the amount of energy consumption by the SN and mobile sink due to the different trends according to the value of K. Therefore, we find that trade-off point through an iterative approach.

Algorithm 3 shows the pseudocode showing the operation procedure of ISCTO. We define the initial Δ, Γ and Λ as follows:

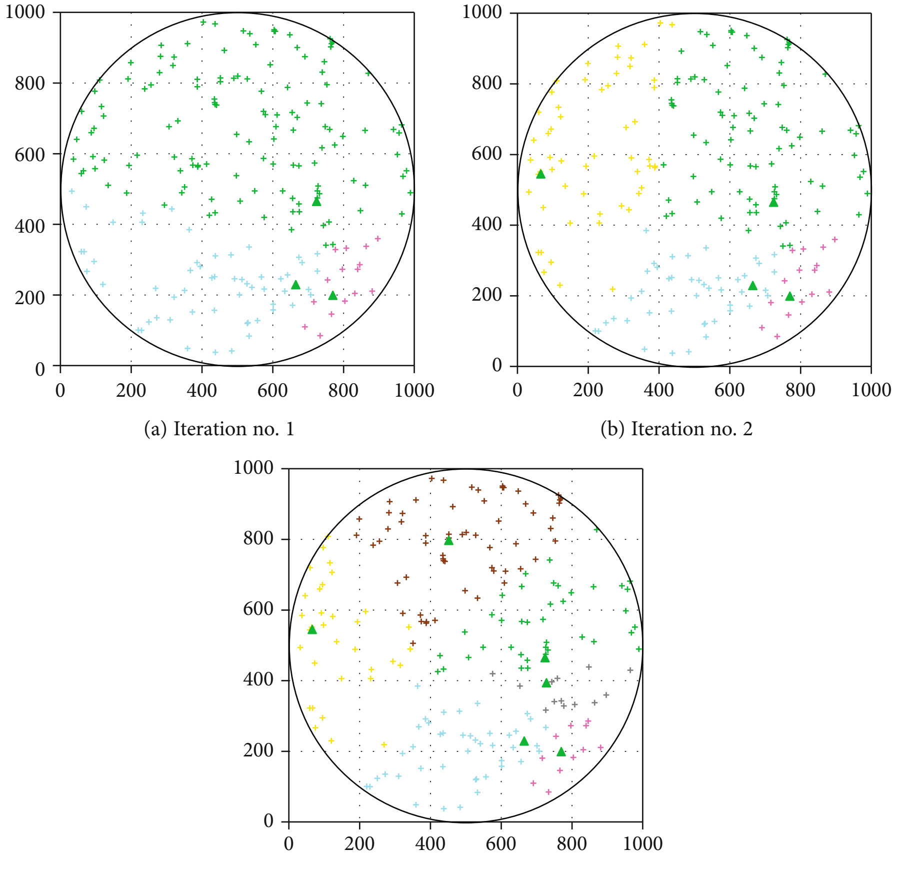
$$\begin{aligned}\Delta(0) &= \left\{0 \mid \forall\delta_{k,n}, k\in\mathcal{K}, n\in\mathcal{N}\right\},\\ \Gamma(0) &= \left\{0 \mid \forall\gamma_{k,n}, k\in\mathcal{K}, n\in\mathcal{N}\right\},\\ \Lambda(0) &= \left\{0 \mid \forall\lambda^{m}_{n,n'}, k\in\mathcal{K}, n\in\mathcal{N}^{\mathrm{CH}}\right\}.\end{aligned} \tag{22}$$

```
Input  : $\mathcal{N}, \mathcal{M}, \Psi, \mathcal{K} = \phi$
Output: $K, \Delta, \Gamma, \Lambda$
Initialize $\Delta, \Gamma, \Lambda$ as $\Delta(0), \Gamma(0), \Lambda(0)$
$K = M$
$\Delta(1), \Gamma(1) = Algorithm1(\Delta(0), \Gamma(0), K)$
$\Delta(1) = Algorithm2(\Delta(1), \Lambda(0), K)$
$\mathbb{E}^{old}_{tot}$ = calculate $\mathbb{E}_{tot}(K, \Delta(1), \Gamma(1), \Lambda(1))$
$t = 2$
while $\mathbb{E}^{new}_{tot} \le \mathbb{E}^{old}_{tot}$ do
    $K = K + 1$
    $\Delta(t), \Gamma(t) = Algorithm1(\Delta(t-1), \Gamma(t-1), K)$
    $\Lambda(t) = Algorithm2(\Delta(t), \Lambda(t-1), K)$
    $\mathbb{E}^{new}_{tot}$ = calculate $\mathbb{E}_{tot}(K, \Delta(t), \Gamma(t), \Lambda(t))$
    if $\mathbb{E}^{new}_{tot} = \mathbb{E}^{old}_{tot}$ then
        $\mathbb{E}^{old}_{tot} = \mathbb{E}^{new}_{tot}$
        $\mathbb{E}^{new}_{tot} = 0$
        $t = t + 1$
    end
end
return $K - 1, \Delta(t-1), \Gamma(t-1), \Lambda(t-1)$
```

Algorithm 3. ISCTO.

Table 2: System parameters.

Notation	Value
N	50-400
M	3, 5
R	500 (m)
h_m	50, 100 (m)
x	10
f	1 (GHz)
c	0.6
η	1
P^{TX}	10 (dBm)
P^{min}	-70 (dBm)
e^{move}	10^{-5}
α	54



(a) Iteration no. 1

(b) Iteration no. 2

(c) Iteration no. 4

Figure 4: Clustering example of random deployment.

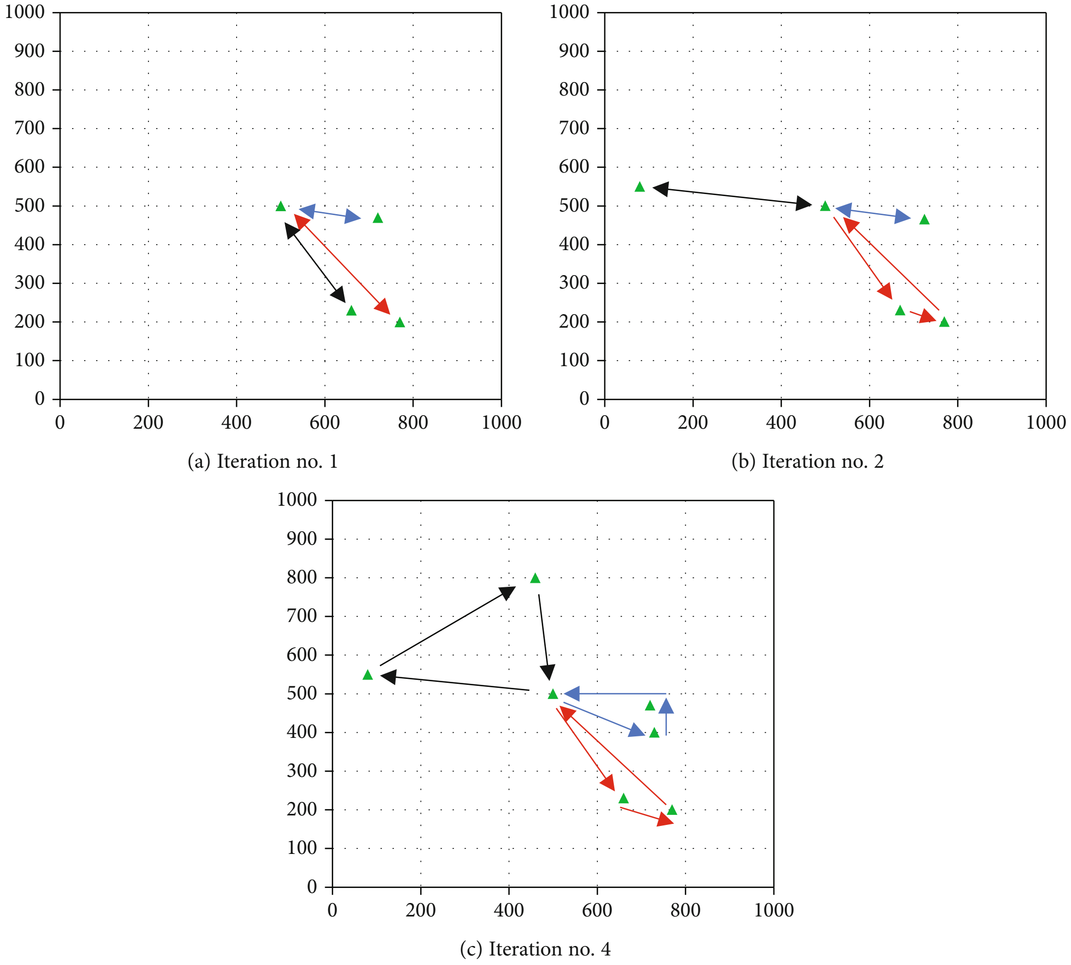
First, M initial cluster headers must be set to form M sectors. If initial M cluster headers are given, M initial sectors are formed based on the location of the cluster headers. Based on the two algorithms, the energy consumption $\mathbb{E}_{\text{tot}}$ is calculated using the given sensor cluster and trajectory. At this time, $\mathbb{E}_{\text{tot}}$ is defined as the first $\mathbb{E}_{\text{tot}}^{\text{old}}$. If $\mathbb{E}_{\text{tot}}^{\text{old}}$ is defined, we have to process the sensor cluster algorithm again, select one new cluster header, and perform trajectory optimization by considering the newly added cluster header. The next step is to calculate the energy consumption $\mathbb{E}_{\text{tot}}^{\text{new}}$ using the newly obtained sensor cluster and trajectory. Then, the values of $\mathbb{E}_{tot}^{\text{old}}$ and $\mathbb{E}_{\text{tot}}^{\text{new}}$ are compared. If $\mathbb{E}_{\text{tot}}^{\text{old}}$ is higher, it means that the new cluster header and trajectory represent better energy efficiency. Therefore, $\mathbb{E}_{\text{tot}}^{\text{old}}$ is overwritten with $\mathbb{E}_{\text{tot}}^{\text{new}}$. After that, the algorithm goes back to the sensor cluster algorithm and add a new cluster. If $\mathbb{E}_{\text{tot}}^{\text{old}}$ has a value lower than $\mathbb{E}_{\text{tot}}^{\text{new}}$, this algorithm is ended. The sensor cluster and trajectory from the last step are derived as the output of the entire algorithm.

The ISCTO algorithm forms clusters independently of the mobility of mobile sinks to improve energy efficiency in hotspots where the SN is densely located. As the defined density value in (4) is smaller, the probability of being selected as a cluster header increases, so a relatively large number of clusters are formed in the hotspot. It is possible to mitigate the energy hole issue of the cluster header by inducing that cluster headers in charge of hotspot and nonhotspot have similar member nodes. In order to improve the energy efficiency of the mobile sink, the ISCTO algorithm divides the entire network into M sectors to form the trajectory of the mobile sink. The mobility of the mobile sink is dependent on the location of the cluster headers located in its own sector.

5. Performance Evaluation

In this section, we present the performance of the proposed algorithm through simulation. The simulation environment performed is described. We consider two uneven sensor deployment scenarios. One is the fully random deployment of sensors. The other is a hotspot scenario where the majority of sensors are placed in a specific area. We present each performance in each sensor node deployment scenario. Finally, we present a graph analyzing the running time of the proposed algorithm.

5.1. Simulation Environment. We create a simulation to demonstrate the performance of the ISCTO. Table 2 shows the system parameters used in the created simulation. System parameters are determined based on [39, 40]. The values of all graphs are the average values of 10,000 times of experiments with the changing deployment of sensors. We considered two SN deployment scenarios. One is a scenario in which SNs are completely randomly deployed in the entire



(a) Iteration no. 1

(b) Iteration no. 2

(c) Iteration no. 4

Figure 5: Trajectory example of random deployment.

area of the WSN and is called a “fully randomly deployment” scenario. The other is a scenario where many SNs are placed in a specific area of the entire area of the WSN, which is called a “hotspot deployment” scenario. For the “hotspot” scenario, we experiment by placing 50% of the SN in the 1/4 area of the WSN.

5.2. Amount of Energy Consumption. Figure 4 shows the clustering transition for each iteration when $N = 200$ in a fully random deployment scenario. Figure 4(a) is the clustering result at the first iteration. The number of the first cluster is 3 because of the initial process of the sensor clustering algorithm when $M = 3$. Figure 4(b) is the result of the second iteration. In this figure, we can confirm that the new cluster head is separated from the green cluster head which had the most member nodes. Additionally, we can see that the cluster has been adjusted as a result. Figure 4(c) is the state at the end of sensor clustering; this example shows that the total of 6 clusters is formed. Figure 5 depicts the trajectory of each mobile sink formed in Figure 4. Each mobile sink trajectory is identified by the color of the arrow. Figure 5(a) is the situation where the first M cluster header is formed; each mobile sink is configured to handle one cluster header. Figure 5(b) shows the updated trajectory after the new cluster header was created. In this example, the black arrow means the source sector, and the red arrow means the destination

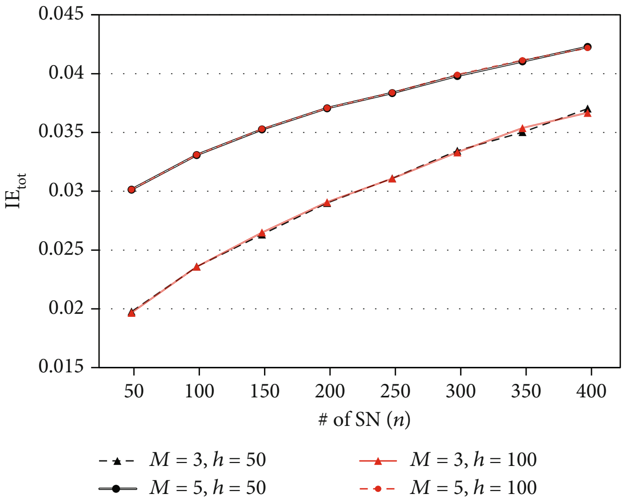
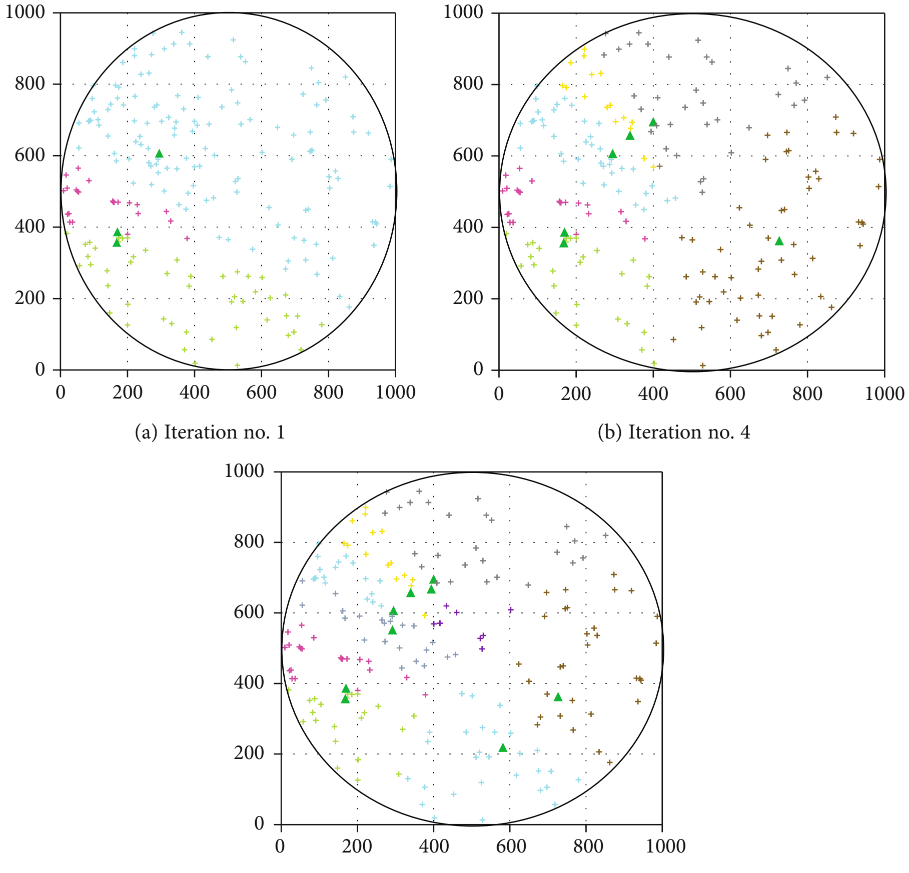


Figure 6: Performance of random deployment.



(a) Iteration no. 1

(b) Iteration no. 4

(c) Iteration no. 7

Figure 7: Clustering example of hotspot deployment.

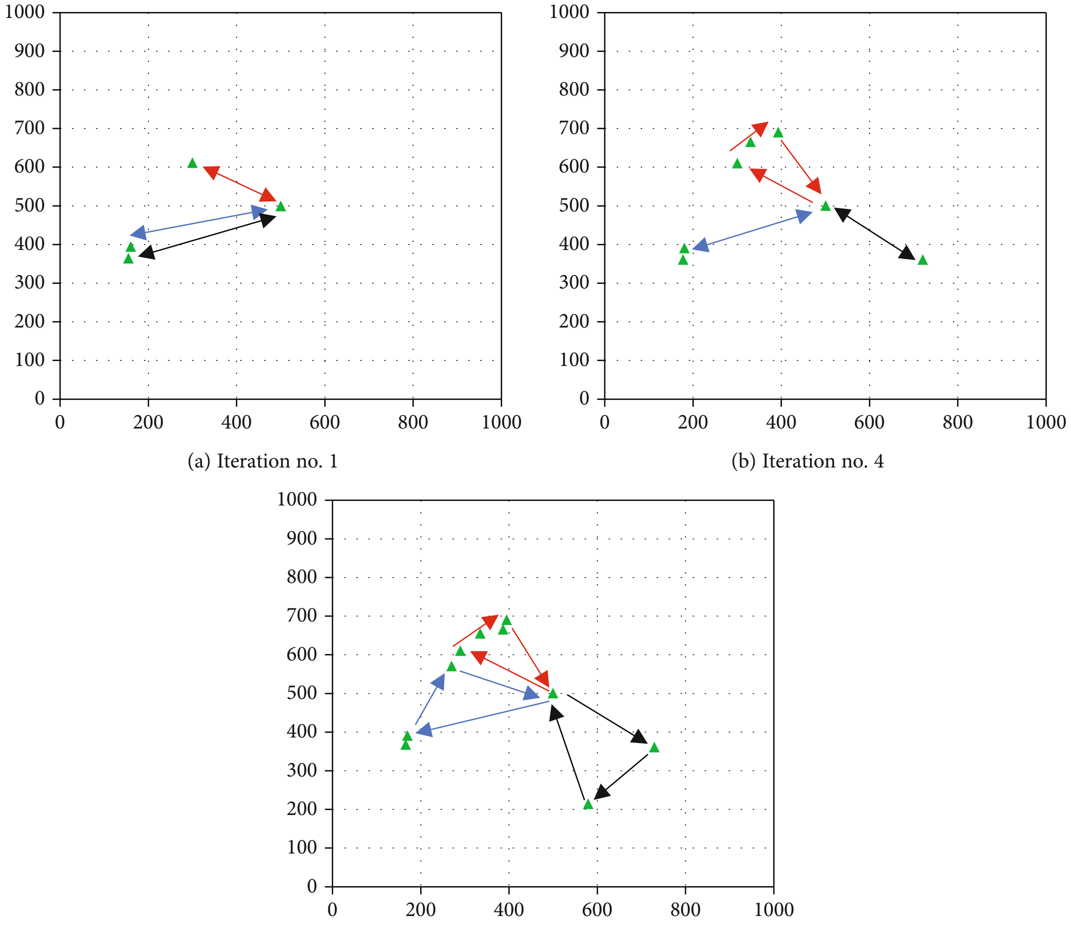
sector. In this situation, due to the fact that C2 returned a shorter trajectory, one cluster header was moved closest to the destination sector in the source sector. With repetition of the aforementioned process, we obtain the final trajectory like Figure 5(c).

Figure 6 shows the graph of the transition of $\mathbb{E}_{\text{tot}}$ according to the change of N in the random deployment scenario. The graph contains the results of experiments on four cases according to changes in the number of the mobile sink, M, and the altitude, h, of the mobile sink. It can be said that as the number of SNs increases, $\mathbb{E}_{\text{tot}}$ increases. The reason for $M = 3$ showing a lower result compared to $M = 5$ is that $\mathbb{E}^{\text{move}}$ increases with the increasing of M. It can be said that the altitude does not significantly affect the overall result. If the value of K is not extremely large, it can be said that the term $\mathbb{E}^{\text{AG}}$ does not play a large part in the result.

Figure 7. shows the cluster formation process for each iteration in a hotspot deployment scenario. In this example, it can be seen that the hotspot is formed at the upper left of the WSN. Since the density value, $\mathcal{X}$, is considered in the process of determining the cluster header, it can be seen that the sensor belonging to the hotspot was continuously determined as the cluster header in the beginning. In Figure 7(b), the first cluster header that does not belong to the hotspot is selected as a brown cluster. Figure 8 shows the trajectory formation process for each mobile sink in Figure 7. As it can be seen in Figure 7, there is a probability which the number of the cluster headers in charge of mobile sinks can be unbalanced as many cluster headers are generated in the hotspot. In this example, it can be seen that the mobile sink, which follows the red arrow, is in charge of more than two or three cluster headers than other mobile sinks. However, the sector area between mobile sinks is adjusted to have a uniform central angle as much as possible, so that the total length of trajectory for each mobile sink is set to be balanced.

Figure 9 is the graph showing $\mathbb{E}_{\text{tot}}$ according to the change of N in the hotspot deployment scenario. The experiment also compared the results of the four cases according to the changes of M and h. Most of the trend of performance value change shows the same trends shown in the random deployment scenario, though it shows a higher result value compared to the random deployment scenario. This is because hotspot deployment scenarios tend to create more clusters than random deployments. As the number of clusters increases, the trajectory length of the mobile sink also increases, and this causes an increase in the result.

5.3. Distribution of Member Node. We present the distribution of cluster member nodes as a performance indicator associated with the energy hole issue. We investigate the distribution of member nodes by cluster after experiments in two SN deployment scenarios when $M = 3$ and $M = 5$.



(a) Iteration no. 1

(b) Iteration no. 4

(c) Iteration no. 7

Figure 8: Trajectory example of hotspot deployment.

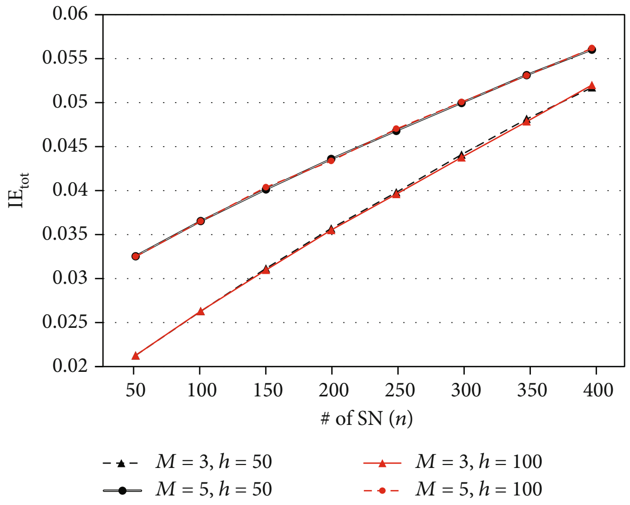


Figure 9: Performance of hotspot deployment.

Figures 10 and 11 represent the CDF of the number of member nodes for each cluster when$N = 200$and$N = 400$, respectively. As the graph is directed to the left, it means the average number of member nodes in the cluster is less. Both graphs show the same trend regardless of N. Also, in the case of the same SN deployment scenario, it can be seen that the larger the M, the fewer the member nodes. The reason for this result is based on the beginning mechanism of proposing sensor clustering algorithm. The sensor clustering algorithm creates the first M cluster at the beginning. Therefore, the larger the M, the more clusters can be formed. As more clusters are formed, the number of member nodes will naturally decrease. Therefore, larger M tends to include fewer member nodes.

In the case of the same number of M, the trend of having fewer member nodes in the hotspot deployment than in the random sensor deployment can be seen. In the hotspot deployment scenario, most of the cluster headers are generated in the hotspot due to $\gamma_{\mathbb{L}_D}$.

Although most of SNs are in the hotspot, hotspot clusters are more finely divided as many cluster headers are generated. This results in creating clusters with fewer member nodes on average in hotspot deployments. In addition, as mentioned above, the tendency for more clusters to occur in a hotspot deployment scenario is also a reason for the cluster to have fewer member nodes.

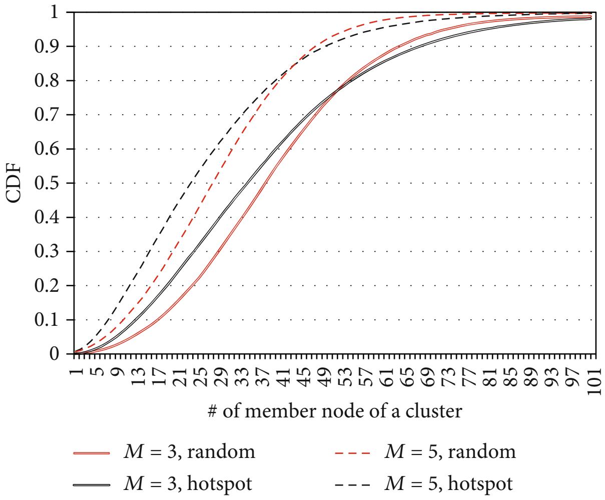


Figure 10: CDF of member node on 200 SNs.

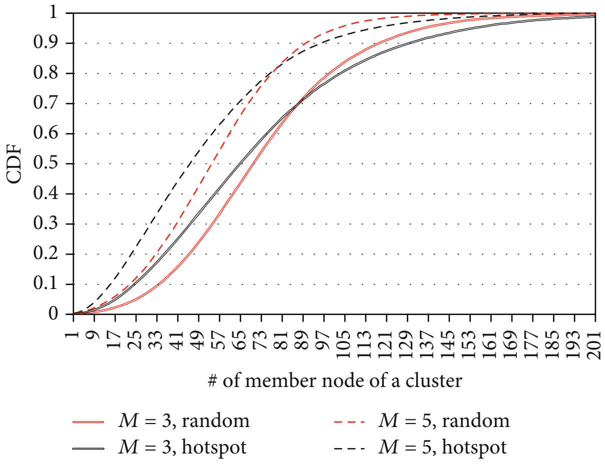


Figure 11: CDF of member node on 400 SNs.

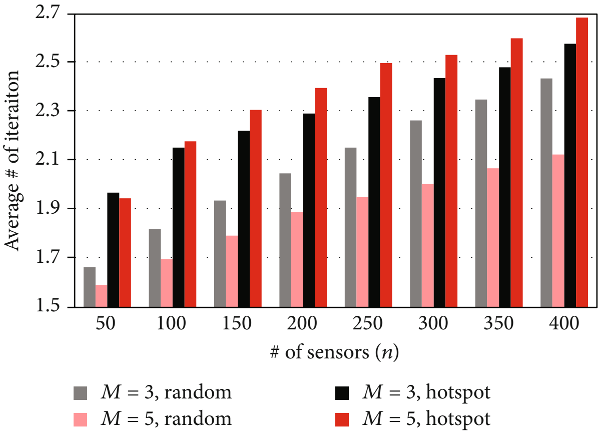


Figure 12: Average iteration count.

On the other hand, the distribution of the number of member nodes in the two deployment scenarios crosses around $1/4 \cdot N$ points. The reasons for this result are as follows. In a random deployment scenario, sensors and cluster headers are relatively evenly distributed. There is low probability that numerous member nodes are included with bias in one cluster. However, in a hotspot deployment, due to $\gamma_{\mathbb{L}_D}$, only a small number of clusters are created in a region outside the hotspot which causes the number of member nodes gathered in one cluster.

5.4. Time Complexity Analysis. To show the running time of the proposed algorithm, we experiment the average number of iteration until the algorithm ended according to changes in N. The same as the above experiment, we investigate in two SN deployment scenarios when $M = 3$ and $M = 5$. Figure 12 is a graph presenting the number of iterations. In the case of the same M, it can be seen that iteration takes more in the hotspot than in the random deployment. As shown in Figure 7, the reason is that the cluster header is continuously generated only at the hotspot. If the newly generated cluster header is not far from the previously generated cluster header, this means that the previously formed trajectory also does not need to be significantly changed. This means that even if a new cluster header is generated, the reduction amount of $\mathbb{E}^{\text{move}}$ does not occur significantly. Because of this, the proposed algorithm causes more iteration in the hotspot deployment environment.

Interestingly, depending on the change of the value of M, the trend of the iteration number changes differently. In the random deployment scenario, the larger the M, the lesser the iteration results. This is because when the M clusters are initially formed, the cluster headers are not biased to one place. Therefore, the benefit of EGG obtained by adding the cluster header is not significantly large. On the other hand, in the hotspot deployment scenario, when the initial M clusters are formed, cluster headers are intensively generated in the hotspot. The EGG generated when a new cluster header is created in the nonhotspot area is significantly large. Therefore, it runs more iterations.

In addition, we can infer that the average number of cluster headers generated by the ISCTO is (M + the number of iteration) through the graph.

6. Conclusion

We propose a sensor clustering and trajectory path planning algorithm named ISCTO for uneven SN deployed WSN with multiple mobile sinks. This algorithm iteratively performs two stages of sensor clustering and trajectory optimization to minimize the amount of energy consumed by WSN components. In the sensor clustering phase, two rankings are defined using density function and residual battery level.

Whenever a new cluster header is needed, the SN with the lowest sum of rankings is added as a new cluster header. In trajectory optimization, WSN is divided into several sectors to minimize the amount of energy that the mobile sink consumes. Whenever a new cluster header is created, consider the two cases defined and find the case that shows the shorter trajectory length. We show the performance of the ISCTO algorithm through simulation. In the future, we will improve the proposed algorithm to support the one million devices per 1 square of kilometer situation, which is a requirement of 5G mMTC.

Data Availability

We create a simulation to demonstrate the performance of our proposed algorithm.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00969, full duplex nonorthogonal multiple access (NOMA) optimization technologies using deep learning for 5G-based autonomous vehicular networks) and was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2018R1D1A1B07049043).

References

[1] T. Kim, J. Noh, and S. Cho, "SCC: storage compression consensus for blockchain in lightweight IoT network," in *2019 IEEE 684 International Conference on Consumer Electronics (ICCE)*, pp. 1–4, Las Vegas, NV, 2019.

[2] V. Bapat, P. Kale, V. Shinde, N. Deshpande, and A. Shaligram, "WSN application for crop protection to divert animal intrusions in the agricultural land," *Computers and Electronics in Agriculture*, vol. 133, pp. 88–96, 2017.

[3] V. Garcia-Font, C. Garrigues, and H. Rifà-Pous, "A comparative study of anomaly detection techniques for smart city wireless sensor networks," *sensors*, vol. 16, no. 6, p. 868, 2016.

[4] H. Mostafaei, "Energy-efficient algorithm for reliable routing of wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 7, pp. 5567–5575, 2019.

[5] Y. Liu, A. Liu, N. Zhang, X. Liu, M. Ma, and Y. Hu, "DDC: dynamic duty cycle for improving delay and energy efficiency in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 131, pp. 16–27, 2019.

[6] T. Ruan, Z. J. Chew, and M. Zhu, "Energy-aware approaches for energy harvesting powered wireless sensor nodes," *IEEE Sensors Journal*, vol. 17, no. 7, pp. 2165–2173, 2017.

[7] V. L. Quintero, C. Estevez, M. E. Orchard, and A. Perez, "Improvements of energy-efficient techniques in WSNs: a MAC-protocol approach," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1188–1208, 2019.

[8] S. Abdollahzadeh and N. J. Navimipour, "Deployment strategies in the wireless sensor network: a comprehensive review," *Computer Communications*, vol. 91-92, pp. 1–16, 2016.

[9] B. Wang, J. Zhu, L. T. Yang, and Y. Mo, "Sensor density for confident information coverage in randomly deployed sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3238–3250, 2016.

[10] D. Saha and S. Bandyopadhyay, "Finding minimum node density for energy-efficient in-hop cooperative relaying in industrial WSNs," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, vol. 74no. 12, pp. 6633–6645, Jaipur, 2018.

[11] H. Asharioun, H. Asadollahi, T.-C. Wan, and N. Gharaei, "A survey on analytical modeling and mitigation techniques for the energy hole problem in corona-based wireless sensor network," *Wireless Personal Communications*, vol. 81, no. 1, pp. 161–187, 2015.

[12] K. Cengiz and T. Dag, "Energy aware multi-hop routing protocol for WSNs," *IEEE Access*, vol. 6, pp. 2622–2633, 2018.

[13] J. Wang, J. Cao, R. S. Sherratt, and J. H. Park, "An improved ant colony optimization-based approach with mobile sink for wireless sensor networks," *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6633–6645, 2018.

[14] J. Wang, Y. Gao, X. Yin, F. Li, and H. J. Kim, "An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 9472075, 9 pages, 2018.

[15] N. Jan, N. Javaid, Q. Javaid et al., "A balanced energy-consuming and hole-alleviating algorithm for wireless sensor networks," *IEEE Access*, vol. 5, pp. 6134–6150, 2017.

[16] X. Liu, "A novel transmission range adjustment strategy for energy hole avoiding in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 67, pp. 43–52, 2016.

[17] R. E. Mohemed, A. I. Saleh, M. Abdelrazzak, and A. S. Samra, "Energy-efficient routing protocols for solving energy hole problem in wireless sensor networks," *Computer Networks*, vol. 114, pp. 51–66, 2017.

[18] J. Ren, Y. Zhang, K. Zhang, A. Liu, J. Chen, and X. S. Shen, "Lifetime and energy hole evolution analysis in data-gathering wireless sensor networks," *IEEE transactions on industrial informatics*, vol. 12, no. 2, pp. 788–800, 2016.

[19] H. S. Ramos, A. Boukerche, A. L. C. Oliveira, A. C. Frery, E. M. R. Oliveira, and A. A. F. Loureiro, "On the deployment of large-scale wireless sensor networks considering the energy hole problem," *Computer Networks*, vol. 110, pp. 154–167, 2016.

[20] N. Gharaei, K. Abu Bakar, S. Z. M. Hashim, and A. H. Pourasl, "Inter- and intra-cluster movement of mobile sink algorithms for cluster-based networks to enhance the network lifetime," *Ad Hoc Networks*, vol. 85, pp. 60–70, 2019.

[21] K. A. Darabkh, S. M. Odetallah, Z. al-qudah, A.'. F. Khalifeh, and M. M. Shurman, "Energy-aware and density-based clustering and relaying protocol (EA-DB-CRP) for gathering data in wireless sensor networks," *Applied Soft Computing*, vol. 80, pp. 154–166, 2019.

[22] N. Sharmin, A. Karmaker, W. L. Lambert, M. S. Alam, and M. S. T. S. A. Shawkat, "Minimizing the energy hole problem in wireless sensor networks: a wedge merging approach," *Sensors*, vol. 20, no. 1, p. 277, 2020.

[23] A. R. Ansari and S. Cho, "CHESS-PC: cluster-head selection scheme with power control for public safety networks," *IEEE Access*, vol. 6, pp. 51640–51646, 2018.

[24] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, "Mobile sink-based adaptive immune energy-efficient clustering protocol for improving the lifetime and stability period of wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4576–4586, 2015.

[25] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H.-J. Kim, "Energy efficient routing algorithm with mobile sink support for wireless sensor networks," *Sensors*, vol. 19, no. 7, p. 1494, 2019.

[26] R. Yarinezhad, "Reducing delay and prolonging the lifetime of wireless sensor network using efficient routing protocol based on mobile sink and virtual infrastructure," *Ad Hoc Networks*, vol. 84, pp. 42–55, 2019.

[27] G. Xie and F. Pan, "Cluster-based routing for the mobile sink in wireless sensor networks with obstacles," *IEEE Access*, vol. 4, pp. 2019–2028, 2016.

[28] Y. Zhang, S. He, and J. Chen, "Near optimal data gathering in rechargeable sensor networks with a mobile sink," *IEEE Transactions on Mobile Computing*, vol. 16, no. 6, pp. 1718–1729, 2017.

[29] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, 2018.

[30] S. Najjar-Ghabel, L. Farzinvash, and S. N. Razavi, "Mobile sink-based data gathering in wireless sensor networks with obstacles using artificial intelligence algorithms," *Ad Hoc Networks*, vol. 106, p. 102243, 2020.

[31] W. Wen, S. Zhao, C. Shang, and C. Y. Chang, "EAPC: energy-aware path construction for data collection using mobile sink in wireless sensor networks," *IEEE Sensors Journal*, vol. 18, no. 2, pp. 890–901, 2018.

[32] C. Konstantopoulos, N. Vathis, G. Pantziou, and D. Gavalas, "Employing mobile elements for delay-constrained data gathering in WSNs," *Computer Networks*, vol. 135, pp. 108–131, 2018.

[33] Y. C. Wang and K. C. Chen, "Efficient path planning for a mobile sink to reliably gather data from sensors with diverse sensing rates and limited buffers," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1527–1540, 2019.

[34] P. V. P. Raj, A. M. Khedr, and Z. A. Aghbari, "Data gathering via mobile sink in WSNs using game theory and enhanced ant colony optimization," *Wireless Networks*, vol. 26, no. 4, pp. 2983–2998, 2020.

[35] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," *IEEE Global Communications Conference*, pp. 2898–2904, 2014.

[36] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: design and optimization for multi-UAV networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1346–1359, 2019.

[37] 3GPP TS 25 996, *Spatial channel model for multiple input multiple output (MIMO) simulations*, 3GPP, 2018.

[38] M. J. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness, 1st ed*, Siam Review, New York, NY, USA: Freeman, 1979.

[39] 3GPP TR 22 289, *Enhancement for unmanned aerial vehicles; stage 1 (Rel. 17)*, 3GPP, 2019.

[40] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8849433, 17 pages
https://doi.org/10.1155/2020/8849433

Research Article

A Multidimensional Internet of Things Testbed System: Development and Evaluation

Ibrahim S. Alsukayti

Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia

Correspondence should be addressed to Ibrahim S. Alsukayti; skiety@qu.edu.sa

Received 4 August 2020; Revised 5 September 2020; Accepted 25 September 2020; Published 14 October 2020

Academic Editor: Shuhui Yang

The technological breakthrough of the Internet of Things (IoT) drives the emergence of a wide scope of smart IoT solutions in different domains. Advancing the different technological aspects of these solutions requires effective IoT implementations and experimentations. This is widely addressed following low-cost and scalable methods such as analytical modeling and simulation. However, such methods are limited in capturing physical characteristics and network conditions in a realistic manner. Therefore, this paper presents an innovative IoT testbed system which facilitates practical experimentation of different IoT solutions in an effective environment. The testbed design was developed towards a general-purpose multidimensional support of different IoT properties including sensing, communication, gateway, energy management, data processing, and security. The implementation of the testbed was realized based on integrating a set of robust hardware components and developing a number of software modules. To illustrate its effectiveness, the testbed was utilized to experiment with energy efficiency of selected IoT communication technologies. This resulted in lower energy consumption using the Bluetooth Low Energy (BLE) technology compared to the Zigbee and 6LoWPAN technologies. A further evaluation study of the system was carried out following the Technology Acceptance Model (TAM). As the study results indicated, the system provides a simple yet efficient platform for conducting practical IoT experiments. It also had positive impact on users' behavior and attitude toward IoT experimentation.

1. Introduction

Internet of Things (IoT) is a technological move towards effective convergence of the physical and digital worlds. It starts to significantly influence our daily lives as it finds its way into different governmental, industrial, and commercial domains. Nowadays, there is a growing interest in IoT development considering different applications such as smart home, e-healthcare, and intelligent transportation systems. Real deployment of IoT systems becomes widespread in different regions of the globe. The number of IoT-enabled objects is increasing to reach high figures in the near future.

Along with the rapid development of the IoT technology, wide range of IoT solutions has emerged. On the other hand, advancing the different technological aspects of IoT becomes a trend research direction. Different research questions have been raised up in a broad array of IoT applications. These are related to different aspects such as which and how IoT communication technologies should be set up, what and how IoT data should be collected and processed, and where IoT nodes should be positioned. However, such considerations critically require effective testing and experimentation.

There are a number of methods that can be adopted for this process. The most common ones are analytical modeling, simulation, and testbed experimentation. Analytical modeling relies on mathematical and statistical methods, whereas simulation is widely adopted as a low-cost and flexible experimentation approach. However, these methods have well-known limitations as they rely on modeling physical characteristics and reflecting physical phenomena using approximations and simplifications. This makes it complex to capture all natural hardware characteristics and realistic network conditions. It is challenging for simulation software to efficiently well-handle imperfections in radio communication, hardware interaction, sensing, and network traffic. Therefore, the need for IoT implementation and evaluation in a natural and real environment is evident. Having experimental IoT setups developed using physical IoT components with real IoT data traffic is a more effective experimentation approach. Although it comes with a cost in terms of money and time, building IoT testbed systems would allow increasing the realism of testing environment and improving the credibility of the evaluation process.

On the other hand, building a motivating academic and research environment is important for advancing the process to a further limit. This is one of the main strategic objectives towards which the College of Computer in Qassim University strives. Accordingly, a number of academic resources have been made available and a variety of research activities has been supported. However, there are still no customized facilities available to support IoT research and education in the college. Researchers usually have to build their own experimental IoT setups for carrying out certain implementation and testing procedures. Educators have limited resources to acquire hands-on experience and interactive learning when it comes to practical IoT concepts. Given the growing interest in IoT technology, having a general-purpose and open-access IoT testbed would provide collaborative research environment and interactive learning platform. It can be, for example, utilized to investigate IoT communication technologies, study IoT network performance, test IoT security techniques, prototype IoT applications, and analyze IoT data. Developing an IoT testbed is envisaged to open the door for different opportunities including multidisciplinary cooperation. It is a cost-effective and time-saving approach to effectively promote IoT research and education.

Such a challenge is addressed in this paper which presents the design and implementation of a practical testbed system for IoT experimentation. It represents a step forward towards moving IoT experimentation to a more realistic and practical level. The main contribution of the proposed testbed system is the novel multidimensional support providing the following:

(i) An IoT experimentation facility for effective testing of different IoT aspects in a realistic and controlled environment. It can be practically utilized to experiment with IoT networking considering a number of IoT communication protocols. It also supports IoT experimentation targeting IoT data processing, gateway operations, cloud integration, node deployment, and security

(ii) An educational platform for practical and interactive IoT teaching-learning process. It can be easily implemented for practical teaching of different IoT concepts and technologies in addition to enabling interactive and practice-based IoT education

The testbed also features a novel combination of a number of characteristics including the following:

(i) General-purpose multidimensional support of different IoT properties (sensing, communication, gateway, energy management, data processing, and security)

(ii) Scalable, portable, and simple system architecture

(iii) Cost-effective design and implementation using Commercial Off-The-Shelf (COTS) components

(iv) Effective support of a wide range of IoT communication technologies (BLE, Zigbee, 6LoWPAN, and LoRaWAN)

(v) Easy integration of different sensor devices

(vi) Cloud integration

(vii) Effective IoT data visualization

(viii) Flexible access to the testbed using a web interface and API calls

(ix) Easy implementation of a wide range of IoT applications

(x) Support of node mobility

The testbed was built based on effective design considering high levels of simplicity, scalability, usability, portability, and cost-effectiveness. The main focus was on the realization of a ready-to-use and easy-to-control multidimensional testbed for time-saving and cost-cutting IoT experimentation. The testbed implementation was realized based on integrating a set of well-known hardware components and developing a number of software modules. The efficiency of the testbed to implement and run different IoT experiments is illustrated in this paper by a simple IoT use case. The evaluation of the system was conducted using the Technology Acceptance Model (TAM). As the results indicated, the system provides simple-to-use and efficient facility for effective IoT testing. It is also evident that the system succeeded in promoting positive attitudes towards practical IoT experimentation and attractive facility for future IoT testing.

In the following section, a summary of the related work is provided. Section 3 presents a technical overview of the IoT technology is presented. In Section 4, the main requirements of the proposed testbed system are discussed. Sections 5 and 6 present the system design and implementation of the proposed IoT testbed, respectively. Section 7 illustrates some example use cases of the testbed. A description of the system evaluation and discussion of the results are presented in Section 8. Section 9 is the conclusion to this paper.

2. Related Work

The research community has realized the evolving IoT technology and recognized the need for effective IoT experimentation. As indicated in [1], it is a common research practice for the simulation-based IoT studies to be validated with physical testbed experimentation. There have been a number of research efforts made to develop and deploy different physical IoT experimentation testbeds. Most of these testbeds were made publically available and openly accessible with varying facilities and functionalities. This has led to the emergence of new concepts in regard to IoT experimentation.

These include the IoT Testbed-as-a-Service (TaaS) [2] and Experiment-as-a-Service (EaaS) [3].

In [4], a small-scale testbed with cost-effective and easy-to-build design was presented. It was deployed at the building of Electrical Engineering Department of the University of North Texas. The design of the testbed was based on a single base station and a set of sensor nodes running the Zigbee protocol and having different sensors. In addition, a number of software components for sensor data transmission, gateway functionality, database management, and user interface were implemented.

The Twonet testbed was proposed in [5] with a large-scale design. It was deployed throughout a university building of four floors. The main objective of Twonet is supporting multichannel wireless sensor networking. The design of Twonet is based on three-tier architecture. At tier 1, a Linux PC server exists to control testbed access and experimentation through a custom web interface. It also has a set of 20 Raspberry Pi nodes, interconnected using PoE switch, as proxies at tier 2 for collecting debug logs and sensor node data. Each proxy interconnects a set of sensor nodes (at tier 3) implemented using Opal sensor devices.

Another large-scale testbed is SmartCampus [6] located in the Centre for Communication Systems Research (CCSR) building at the University of Surrey, UK. It was developed as part of the European SmartSantander experimental facility [7]. It was deployed across an entire building of three floors in a real-world environment with realistic operational conditions. The testbed had a user-centric design which enables effective study of user behavior in IoT environments and the efficiency of loT solutions. The testbed was designed following a 3-tier model connecting a number of IoT nodes to different gateways to be then connected to a cloud. Each IoT node was implemented using a TelosB mote with different sensors and a IEEE 802.1 5.4 radio module. The testbed also incorporates 30 Android Smartphone carried by the users and connected to the gateways. The testbed relies on a management framework for configuring and controlling its functionalities. It provides a user interface enabling resource discovery and reservation, topology display, experiment configuration and execution, and data analysis.

In [8], the testbed was built with a small-scale setup and designed for different IoT application domains. The main objective of the testbed is providing a low-cost, reliable, and customizable testbed. It enables low power consumption, test repeatability, and code portability. The testbed design is based on a number of sensor nodes wirelessly connected a dedicated gateway using nRF24L01+ chipsets with low power consumption. The sensor nodes and gateway were implemented using Arduino UNO and Arduino DUE, respectively. The gateway is connected to a server which runs three software components: Python adapter server, Apache web server, and MySQL database server.

Supersensors were presented in [9] as a campus-wide testbed for the University of Glasgow. It was based on a distributed data collection system with flexible and scalable microcomponent design architecture. The testbed can be used for implementing different use cases for smart campus development. The testbed architecture is composed of a number of sensor nodes (Raspberry Pi with a set of sensors) connected to a central server. The sensor nodes run different daemons for real-time sensor data collection and network connectivity establishment. At the server side, a publish/subscribe model with a queuing system is implemented for effective real-time data processing and storage in a database. The Protocol Buffers language (proto3) is also used for serializing and parsing data. In addition, a heartbeat mechanism is implemented in the server to enable remote health and availability monitoring as well as reprogramming and configuration of the sensor nodes. The server provides a user interface allowing users to access node and sensor data using REST APIs.

In [10], another IoT testbed deployed at Okayama University of Science across a five-floor academic building was presented. The testbed was constructed using a set of Note PCs and Raspberry Pi devices. A Distributed Internet Traffic Generator software was implemented to generate different types of network data traffic in the testbed. It was also used to obtain information about different metrics in regard to data transmission. The testbed enables effective evaluation of IoT protocols and algorithms considering different realistic networking scenarios and parameters.

In [11], an IoT testbed was developed for dependability competition to quantitatively test low-power wireless sensing systems in environments with high radio interference. The testbed was deployed using a number of TelosB devices with low-power wireless radios in a building at Graz University of Technology in Austria. A Raspberry Pi running a control software is interfaced with each TelosB device using an open-hardware interface board capable of energy consumption monitoring. Users can load their binary images into the boards to run IoT experiments for certain duration. The system provides different performance measures such as delay, consumed energy, and reliability.

An IoT testbed developed following a minimalistic approach to maintain cost-effective implementation was presented in [12]. It was deployed at the Inria Paris across two multistory buildings. The testbed was built using a set of nodes; each has a number of components contained in an OtBox. These include Raspberry Pi, multiple OpenMote devices, and a screen. Each OtBox has a testbed control software running on the Raspberry Pi and connecting to an MQTT broker using WiFi connectivity. Users can use APIs for mote configuration and OtBox management. They also can use the screen to interact with the system using a web-based user interface with a dashboard running as a service on IBM Cloud. The testbed is an open-source and open-hardware system which provides open and remote access.

In [13], the MakeSense testbed which enables real-life and large-scale IoT experimentation for social research was introduced. It enables real-time indoor activities monitoring and situation-aware applications. The testbed design is based on a scalable and flexible client-server model. A set of sensor nodes were deployed as clients and connected to a server using the Lightweight Machine-to-Machine protocol through Broadband Internet. The server is hosted in the cloud for better IoT data analysis and visualization. The testbed provides API access to sensor nodes for remote management and configuration of available resources. The testbed features different security implementations including the Datagram Transport Layer Security (DTLS), preshared key authentication, and data encryption using Advanced Encryption Standard (AES).

An online testbed for large-scale LoRa experimentations deployed across multiple apartments and buildings was presented in [14]. It has a three-layer design architecture. These are the node, gateway, and cloud server layers. At the node layer, a set of Raspberry Pi were implemented and powered using power banks and solar energy. The gateway node at the next layer is responsible for packet forwarding from the nodes to the server. The server layer provides different functionalities including user management, system control, and job schedule. The testbed provides web-based user interface enabling users to configure the nodes and receive real-time experimentation display.

In [15], an IoT testbed targeting effective combination of IoT technologies and web technologies was presented. It was deployed in a home environment for monitoring and controlling different physical parameters. It has a three-layer architectural design which includes the sensing, middleware, and application layers. A set of sensor nodes were implemented using Arduino devices at the sensing layer. These nodes collect and send IoT data to a gateway PC (at the middleware layer) using the Zigbee protocol. At the application layer, a web server was implemented to enable user access and system configuration using HTTP.

Table 1 summarizes and compares these testbed systems with the proposed one. It can be noticed that most of the presented IoT testbeds target specific IoT communication technologies, sensors, and applications. This work factors in the diversity of IoT technology and the variety in IoT setups and properties. The focus is on facilitating general-purpose and multidimensional support of IoT experimentation while considering different IoT applications. It also envisages the importance of developing an IoT testbed platform for advancing research activities in the College of Computer at Qassim University. The main focus is to build a system that will provide researchers with an efficient IoT facility to promote more effective IoT experimentation.

3. IoT Overview

The IoT technology enables merging the physical and virtual worlds by interconnecting most of the physical objects around us to the Internet. It extends computing capability and network connectivity to everyday objects and enabling them to generate and disseminate data. These objects could include home appliances, wearable devices, medical equipment, and vehicles.

IoT systems are based on the combination of different technological domains such as sensor technologies, networking protocols, embedded systems, cloud integration, data processing, and communication technologies. The IoT technology makes a great push towards digital intelligence and smart control and automation with less human intervention.

The IoT technology opens the horizon for a wide scope of smart applications in different domains such as transportation, healthcare, agriculture, utilities, and surveillance. Smart city is an example of IoT application that can be applied to realize better management of different aspects such as street light, traffic, and daily services. Another example application is fleet management which enables real-time tracking of fleet assets. In addition, IoT can also be applied for healthcare applications to develop remote patient monitoring and alerting systems.

Most of the IoT applications require large deployment of interconnected IoT nodes as presented in Figure 1. These nodes are usually implemented with small-sized devices limited in computation, memory, and energy resources. Wireless sensor networks are typically established among these nodes to be then connected to the Internet using IP-enabled gateways. Each node is capable of sensing and collecting IoT data targeting specific application. Data communications can be established between the IoT objects and the Internet as well as among the objects themselves.

To this end, IoT relies on different essential hardware and software components. These include sensor devices which are important for reading varying parameters of the physical world. These could be environmental, meteorological, ecological, and medical parameters. Each node also must have a processing unit which can be built using a microcontroller/-microprocessor and memory to support data processing and storing. The communication module is also another essential component for any IoT system to enable data transmission to the Internet. It is evident that wireless and low power communication is critically required in this context as most of the IoT nodes would be battery-powered. Example technologies for wireless and low power connectivity are Zigbee, Bluetooth Low Energy (BLE), 6LoWPAN, and LoRaWAN. All these node components need to be powered using a power supply unit which could be a battery or a solar cell.

In IoT systems, sensor nodes collaboratively sense certain parameters and communicate the readings with the gateway which then forwards the data to the Internet. During this, the gateway can also be used to carry out local IoT data preprocessing at the edge of the IoT network. IoT data is typically sent to cloud systems in the Internet for effective data storing and management. IoT applications generate big data that requires further application-specific data processing and analysis.

However, IoT poses different challenges which need to be addressed for different IoT solutions. These include effective support of security and high level of protection against potential IoT attacks. Privacy and the ownership of collected data also become critical concerns, in addition to the need for establishing clear privacy protection policies. Another challenge is concerned with legalization and having a global IoT governance system. It is also challenging to maintain interpretability and support of homogeneity in IoT systems. IoT solutions also require high level of Internet availability with effective cost management particularly for large-scale IoT deployment.

4. System Requirement Analysis

The IoT is an interdisciplinary concept that incorporates variety of computing disciplines such as networking, communication, embedded systems, cloud computing, and data

Table 1: Comparison of the proposed testbed with other systems.

Ref.	Scale	Objective	IoT communication	Implemented IoT devices	No of nodes	Sensors	Interface	Cloud integration	Data visualization	Node mobility	Portability
[4]	Small	Support of IoT applications related to environmental monitoring	Zigbee	RPi, Arduino, XBee S2B	6	Temperature, humidity	Web API	Not supported	Not supported	Not supported	Supported
[5]	Large	Support of multichannel WSNs	Dual-radio at 2.4 GHz & 900 MHz	RPi, opal mote, debug board	100	—	Web	Not supported	Not supported	Not supported	Not supported
[6]	Large	Real-life user-centric experimental IoT research facility	IEEE 802.15.4	TelosB mote, sensor board, GuruPlug, Android phones, Android tablets	200	Temperature, light, noise level, motion	Web API	Supported	Supported	Not supported	Not supported
[8]	Small	Support of a customizable and multipurpose experimentation	2.4 GHz radio	Arduino, ATtiny85, ATmega328P, ESP8266, nRF24L01+	—	—	Web	Not supported	Supported	Not supported	Supported
[9]	Small	Support of flexible data collection and processing	WiFi	RPi	10	Temperature, light, sound, motion	Web API	Not supported	Supported	Not supported	Supported
[10]	Small	Support of effective evaluation of IoT protocols and algorithms	Ad hoc	RPi, Note PCs	8	—	API	Not supported	Not supported	Supported	Supported
[11]	Large	Support of the dependability of low-power WSNs in environments with high radio interference	—	RPi, TelosB mote, NavSpark-GL, TI LMP92064	45	Light	Web	Not supported	Supported	Not supported	Supported
[12]	Large	Support of cost-effective large-scale IoT experimentation	IEEE 802.15.4	RPi, OpenMote, display screen	80	Temperature, humidity	Web API	Supported	Supported	Not supported	Not supported
[13]	Large	Support of real-life experimentation for social research on indoor activities	BLE	Arch Pro	100	Temperature, humidity, light, noise level, motion, dust density	Web API	Supported	Supported	Not supported	Supported
[14]	Large	Support of cost-effective LoRa experimentation in a large-scale and realistic environment	LoRaWAN	RPi, LoRa module	50	—	Web	Supported	Not supported	Not supported	Not supported

Table 1: Continued.

Ref.	Scale	Objective	IoT communication	Implemented IoT devices	No of nodes	Sensors	Interface	Cloud integration	Data visualization	Node mobility	Portability
[15]	Small	Support of effective combination of IoT technologies and web technologies	Zigbee	Arduino, XBee	—	Temperature, humidity, light, gas	API	Not supported	Not supported	Not supported	Supported
Proposed testbed	Small-large	General-purpose and multidimensional support of IoT experimentation	BLE Zigbee 6LoWPAN LoRaWAN	RPi, Arduino, GrovePi, XBee, 6LoWPAN module, LoRa module, UPS HAT battery	12 (extendable)	A wide range of GrovePi sensors	Web API	Supported	Supported	Supported	Supported

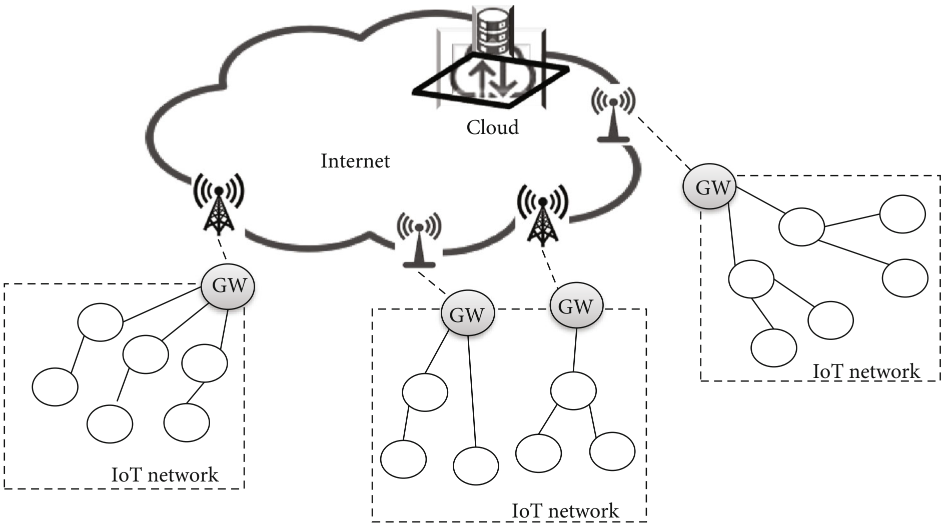


Figure 1: Typical IoT deployment.

science. IoT systems have distinct networking requirements in terms of low power communication and lightweight computing. It also relies on easy-to-setup IoT network, cloud integration, and data processing. This makes IoT system challenging to be developed and implemented. For effectively experimenting the different aspects of IoT technology, the design of the IoT testbed should consider different characteristics. These would include efficiency, practicality, simplicity, usability, and interactivity.

IoT architecture consists of different layers as presented in Figure 2. These are the sensing, network, data, and application layers. The testbed should be designed considering this architectural model in a modular design. At the sensing layer, it is important to have a variety of sensor devices that can capture wide range of IoT data. For the network layer, it is required to incorporate the different IoT communication technologies including Zigbee, BLE, 6LoWPAN, and LoRaWAN. In addition, support of different networking aspects such as data type, network topology, node mobility, network density, and node distribution should be considered. At the data layer, the system should be integrated with a cloud system that enables effective management and processing of the IoT data. The design also needs to facilitate easy use of the testbed to establish certain IoT applications.

Other key requirements include easy control of the system operations. Users should be able to easily configure the testbed for different IoT scenarios and applications. The system should enable the users to have control on which sensors to be used, communication technology to run, and application to build. It is also important to allow the users to configure each setup in terms of different aspects such as sensor reading interval and experiment duration. It is also important to enable the experimentation of different IoT networking problems such as load balancing, failure recovery, QoS, mobility, and security.

This also includes defining the scenario which each node should apply. For example, a setup can be configured to emulate a simple smart home application for 30 minutes using 5 IoT nodes interconnected over a Zigbee network. Each node is configured to read data from a temperature, pressure, light, and motion sensors every 10 seconds and send the data to the cloud. An experimental network scenario can be applied to this setup to have a multihop network topology. During the experiment, node failure can be triggered for certain period in order to examine network performance.

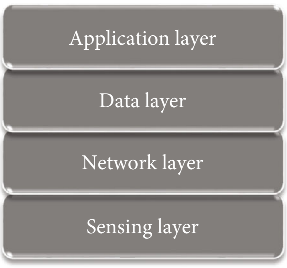


Figure 2: The layered IoT architecture.

Moreover, simple accessibility of the testbed needs to be realized for the different kinds of potential users. It is also important in this context to allow remote and online access to the testbed. On the other hand, secure design of the system should be realized to at least prevent unauthenticated access and unauthorized use of the system.

There are a number of nonfunctional requirements that need to be fulfilled by the system. These mainly include scalability, portability, flexibility, and cost-effectiveness. The system needs to be easily scaled up at the hardware and software levels. Making the system effectively portable and easily deployable is another important consideration. Different users can have different requirements, thus the design needs to be maintained general and flexible. It is important to keep the development cost to the minimum without compromising its functionality and quality.

5. System Design

The design of the proposed IoT testbed system is based on a modular and scalable architecture, and overview description

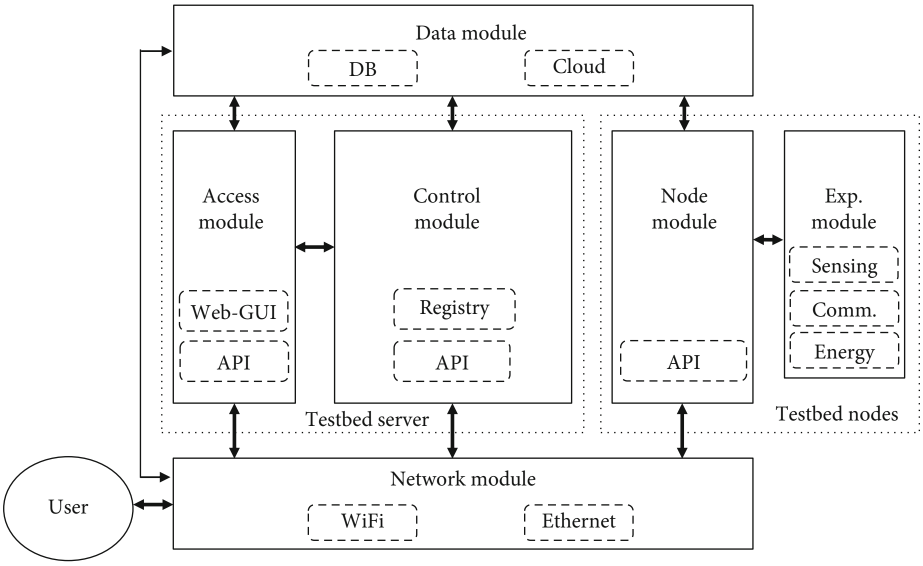


Figure 3: Overview of the testbed design architecture.

is provided in Subsection 5.1. A set of hardware and software components were incorporated in the system design as presented in Subsection 5.2. In Subsection 5.3, a detailed operational overview of the testbed functionality is presented.

5.1. Functional Design Architecture. Figure 3 presents an overview of the testbed design architecture. The modular design of the testbed is realized with different functional modules incorporated over the different architectural entities of the testbed. These entities include the central server and testbed IoT nodes. The current design of the testbed is composed of six main modules structured in a systematic manner as follows.

5.1.1. Access Module. The access module is the main front-end layer of the testbed. It facilitates simple and remote access to the system through a web-based Graphical User Interface (Web-GUI). It also enables user access using predefined Application Programming Interface (API) calls. In both cases, the module manages secure access to the system with reliable user authentication and authorization.

5.1.2. Control Module. The control module provides central control and management for the testbed system. It is responsible for monitoring the health and availability of the testbed resources including the IoT nodes. It includes a node registry unit for real-time management of node state. The module is also responsible for managing system functionality and controlling user experiments. It receives user requests from the access module once submitted through the web-GUI. User request can also be received from the user directly. In both cases, the commands are sent as API calls. Then, the control module manages experiment configurations, initiates experiments with the nodes, and controls experimentation setups. Logs on current system state in addition to the data of experiment input and output are collected by the module to be then sent to the data module for permanent storage.

5.1.3. Node Module. The node module manages the different operations of the node entity. It applies requested experiment configurations as received from the control module or directly from the user using the system's APIs. The node module is then responsible for running and controlling experimentation setups. It also communicates real-time data of experiment output with the control and data modules.

5.1.4. Experimentation Module. The experimentation module contains the sensing, communication, and energy units of the node entity. These are the main elements for setting up IoT experiments and applications on the testbed. The experimentation module controls the operations of these units according to the commands received from the node module. Once running an experiment, the module collects sensor readings, communication measures, and energy consumption information. These outputs are then delivered to the node module in a real-time manner.

5.1.5. Data Module. The data module maintains the required functionality for data storage and management. It is responsible of permanently storing system logs, experiment input, and collected experimentation data in the database unit. It also manages user data for enabling secure access to the system. The data module also provides effective integration with web-based cloud systems. The output data of user experiments can be forwarded to the cloud for further data management and processing.

5.1.6. Network Module. The network module provides the underlying communication infrastructure to interconnect most of the system modules. It enables wired and wireless connectivity to the system via WiFi and Ethernet, respectively. The network module is responsible for establishing a system control network among the testbed entities, users, and integrated cloud systems.

5.2. Main Design Components. The testbed design architecture is composed of two main architectural entities of the

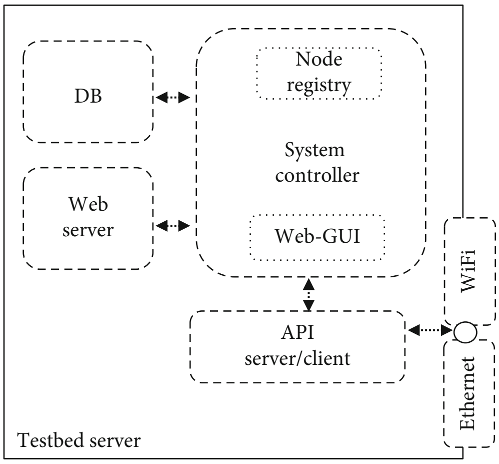


Figure 4: The architectural components of the server entity.

testbed. These are the central server and testbed IoT nodes. Both entities are interconnected over a wireless control network providing permanent Internet access.

The server entity consists of a set of functional components as shown in Figure 4. These include the system controller, web server, and database:

(i) The controller runs the functionality of the control module at the server entity. It facilitates effective testbed management and experimentation control at the main back-end layer of the system. It maintains a node registry service for managing the availability and monitoring the performance of the testbed nodes

(ii) The web server provides the functionality of the access module at the server entity. It enables web-based access to the testbed system in addition to facilitating easy configuration and setup of the different testbed resources. It also provides the sufficient support to prevent unauthenticated and unauthorized user access

(iii) The server manages a database that provides local data storage as part of the data module. It mainly retains user access information, system logs, and experiment data

The testbed node entity represents a set of IoT node devices. Each of these nodes has a number of hardware and software components as shown in Figure 5. The hardware components are the sensors, communication devices, and power source:

(i) The sensors include a set of IoT sensor device that measures a set of parameters such as environmental, biochemical, physiological, behavioral, and visual ones

(ii) The communication devices enable establishing wireless IoT networks among the IoT nodes. The main IoT communication technologies currently supported by the testbed are LoRaWAN, 6LoWPAN, Zigbee, and BLE

(iii) The design of testbed also relies on portable and sustainable power unit that allows for monitoring power consumption and controlling remaining energy

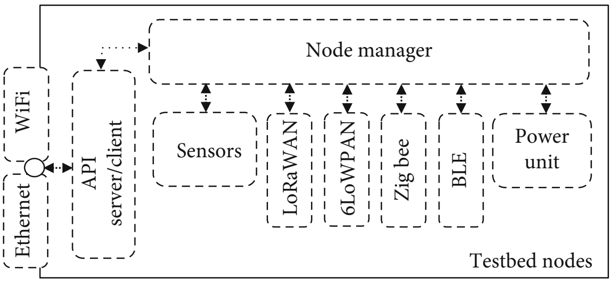


Figure 5: The architectural components of the IoT node entity.

These components of the IoT nodes support easy reconfiguration, lightweight portability, cost-effective implementation, and effortless deployment. The experimentation module is completely run at the node entity to manage the operations of the hardware components. It enables each component to apply relevant user configurations for effectively running the requested experimental setup.

The node entity also includes a central software component which is the node manager. It runs the functionality of the node module to control the node operations. These operations include initial node registration and frequent status reporting with the system controller. In addition, the node manager controls user experiments according to the requested setup and component configurations. For example, it manages turning on/off specific communication module, initiating sensor reading, and communicating IoT data. The node manager also performs real-time collection and transmission of experiment data to the system controller at the server. The data can also be sent by the node manager to the cloud if required by the user.

For better system usability, the design facilitates easy and simple access via an interactive web-GUI, which enables the following:

(i) Resource discovery and reservation: the system allows the reservation of a certain number of the nodes, that are currently available, for each experimentation setup

(ii) Experiment configuration and execution: users can select which communication modules and sensor devices to run for each selected node. There are also entries for setting experiment duration and sensing interval. Users can also provide experimental script to apply certain experiment scenario on the selected resources. This enables setting network topology, setup mode, data traffic type, and packet size

(iii) Topology display: the interface incorporates an experiment representational display for real-time visualization of the currently running setup. It visualizes the flow of IoT data among the experimentation resources during each run

(iv) Data dashboard: the interface provides visual display of collected IoT experiment data in a real-time manner

The system also includes other software components such as the API servers and clients at the server and node entities. User requests and experiment configurations are exchanged among the server and client using predefined API calls. After the initial startup time, each node runs an API server that waits for experimentation calls from the API client. User input via the web-GUI is firstly parsed into API calls and then transmitted by the API client to the API server. These are then processed and forwarded to the node manager for running the requested experimentation setup. The API server also accepts API calls from users directly in the case of having no functioning control module.

The communication of API calls among the different entities is maintained using the network module. This is carried out over a wireless control network established using a private WiFi access point. Using wired Ethernet connectivity is also supported as an alternative option. In both cases, the entities are provided with Internet access.

5.3. Operational Overview of the System. The sequence diagram presented in Figure 6 provides an overview of the main operations of the testbed system. At the startup stage, each node discovers its currently available experimentation modules and hardware components. At the node entity, the node module sends a status check message to the experimentation module which responds with status information of the available experimentation resources. Then, the node module sends a registration request message to the control module. This message contains different node information including its ID, IP address, available experimentation components, and remaining power. The message is processed by the node registry unit to maintain the availability and monitor the performance of the testbed nodes. It then replies to the node module with a response message. The node information is also sent to the data module for permanent storage of the data in the database.

User access to the system is managed by the access module during the login stage. New users are required to firstly register to the system. The registration information is forwarded to the data module for permanent storage. After receiving access request, user data is obtained from the local database to carry out secure user authentication and authorization. Users receive access tokens if and only if the access is legitimate.

User requests for IoT experimentations can be issued through the web interface or API calls. In the former case, experiment configurations and setup script entered by the user are forwarded to the control module in a request message. After copying the request data into the database, the control module forwards the message to the node module using internal API calls. The request is processed by the node manager software at the testbed nodes selected for the current experimentation. It then replies with a response message which is forwarded by the control module to the display unit for user notification. Upon that, the node manager sends the user configurations to the experimentation module to configure the sensing, communication, and energy components accordingly. After that, a confirmation message is sent back to the node module in order to signal the readiness of the experimentation resources. The node manager can then issue a run command and get the experiment started. It also applies the setup script to control the experimentation scenario accordingly.

For the experiment duration, the experimentation module keeps sending the collected IoT data and measures to the node module in a real-time manner. These are then forwarded to the control module which forwards them to the access module for real-time visualized display on the web interface. The result data is also sent to the data module for local storage and cloud backups.

In the other case, users issue experimentation requests directly to the testbed nodes using API calls. This supports system availability during the failure of the server entity. The request message is received and processed by the node module at each of the selected IoT nodes. The node manager issues a response to the user, copies the request data to the data module for cloud storage, and then sends the user configurations to the experimentation module. It also initiates the experiment after receiving a setup confirmation message from the experimentation module. Once the experiment started, the collected IoT experimentation data is forwarded to the node manager in a real-time manner. The data is then forwarded to the data module to be uploaded to the cloud system for further data processing and visualization. The user then can freely access the data at the cloud anytime the data module using the given cloud API calls.

6. System Implementation

The proposed IoT testbed system was implemented based on integrating a set of hardware components and developing different software modules. A representational overview of the testbed implementation is presented in Figure 7. The following subsections, 6.1 and 6.2, provide a thorough description of the current hardware and software implementations, respectively.

6.1. Hardware Implementation Overview. A collection of open-source and cost-effective hardware components was utilized to develop the current implementation of the testbed. These include a powerful PC laptop which was temporarily set up as the testbed server. It is wirelessly connected to a private WiFi Access Point (AP) which provides access to the Internet and the cloud service. The IoT nodes of the testbed were implemented using multiple Raspberry Pi boards of Version 3 and Model B. The nodes are wirelessly connected to the AP to establish a control network with the server and gain Internet access.

Each Raspberry Pi board runs a stable and open-source Linux-like operating system, namely, Raspbian.

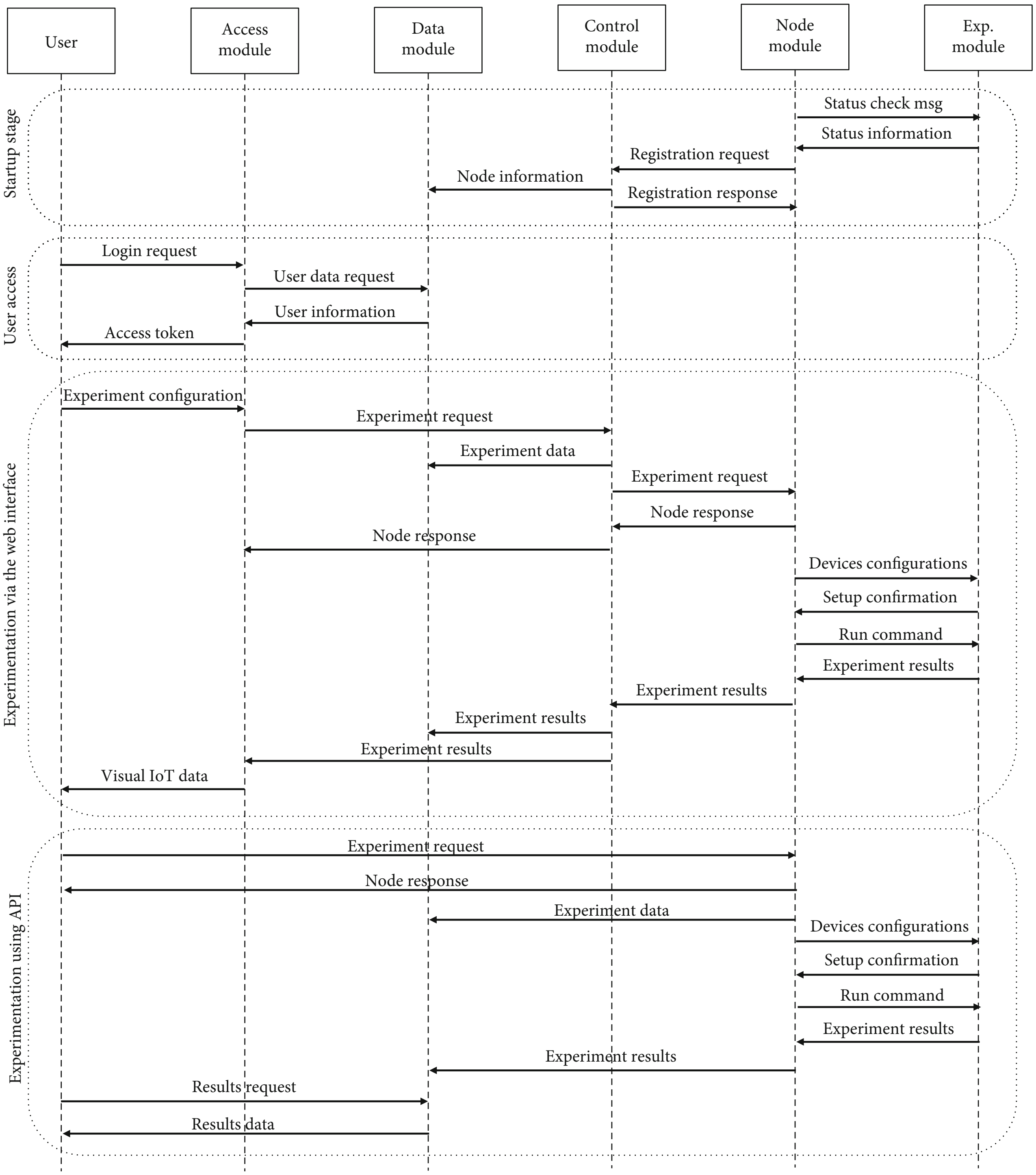


Figure 6: Operational overview of the testbed system.

The decision of adopting the Raspberry Pi board was made considering its capabilities in terms of CPU, memory, and connectivity. It also has a built-in WiFi and Bluetooth Low Energy (BLE) modules which are essential components of the testbed. In addition, it supports easy integration of wide range of IoT sensors and communication modules over its I/O interfaces such as the General-Purpose Input/Output (GPIO) pins and serial ports. These were used in the current implementation to attach the following external hardware modules to each Raspberry Pi board:

6.1.1. XBee S2C. XBee S2C is an RF module developed by Digi International to support short-range wireless Zigbee communication.

6.1.2. 6LoWPAN Chip. A tiny 802.15.4 radio chip was developed by OpenLabs to provide support for 6LoWPAN over

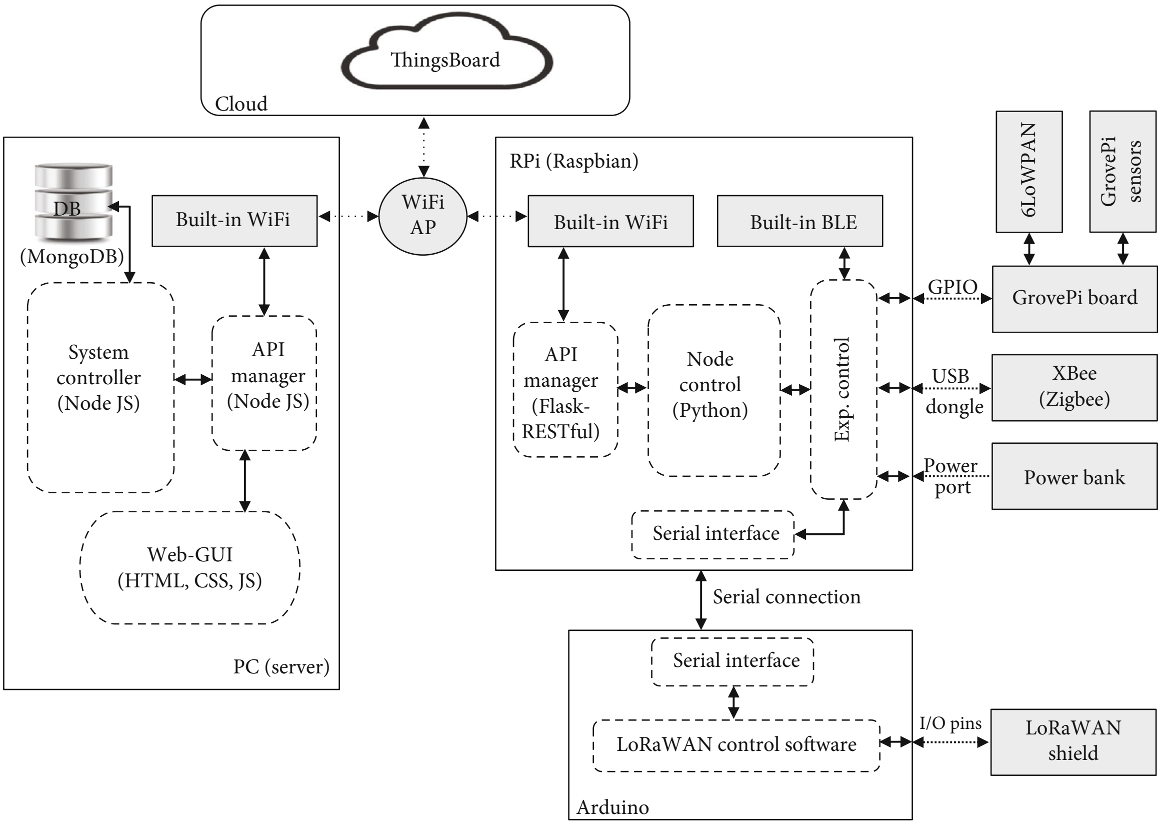


Figure 7: Representation of the current implementation of the system.

802.15.4 to the Raspberry Pi board. This RF module is designed to be attached to the GPIO pins of the board.

6.1.3. Arduino Board. Arduino is a microcontroller board that provides an open-source development environment. It has a number of digital and analog I/O pins which can be used for further extensions such as sensor devices and communication modules. There are different versions of Arduino, and the Arduino UNO version is the one being used in the current implementation.

6.1.4. LoRaWAN Arduino Shield. The LoRaWAN shield was developed by Dragino for Arduino UNO boards. It provides a long range transceiver that is based on open-source library. The shield has a set of I/O pins which were used for direct attachment to the Arduino board.

6.1.5. GrovePi+. GrovePi+ is an add-on sensor board developed by Dexter Industries to provide a simple solution for sensor attachment. It provides 7 digital, 3 analog, and 3 I2C ports. The testbed has GrovePi+ boards mounted on each Raspberry Pi board.

6.1.6. Collection of Compatible Sensor Devices. These are connected to the different ports of the GrovePi+ board. These include temperature, humidity, barometer, dust, air quality, gas, light, sound, and motion sensors. However, users are provided with the flexibility to replace these sensors with a variety of alternative sensors to support different IoT applications.

6.1.7. External Power Module. The testbed provides two different powering options for the battery modules. These are a portable power bank and the UPS HAT Battery Adapter Power Supply Extension Board. The testbed also supports the integration of any other power sources such as integrated solar cells.

Each Raspberry Pi board in the current implementation has a GrovePi+ board attached using the GPIO pins. The sensor devices are directly connected to the ports of the GrovePi+ board. The 6LoWPAN module is also attached to the extended GPIO pins on the GrovePi+ board. On the other hand, The XBee S2C module is attached to a customized dongle which is connected to the Raspberry Pi board via its serial port. The communication between the board and the chip is established over a serial connection.

In addition, another serial port is used to connect each Raspberry Pi board to an Arduino board which is used to mount and operate the LoRaWAN module. This can also be used as a further development board to support the scalability and flexibility of the system. However, the Arduino board and LoRaWAN module are completely controlled by the experimentation module implemented in the Raspberry Pi. This is maintained over a permanent serial connection.

In regard to the power module, it can be a power bank that is connected to the micro-USB power input of the Raspberry Pi board. It also could be the Battery Adapter Power Supply Extension Board which can be attached to the GPIO pins.

All the hardware components of each IoT node are contained in a compact-sized and lightweight box in order to support the portability of the system. Users are provided with the ability of placing and moving the IoT nodes in a flexible manner. The current implantation was developed with 12 IoT nodes. However, the scalable design and flexible implementation of the system allows for easily integrating additional IoT nodes in future.

6.2. Software Implementation Overview. A functional implementation of the main modules of the testbed design was developed over the different hardware components of the system. This was accomplished using a set of software development tools as follows:

6.2.1. Access Module. The web-GUI was developed as the front-end of the system using HTML, CSS, and JavaScript.

6.2.2. Data Module. The database was implemented using MongoDB. For the cloud service, the current implementation is integrated with ThingsBoard, a public cloud platform. It provides a set of APIs which was used for managing the communications with the cloud service.

6.2.3. Control Module. The control module's functionalities were implemented as the system back-end using Node JS.

6.2.4. Node Module. The different functionalities of the node module were implemented using Python and Flask-RESTful.

6.2.5. Experimentation Module. The communication, sensor, and battery submodules were also developed using open-source Python classes. As it was attached to an Arduino board, however, the development of the LoRaWAN submodules was carried out in the Arduino C-like environment.

The current implementation of the testbed server includes a web-GUI run by the access module. The interface enables a testbed user to input a setup script that describes the main experimentation setup of an experiment. The script is formed according to the user selections of the number of nodes, communication module, sensors, setup mode, scenario type, and experiment duration. A variety of experiment configurations can also be entered to configure the components of the established setup. These include sensing interval, transmission interval, packet size, battery level threshold, node operational mode, communication channel, and network topology.

The testbed server also incorporates a Node JS implementation of the system controller which provides the different functionalities of the control module. It runs the node registry which relies on a developed heartbeat mechanism for effective remote monitoring of the IoT node health and availability. The mechanism was implemented to frequently examine the server node connectivity to detect any failure.

On the other hand, the system controller was also implemented to receive setup script and experiment configurations in a POST request. The request data is then copied into the experiment table of the system database and forwarded to the selected nodes. The control module handles the communications with the IoT nodes using a set of developed REST API calls. Received API requests by the node module are handled by the Python implementation of the node control software. It processes the setup script to control the experiment accordingly. The experiment configurations are obtained by the experimentation control module as input to configure its components accordingly. Once the experiment is run, data are obtained and processed by the node control software as JSON data. It is then sent back to the server in response to the API request. The data is frequently obtained by the access module for real-time display on the interface.

7. Use Case

The current implementation of the proposed IoT testbed system enables running effective experiments to examine different aspects of the IoT technology. These include the characteristics of IoT communication in regard to different parameters such as link quality, transmission range, power consumption, and data rate. This can be carried out in experimental scenarios defined according to different considerations including setup size, network topology, hardware configurations, and traffic type. Users can select to run experimental setups using one of the implemented IoT communication technologies, namely, Zigbee, 6LoWPAN, LoRaWAN, and BLE. These would also facilitate studying different properties such interference and noise for different IoT network setups. In addition, the portability of the testbed nodes enables customized node positioning and allows for flexible establishment of mobility setups and dynamic scenarios. It facilitates studying network performance during the mobility of certain nodes or considering the movement of the whole network under realistic conditions.

Experiment can also be carried out for effective IoT data collection and processing. Since the testbed provides many sensing options using different sensor devices, data can be easily acquired for environmental, meteorological, ecological, and many other purposes. The integration with the cloud systems enables effective processing and simple visualization of the collected data. Therefore, a variety of IoT applications can be established using the testbed for different purposes such as monitoring, control, or automation. Examples include indoor environmental quality measurement, human activity monitoring, smart campus, indoor asset tracking, and smart classroom management.

To present a usage example of the testbed, an experiment was carried out to measure energy consumption of selected IoT communication technologies. In this experiment, the testbed was accessed using API calls which are received and processed by the nodes directly. Output data was then obtained from the node after having the experiment completed. Table 2 lists the main setup and configurations of the experiment. It was carried out over a simple network topology of only five IoT nodes. Three networking setups were developed to interconnect the nodes in a start topology using different IoT communication technologies. These are BLE, Zigbee, and 6LoWPAN. In all the scenarios, however, each node was configured to sense and send temperature sensor data every 5 seconds during a period of 45 minutes. Each setup was repeated for 10 times, and the battery of each node

Table 2: Experiment configurations.

Parameter	Setting
Number of nodes	5 nodes
Communication modules	BLE, Zigbee, and 6LoWPAN
Scenario	Data transmission
Duration	60 minutes
Transmission interval	4 seconds
IoT data type	Temperature sensor readings
Data packet size	Low-sized payload
Setup mode	Static
Battery charge status	Fully
Network topology	Star
Number of runs	10 runs

was fully recharged for each run. Data indicating the consumed energy is collected for each run, and the average consumption for each setup is presented in Figure 8. It can be noticed that low energy consumption was maintained in all the setups. However, data transmission over BLE relatively consumed less battery power compared to the Zigbee and 6LoWPAN setups.

The testbed system is also now being used for a new IoT experiment which requires the deployment of the testbed nodes across the building of the College of Computer at Qassim University. The main objective of the experiment is to experimentally examine the effect of IoT node positioning on the overall network performance in indoor environments. The focus is on two different IoT communication technologies which are Zigbee and LoRaWAN. Ten nodes are currently utilized to run different scenarios with different topological setups. In each of these scenarios, the nodes are placed in varying locations inside the building. In some of the scenarios, some of the nodes will be mobile with a simple movement pattern. This would enable to establish different setups considering different criteria such as topological distance and the presence of obstacles.

8. Evaluation

For effective evaluation, the system was assessed using the Technology Acceptance Model (TAM) [16]. It is a commonly adopted method for analyzing technology usage behavior and user attitude. Accordingly, a TAM-based questionnaire was developed for this study in respect to the common TAM factors. The questionnaire was then filled by a total of 46 participants who were involved in the use of the system.

The TAM-based questionnaire was formed with the main TAM factors. These are as follows: perceived usefulness, perceived ease of use, attitude toward usage, and behavioral intention to use. The questionnaire basically measures 16 items according to a 5-point rating scale ranging from “1: strongly disagree” to “5: strongly agree”. Figure 9 presents the TAM model of the system.

The participants were grouped in different groups of 2-3 participants, and some of the participants used the system individually. There were a total of 16 different groups and 4 individuals. There was good diversity among the participants in terms of age, educational level, and technical knowledge. That is, age range of the participants was 19-48 years. About 70% of the participants were MSc and BSc students, whereas the rest were PhD and MSc degree holders. In terms of their technical knowledge, most of the participants had no IoT experience, whereas few of them had some IoT knowledge.

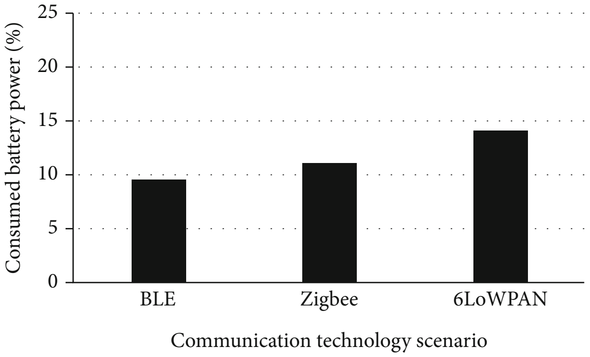


Figure 8: Experimental results.

The proposed testbed was used by all the participants to test the different functionalities of the testbed. This took place in one of the labs at the College of Computer in Qassim University. The participants were involved in building and running three IoT experiments considering IoT data collection and processing, IoT networking using the different communication technologies, and the development of IoT applications. The main details of these experiments are presented in Table 3. Each experiment was developed for certain user level, ranging from basic to advanced, and configured for specific testing duration. The experiments also varied in terms of the number of nodes, communication technology, and topology setup. The participants also used different types of sensors with varying sensing interval in each experiment. They also experienced different modes to access and configure the system. It is important to note that every participant group and individual was involved in at least two of the experiments; each was conducted for only a single run.

The evaluation was based on the observation of the participants during their use of the testbed. The TAM questionnaire was also filled by every participant individually afterwards. After collecting the evaluation data, the reliability of the TAM results was firstly evaluated. This was based on calculating Cronbach’s alpha coefficient for each of the considered factors. It is a consistency coefficient representing the dimensionality of the TAM questionnaire’s questions. The resulted coefficient values are listed in Table 4. As it can be seen, high value of Cronbach’s alpha coefficient was achieved by each factor. There was no coefficient value lower than the minimum acceptable value of 0.7.

On the other hand, Table 5 shows the evaluation results of the TAM questionnaire. It lists the resulted values of the calculation of the scoring average and standard deviation considering each TAM factor. It can be seen that the proposed IoT testbed was able to achieve good evaluation results

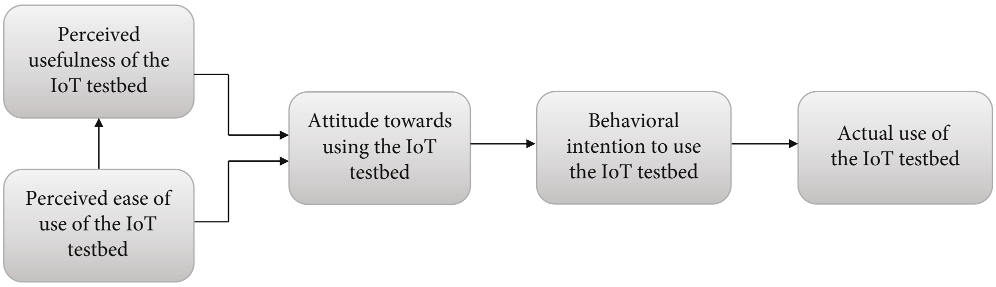

Perceived usefulness of the IoT testbed
Perceived ease of use of the IoT testbed
Attitude towards using the IoT testbed
Behavioral intention to use the IoT testbed
Actual use of the IoT testbed

Figure 9: The TAM model of the proposed IoT testbed.

Table 3: The main details of the experiments for the TAM evaluation.

Setting	Exp. 1	Exp. 2	Exp. 3
Experiment type	Smart home application	IoT data processing	IoT networking
Level	Basic	Intermediate	Advanced
Duration (min)	25	35	45
No. of nodes	6	4	8
Communication tech.	BLE	Zigbee	Zigbee 6LoWPAN LoRaWAN
Network topology	Star	Star	Star Multihop
Sensors	Light Motion Gas	Temperature humidity air quality	Temperature
Sensing interval (sec)	5	10	5
Data packet size	Small	Small	Small-large
Access mode	Web-GUI	Web-GUI API	API
Data storage	Cloud	Cloud Local	Local
Participant no.	13 groups 3 individuals	15 groups 4 individuals	8 groups 1 individual

Table 4: Cronbach's alpha coefficient results.

Factor	Cronbach's alpha coefficient
Perceived usefulness	0.914
Perceived ease of use	0.882
Attitude toward usage	0.931
Behavioral intention to use	0.844

Table 5: Average scoring results.

Factor	Average	STD
Perceived usefulness	4.41	0.71
Perceived ease of use	4.34	0.76
Attitude toward usage	4.25	0.68
Behavioral intention to use	4.23	0.79

as high average scores and low standard deviation was obtained for all the factors. It is evident that the testbed succeeded in maintaining usefulness and simplicity when it comes to realistic IoT experimentation. In addition, the testbed provides an attracting and motivating experimental IoT facility for the different users as they showed a positive attitude toward the testbed with good intention to use it in future for IoT experimentation.

Furthermore, the correlation between the considered TAM factors was measured and analyzed based on the calculation of the Pearson correlation (PC). This is a critical statistical indicator for analyzing how the dependency relationship formed among the factors. The calculation results of the Pearson correlation are listed in Table 6. It can be noticed that there was a positive correlation among the considered TAM factors for the proposed IoT testbed. The perceived usefulness factor well-correlated with the ease of use factor. Both of these factors had good correlation with the attitude toward the usage of the system. It is accordingly evident that the testbed attracted the different users to utilize its functionality as a result of its simplicity and efficiency for effective IoT experimentation. In addition, the relationship among the attitude toward usage and intention to use factors achieved a higher correlation value. This clearly indicates that the

Table 6: Relationships among TAM factors.

Independent factor	Dependent factor	PC	R^2
Perceived ease of use	Perceived usefulness	0.798	0.67
Perceived usefulness	Attitude toward usage	0.817	0.72
Perceived ease of use	Attitude toward usage	0.791	0.69
Attitude toward usage	Behav. intention to use	0.836	0.70

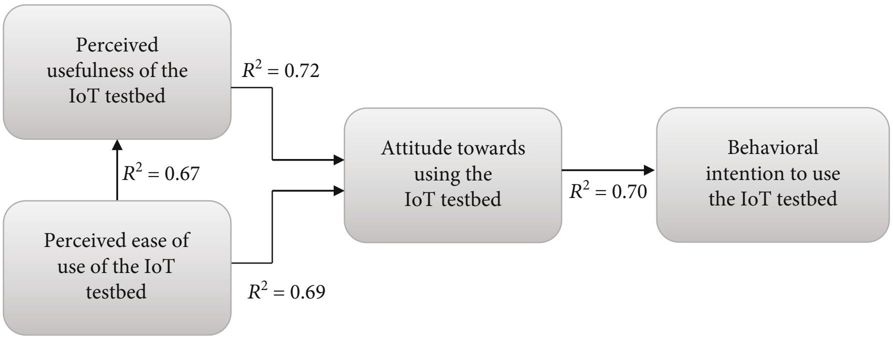

Perceived usefulness of the IoT testbed
$R^2 = 0.72$
Attitude towards using the IoT testbed
Behavioral intention to use the IoT testbed
$R^2 = 0.67$
$R^2 = 0.70$
Perceived ease of use of the IoT testbed
$R^2 = 0.69$

Figure 10: The TAM model with the R-square values.

system was able to maintain high level of user satisfaction and attract user interest for future IoT experimentation.

To determine the variance of the independent factors, the R-square is also calculated and listed in Table 6. It indicates the closeness of the data to the fitted regression curve. Figure 10 shows the calculated R-square values on the TAM model of the proposed IoT testbed. It is evident that these results statistically present effective variation in the response variable.

9. Conclusion

An effective IoT testbed system that takes IoT experimentation to a further level of simplicity and practicality was presented in this paper. The system provides a motivating experimental testbed environment for effective IoT experimentation. It enables researchers to carry out IoT experiments targeting the different aspects and applications of the IoT technology. The testbed design provides a usable, scalable, portable, and flexible system. Testing and experimenting IoT solutions was made inspiring in simple and practical design architecture. The current implementation of the testbed system was accomplished using a collection of efficient and cost-effective components. The results of the TAM-based evaluation demonstrated how the proposed testbed system is usable and useful for conducting experimental IoT studies in a realistic manner. The testbed received positive attitude in regard to its efficiency for IoT experimentation. The testbed was also found to be effective for future use in future IoT research works.

Given its design modularity and flexibility, the testbed can be extended to target more customized IoT experiments. In a future work, the testbed will be utilized to carry out a number of IoT experiments to study IoT network performance considering dynamic and mobile network setups. It is also envisaged that the system can be deployed for a smart classroom application which would require integrating and comparing some artificial intelligence techniques.

Data Availability

According to the funding policy of this work, data cannot be shared or made publicly available during the funding contract.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Deanship of Scientific Research, Qassim University, according to the agreement of the funded project No. SRD-coc-2018-1-14-S-5142. The author thanks the sponsor of this work for their support.

References

[1] G. Z. Papadopoulos, A. Gallais, G. Schreiner, E. Jou, and T. Noel, “Thorough IoT testbed characterization: from proof-of-concept to repeatable experimentations,” *Computer Networks*, vol. 119, pp. 86–101, 2017.

[2] M. Hossain, S. Noor, Y. Karim, and R. Hasan, “IoTbed: a generic architecture for testbed as a service for Internet of Things-based systems,” in *2017 IEEE International Congress on Internet of Things (ICIOT)*, Honolulu, HI, USA, June 2017.

[3] T. W. Edgar and T. R. Rice, “Experiment as a service,” in *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, Waltham, MA, USA, April 2017.

[4] S. Ferdoush and X. Li, “Wireless sensor network system design using Raspberry Pi and Arduino for environmental

monitoring applications," *Procedia Computer Science*, vol. 34, pp. 103–110, 2014.

[5] Q. Li, D. Han, O. Gnawali, P. Sommer, and B. Kusy, "Twonet: large-scale wireless sensor network testbed with dual-radio nodes," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, Roma, Italy, November 2013.

[6] M. Nati, A. Gluhak, H. Abangar, and W. Headley, "Smart Campus: a user-centric testbed for Internet of Things experimentation," in *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Atlantic City, NJ, USA, June 2013.

[7] L. Sánchez, J. A. Galache, V. Gutiérrez et al., "SmartSantander: experimentation and service provision in the smart city," in *Proceedings of the Future Network & Mobile Summit*, Warsaw, Poland, June 2011.

[8] A. Di Nisio, T. Di Noia, C. G. C. Carducci, and M. Spadavecchia, "Design of a low cost multipurpose wireless sensor network," in *2015 IEEE International Workshop on Measurements & Networking (M&N)*, Coimbra, Portugal, October 2015.

[9] K. Hentschel, D. Jacob, J. Singer, and M. Chalmers, "Supersensors: Raspberry Pi devices for smart campus infrastructure," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Vienna, Austria, August 2016.

[10] H. Oda, E. Kulla, R. Ozaki, and N. Nishihara, "Design of an adhoc testbed for IoT and WSAN applications using Raspberry Pi," in *Proceedings of the 11th International Conference on Broad-band Wireless Computing, Communication and Applications (BWCCA)*, Asan, Korea, November 2016.

[11] M. Schuß, C. A. Boano, M. Weber, and K. Römer, "A competition to push the dependability of low-power wireless protocols to the edge," in *Proceedings of the 14th International Conference on Embedded Wireless Systems and Networks (EWSN)*, Uppsala, Sweden, February 2017.

[12] J. Munoz, F. Rincon, T. Chang et al., "OpenTestBed: poor man's IoT testbed," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Paris, France, France, April-May 2019.

[13] J. Jiang, R. Pozza, N. Gilbert, and K. Moessner, "MakeSense: an IoT testbed for social research of indoor activities," 2019, http://arxiv.org/abs/1908.03380.

[14] W. Wang, X. Guo, L. Liu et al., "Dandelion: design of an online large scale LoRa testbed," in *Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN)*, Beijing, China, February 2019.

[15] A. A. Reshi, A. Alsaeedi, and S. Shafi, "Development and web performance evaluation of Internet of Things testbed," in *2019 International Conference on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, April 2019.

[16] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8876928, 11 pages
https://doi.org/10.1155/2020/8876928

Research Article

Agent-Based Multipath Management for Supporting Sink Mobility in Wireless Sensor Networks

Cheonyong Kim,[1] Hyunchong Cho,[2] Kwansoo Jung,[3] Yongbin Yim,[4] Taehun Yang,[5] Sang-Ha Kim,[5] and Sangdae Kim[6]

[1] *Advanced KREONET Center, Korea Institute of Science and Technology Information (KISTI), Daejeon 34141, Republic of Korea*
[2] *Research Institute for Computer and Information Communication, Chungbuk National University, Cheongju 28644, Republic of Korea*
[3] *Department of Fintech, Daejeon University, Daejeon 34520, Republic of Korea*
[4] *Network Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Republic of Korea*
[5] *Department of Computer Science and Engineering, Chungnam National University, Daejeon 34134, Republic of Korea*
[6] *Division of Computer Science and Engineering, Kongju National University, Cheonan Chungnam 31080, Republic of Korea*

Correspondence should be addressed to Sangdae Kim; sdkim.cse@gmail.com

Received 9 June 2020; Revised 6 July 2020; Accepted 1 August 2020; Published 5 September 2020

Academic Editor: Euisin Lee

In wireless sensor networks, sink mobility support is one of the essential functionalities in many applications. With continuous advancement, future applications will require not only sink mobility support but also high-performance data delivery service. Multipath routing is one of the promising technologies for improving data delivery performance by collaboratively using alternative or redundant multiple routing paths. However, existing multipath routing protocols had not dealt with sink mobility. As a result, they lead to bad performance in terms of energy efficiency due to the end-to-end path reconstruction. Consequently, a novel multipath management scheme is required thereby supporting sink mobility without performance degradation. In this paper, we propose a multipath management scheme for supporting sink mobility. The proposed scheme dynamically constructs multipath along the moving path of a sink. In addition, the proposed scheme provides the path shortening schemes according to the sink's movement for reducing energy consumption. Our simulation results show that the proposed scheme is superior to existing path management schemes in terms of reliability and energy efficiency.

1. Introduction

Sink mobility is a fundamental requirement in wireless sensor networks (WSN) because mobile sinks appear frequently in many applications as data collectors, administrators, and users [1, 2]. For example, soldiers in military surveillance, rangers in forest management, and biologists in habitat monitoring are representative mobile sinks in WSN-based data collection systems. The mobile sinks execute their duties with the aid of the adjacent sensor data. In addition, automatic mobile sinks (e.g., unmanned aerial vehicles (UAVs) and mobile robots) can perform challenging tasks in inhospitable areas without reliable infrastructure. The effective sensor data collection services that can be provided by sink mobility are promoting the practical development of wireless sensor networks.

Recently, various techniques for WSN have actively been researched because WSN is receiving attention as an infrastructure for the Internet of Things [3, 4]. In this situation, the new applications will naturally be developed and these applications will surely require more powerful network performance than existing applications. For example, in the habitat monitoring application, the biologist may want to collect not only the location of the target animals but also their statuses such as temperature, heartbeat, and posture. In this case, the network should provide enhanced data throughput in real time. To realize the applications, the sensor network has to be evolved toward high-performance networks.

The multipath routing is one of the network techniques to provide high performance in terms of reliability and throughput of packet delivery [5]. In multipath routing, multiple disjointed paths are established between a source and a sink by path discovery and path construction processes. These disjointed paths improve the packet delivery success ratio by transmitting multiple copies of a packet through the paths. In the redundant packet transmission, the packet delivery fails if and only if all of the paths fail. In addition, the packet delivery throughput can be improved by delivering multiple packets concurrently. Theoretically, the improved data rate of the concurrent packet transmission is proportional to the number of paths. That is, discovering disjoint multipath enhances the packet delivery quality significantly.

Traditional multipath routing protocols [6, 7] assumed that there is a static sink at a fixed location. Therefore, to support sink mobility, the existing protocols need to reconstruct the multipath whenever the sink moves. Multipath construction includes several subprocesses such as searching available paths, selecting appropriate paths, and establishing selected paths. Since the processes require a lot of message exchanges among the sink and the participating nodes, existing multipath discovery schemes lead to a fatal problem. That is, a large amount of energy is needed for reconstructing multipath so the network lifetime would be considerably shortened. Moreover, the data delivery cannot be conducted while the multipath is reconstructed. Technically, the multipath is constructed between a source and an agent. The agent acts as a static sink during a mobile sink is within its radio range. That is, the agent is the last-hop node of each path. Therefore, the multipath has to be reconstructed whenever the sink moves outside the radio range of the agent. In this case, the multipath reconstruction interval decreases as the speed of the sink increases and as the radio range of nodes decreases. The frequent multipath reconstruction dissipates nodes' energy so that the network lifetime will be shortened without performance improvement. In addition, the connection between the source and the sink is disconnected during the multipath reconstruction process. If the reconstruction delay is larger than the reconstruction interval, the connection between the source and the sink will not be restored.

Meanwhile, in single-path routing, a few works have dealt with the path management scheme for supporting sink mobility. They maintain the connection between a source and a mobile sink without end-to-end path reconstruction by the agent control mechanism. For example, in the footprint chaining mechanism [8], a mobile sink makes a connection between the closest neighbor node and the agent. This path extension is continuously conducted and the routing path is constructed along the moving path of the sink. However, in multipath routing, they lead to performance degradation since their path management mechanisms result in constructing a single path along the moving path of a mobile sink.

In this paper, we propose a novel multipath management scheme, called MPM, for establishing end-to-end multiple routing paths during the sink moves. As mentioned above, the existing studies lead to performance degradation because they deal with one agent. On the one hand, in the multipath routing schemes, the end-to-end multiple paths should be reconstructed when the link between the agent and the mobile sink is disconnected. The frequent path reconstruction leads to the excessive energy consumption of sensor nodes. On the other hand, in the single-path routing, a single path is constructed between the current location of a mobile sink and the initial agent. The single path management cannot provide high-performance data delivery. Intuitively, simply applying the single path management scheme to multiple paths seems to be a plausible solution for solving the problem. However, the independent management and disjointness maintenance are nontrivial challenge. As an innovative solution, MPM adopts the agent-based multipath management where an exclusive agent is assigned to each path. Figure 1 shows the radical difference between the existing approaches and MPM. For effective path management, we introduce two kinds of path management functions that are path shortening and path extending and provide different mechanisms for each function. In addition, we provide a discussion about the impact of the multipath reconstruction interval and neighbor list update interval which significantly influences the performance of the MPM.

The rest of this paper is organized as the following. Section 2 provides the background and related works of this paper. Section 3 describes MPM in detail including the data structure and path management algorithms. In Section 4, the results from our computer simulation are presented. Finally, Section 5 concludes this paper.

2. Related Work

In multipath routing, the node or link disjoint multipath is constructed between a source and a sink since the nondisjoint may lead to the negative impact in terms of reliability and aggregated bandwidth. The disjoint multipath improves the performance in terms of the packet delivery success ratio and the throughput [5]. First, the packet delivery success ratio increases by exploiting path redundancy. In this case, the probability of failure decreases as the number of multipath increases if multiple copies are redundantly forwarded through each path. This is because a packet delivery fails only if all paths fail. Second, the throughput increases by bandwidth aggregation. When using a single path, a source transmits one packet at a time. On the other hand, when multiple paths are available, the source can transmit multiple separate packets at a time. This shows the same effect as aggregating bandwidths of multiple paths. Alternative path routing [9] is another usage of multiple routing path where the secondary path is used if data delivery through the primary path fails. Due to the fact that each path should be independently used in multipath routing, establishing disjointed multipath is important for enhancing data throughput and packet delivery ratio.

Due to the great advantages, multipath routing has been an important research subject in WSN. In addition, recently, multipath routing is actively used for various application scenarios based on WSN. Mohanty and Kabat proposed a multipath data transmission scheme for healthcare

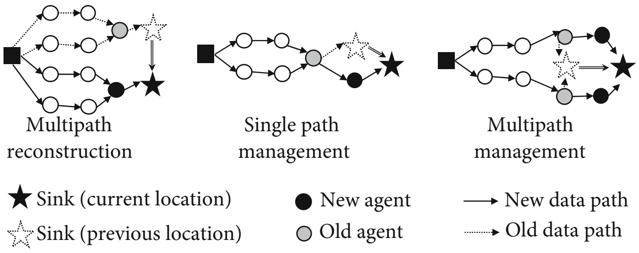


Figure 1: Path management classification.

application [10]. In this, the data packets are classified based on the criticality. The intermediate nodes are responsible for prioritized buffer management and congestion probability computation. In case of congestion, the emergency and sensitive data are transmitted through alternate paths. Hasan et al. proposed a multiconstrained multipath routing protocol for multimedia sensor networks [11]. Their mathematical model based on the Lagrangian relaxation adaptively controls the multipath routing protocol to balance the QoS parameters (i.e., the energy consumption and end-to-end delay constraints). Jaiswal and Anand [12] proposed the energy-efficient multipath routing protocol to improving QoS for IoT applications. Specifically, they focused on achieving high reliability in data packet delivery when the packets are unfairly generated in IoT environment. To this end, the sensors consider lifetime, traffic intensity, and reliability in the path construction phase so that the source can choose high-quality paths. However, the works focus on the performance enhancement and QoS provision in delivering data to a static sink. Therefore, frequent end-to-end multipath construction process should be conducted for supporting sink mobility.

The path management is the most important functionality for supporting sink mobility. The most fundamental management approach is reconstructing paths each time a sink moves, in other words, whenever the last-hop connection between the agent and the sink is lost. Primitive path construction schemes utilize the flooding method [13]. The mobile sink broadcasts its own location and sensor nodes store a neighbor who sends to the location update message as the downstream node while the location update message is disseminated in the network. When a sensor reports a data packet, it sends the packet through the downstream chain toward the sink. However, this scheme wastes lots of energy since the location update flooding is performed each time the sink moves.

The existing studies have focused on reducing flooding costs. Wang et al. exploited the local flooding approach [8]. In the study, a sink selects an agent among neighbors and then the agent floods its own location in the network. Then, the agent periodically performs flooding within the local area by a predefined range. The nodes outside of the local area could deliver data packets toward the agent, and nodes in the area send the data to the location that the sink currently exists. Luo et al. proposed the trajectory forwarding scheme for delivering data packets to the continuously moving mobile sinks [14]. In the trajectory forwarding, a mobile sink is associated with two agents, which are the primary and immediate agents. The primary agent acts as a static sink during the mobile sink is in a cell (i.e., distance-based periodic reconstruction). While keeping movement, the sink continuously elects new immediate agent and sends the location of the immediate agent to the primary agent for future data forwarding. The trajectory forwarding is an effective single path management scheme. Yu et al. proposed a path management scheme based on the overhearing feature of the wireless medium [15]. They include the location information of the sink in the data packets. Therefore, the data packet from the agent has the up-to-date location of the sink. This updated location information is gradually propagated to the successive path nodes; consequently, the path would be smoothly modified toward the current location of the sink. Jain et al. proposed a query-driven routing protocol for WSNs with mobile sinks [16]. They exploit a virtual structure, called wheel, to deliver query and data between sensors and a mobile sink. Due to the fact that the virtual wheel is a closed chain of sensors having one-node width, the location of a mobile sink can be thoroughly traced. The path management schemes support sink mobility without flooding so that considerable energy can be saved. However, in multipath routing, the existing schemes do not guarantee the path disjointness since there is no established path but a location-based connection between the sink and the agent. If the multipath is jointed each other into a single path, it causes loss of not only reliability or bandwidth but also service consistency. For some parts made up of multipath, using an alternative path or increasing bandwidth under the trust of them causes service disruption. This problem should be resolved through a multipath management method guaranteeing disjointness.

Recently, researchers have proposed path management mechanisms for multipath routing. Wang et al. proposed a route adjustment scheme for supporting sink mobility [17]. In this, the sensing field is divided into equal-sized clusters and each cluster head acts as an agent. Therefore, a mobile sink updates its location to the closest cluster head so that the data from sensors are delivered through the cluster head. Aswale and Ghorpade proposed a multipath routing protocol based on the triangle link quality metric for enhancing the traditional link quality estimation method using link quality indicator or packet reception ratio [18]. The proposed protocol includes the path maintenance phase in which the multipath are recovered using route alert and route discovery messages to exclude the exhausted or low-quality path nodes.Sreeram et al. suggested the enhanced route recovery mechanism where the broken link is restored without route alert messages [19]. The path management mechanisms maintain the end-to-end connection. However, they suffer from the route discovery delay when the paths are broken due to the continuous movement of the mobile sink. The initial idea of MPM was presented in [20] but this primitive method is designed for a dense deployment of sensors. Therefore, it is hard to be applied when sensors are sparsely deployed or there are void areas. On the other hand, MPM is dealing with the multihop path management scenarios so that more general environment can be covered.

3. Proposed Scheme

3.1. Preliminary. To elaborate on MPM, we defined the network model as follows. The sensor nodes are uniformly deployed on a square sensor field. There are a mobile sink and a source node that is randomly selected among sensor nodes. A sensor node is aware of the location of itself and neighbors through a localization method and beacon exchanges. At the initial stage of the network, the multipath is constructed between the source and the sink. Let P_N be a path consisting of a set of nodes having same unique path ID N, and we call the nodes the path nodes. The PN_N^H is a path node of P_N with hop distance H from the source. That is, $P_i = \{PN_i^1, PN_i^2, \cdots, PN_i^j\}$ where i is the path ID and j is the length of the path. Among the path nodes, the agent is the node having the largest hop distance (i.e., PN_i^j = the agent of P_i). That is, the agent is a neighbor of the sink.

There are two path management functions: path shortening and path extending. First, path shortening is conducted when the sink meets a path node. Due to the fact that any path node was closer to the source than the agent of the corresponding path, altering the last-hop connection of the path from the agent to the discovered path node provides higher reliability and lower energy consumption. In this case, the discovered path node becomes a new agent, and the subpath from the new agent to the old agent is released. Second, path extending is conducted in case the sink moves out of the transmission range of an agent but there is no path node that has the same path ID with the agent within the sink's transmission range. In this case, the sink has to elect a new agent among neighboring nodes and to establish the connection between the new agent and the old agent for restoring the path. Figure 2 shows the two path management cases according to the above classification. It is worth noting that path management could be applied in a multihop manner as shown in Figures 2(c) and 2(d). To check the existence of a path node or the absence of an agent, the mobile sink should periodically update its neighbor list. We will discuss the impact of the neighbor list update at the later of this section.

The mobile sink manages the status of the agent using the agent table. Generally, a mobile sink is referred to as a powerful node having more energy and better processing power than sensor nodes. To prolong the network lifetime, we actively use the capacity of a mobile sink. In other words, MPM imposes the overheads of multipath management to the mobile sink to reduce the energy consumption of the sensor nodes. In MPM, a mobile sink is responsible for state management, condition check, and process initialization.

A mobile sink supervises the multipath management through the agent table. The number of agent table entries is equal to the number of paths. The agent table includes path ID, agent location, hop distance, path priority, and path state. Path ID is uniquely assigned to each path when the multipath is constructed. The agent location indicates the geographical coordinate of the current agent having the path ID. Hop distance contains the hop count from the source to the agent. To help path management, the packet header of MPM includes path ID, sender's location, and sender's hop distance from the source. A mobile sink can understand and manage the state of each path by extracting the header of the received packet from the sensors.

Path priority and path state are assigned by a sink and specially used for path extending. Most existing path discovery schemes exploit the shortest path or the most reliable path (i.e., a path consisting of most reliable nodes/links) as a primary path. The other paths are subpaths or alternative paths, constructed beside the primary path. In MPM, the priority of the primary path is 0. And the priorities of left- and right-side paths of the primary path are assigned alternately. As a result, the odd priority paths are left side, and the even priority paths are the right side of the primary path. Which side the odd- or even-priority paths reside is of no importance but we designate that for the clear explanation. The path priority values are initialized whenever the multipath is reconstructed. The values of the path state field are 'valid' when the connection from the sink to the corresponding agent is available. Figure 3 shows an example of the constructed multipath and the resulting agent table.

3.2. *Path Management Algorithm.*

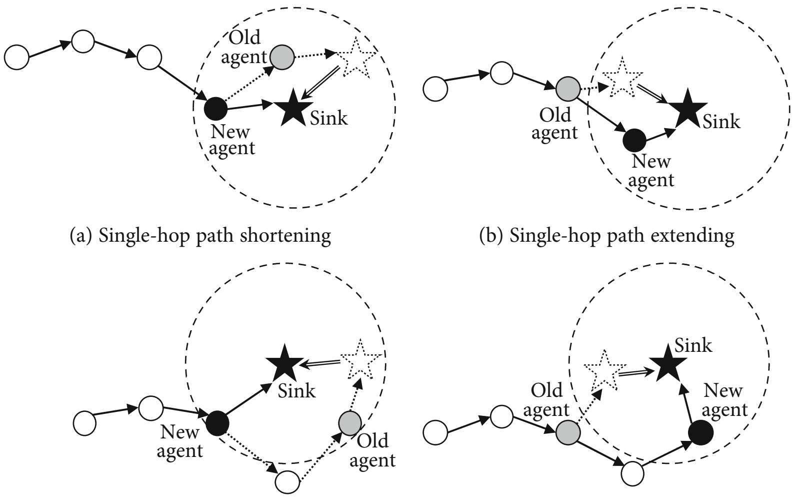
The mobile sink continuously checks the list of the neighboring sensors since the change of the neighbor list might require path management. There are two significant changes in the neighbor list: the addition of a path node and the elimination of an agent. The addition of a path node means the opportunity of shortening the path because the agent has the largest hop distance toward the source. On the other hand, the elimination of an agent means that a path is not available, and consequently, the routing path between the sink and the agent has to be reestablished. Therefore, the sink performs the path shortening or path extending process according to the presence of neighbor nodes and the condition of the agent table.

The addition of a path node and elimination of an agent may occur simultaneously according to the length of the neighbor list update interval and the positions of the path nodes. The concurrence of the addition and elimination events can happen on different paths or on the same path. In the different-path case, the sink has to deal with the elimination event first since recovering a path is related to the 'availability' of the path whereas shortening the length of a path is related to the 'efficiency' of the path. In the same-path case, on the other hand, the sink simply discards the eliminated agent because the availability of the path corresponding to the agent would obviously be restored by electing the added path node as the new agent.

3.2.1. *Path Shortening.*

If the addition of any path node has been identified, the sink conducts the path shortening process. First, the sink notifies the added path node that it has become the new agent of the path to establish the last-hop connection of the path. Then, the sink updates the agent table with the information of the new agent. The sink finds the agent table entry having the same path ID with the new agent and updates the agent location and hop distance fields. The path state field does not change since the connection between the sink and the new agent has been established.

The path shortening process can also be performed by a data packet from a path node. In other words, a mobile sink can identify the existence of the new neighboring path node before updating the neighbor list by overhearing a data packet.



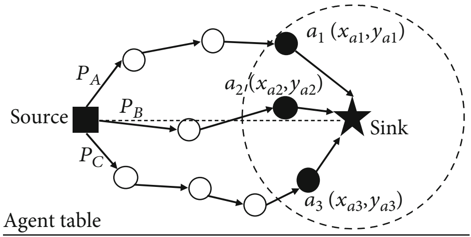
(a) Single-hop path shortening

(b) Single-hop path extending

(c) Multihop path shortening

(d) Multihop path extending

Figure 2: Path shortening and path extending.



Agent table

Path ID	Agent location	Hop distance	Path priority	Path state
A	x_{a1}, y_{a1}	3	2	Valid
B	x_{a2}, y_{a2}	2	0	Valid
C	x_{a3}, y_{a3}	4	1	Valid

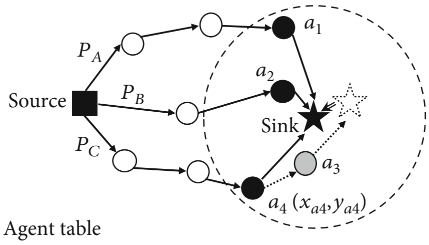
Figure 3: Path construction and agent table.

When a sink receives a data packet, the sink extracts the packet header and finds an agent table entry having the same path ID of the header. Then, the sink compares the hop distance of the header and the one of the agent table entry. If the hop count of the header is smaller, the sink conducts the abovementioned path shortening process with the sender of the data packet as the new agent. After that, the sink adds the new agent in the neighbor list so that the addition of the agent at the next neighbor list update can be ignored.

After the agent table updated, the sink sends the path release message to the new agent. The new agent forwards the message along the old path so that old path nodes including the old agent remove their path states. Figure 4 shows an example of path shortening. In the figure, nodes a_3 and a_4 are the old agent and the new agent, respectively. After node a_4 has become the new agent, the agent table entry about path C is updated with the information of new agent.

3.2.2. Path Extending. On the other hand, if the elimination of an agent was detected, path extending is performed. The sink finds the agent table entry corresponding to the eliminated agent and the changes the path state to 'invalid' which means the last-hop connection is lost. Technically, path extension is the route discovery from the sink to the old agent. The criterion for selecting the next hop node is similar to the void handling techniques. The sink draws a virtual line between the location of itself and the location of the old agent which comes from the agent table and searches the next-hop node clockwise or counterclockwise from the virtual line. The searching direction depends on path priority. That is, the sink searches clockwise for even priority paths and counterclockwise for odd priority paths. In the case of the primary path (i.e. the priority is 0), the direction faces the farther agents of the second-order paths (i.e., the priority is 1 or 2). During the searching process, the firstly discovered node is selected as the new agent whereas the agents of other paths are skipped. The purpose of the strategies for selecting a new agent is to promote the path disjointness. Due to the fact that an extended path consists of the consecutively elected agents, the new agent should be located where there is no impact on other paths. Figure 5(a) shows an example of path extending. The sink searches counterclockwise from the virtual line between the location of itself and the location of a_3 (i.e., the old agent) because the priority of the path C is 1. Due to the searching direction, node n_1 is excluded and it might be the new agent of the primary path. Finally, node n_2, which is the firstly discovered candidate, is elected as the new agent of path C. In the case of the absence of node n_2, the path extension process can establish a multihop route from the sink to the old agent as shown in Figure 5(b). The sink selects node n_3 as the new agent and the searching process continues until the route reaches the old agent. During the searching process, the virtual line toward the old agent and searching direction are maintained from the sink.

The area for the candidates is restricted a half-side with the straight line between the sink and old agent as a center. Which side is the area for the candidates is decided by whether the path priority is odd or even. For example, as shown in Figure 5, the path priority of path C is 1, an odd



Agent table

Path ID	Agent location	Hop distance	Path priority	Path state
A	x_{a1}, y_{a1}	3	2	Valid
B	x_{a2}, y_{a2}	2	0	Valid
C	x_{a4}, y_{a4}	3	1	Valid

Figure 4: Agent table update for path shortening.

number. At the initial stage of path extension process, the candidate for the new agent is 4 (node a_2, n_1, n_2, and n_3). Node a_2, which is already a path node, is excluded. Node n_1 is also excluded because it is on the right side. Between node n_2 and n_3, n_3 is finally elected as a new agent since node n_3 is further from the straight line between the sink and node a_3, an old agent. In case that the path order is 0 (primary path), the new agent can be elected from both areas. Among the nodes in the area, the highest priority candidate is the node that is closest to the sink for reducing the frequency of the path extending process. After the new agent is elected, the agent table is updated.

3.3. Discussion

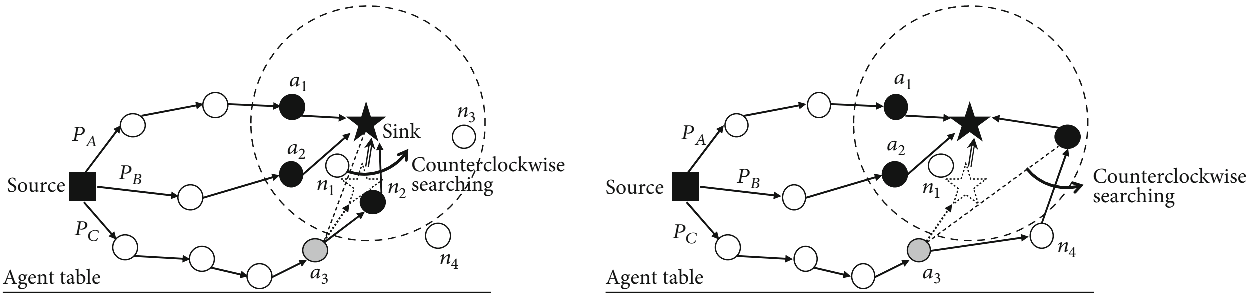
3.3.1. Path Reconstruction. After a mobile sink moves a long distance, overly long extended paths or inefficiently extended paths (e.g., zigzag) might be constructed. In this case, the end-to-end multipath should be reconstructed rather than keeping an immoderate path extension. There are two methods that can be adapted: cost-based and periodic reconstruction. The cost-based reconstruction is performed when the cost of conducting reconstruction is smaller than the cost of keeping extended paths. Therefore, defining the cost is not only the nominally important factor but also a key factor influencing the actual performance. Definitely, both costs (i.e., for conducting reconstruction and keeping extended paths) have to include not only the path construction cost but also future data delivery costs. This approach should be designed complementary with the path construction scheme and the purpose of the application. Moreover, it is seriously influenced by external factors, such as the sink's mobility pattern, data generation pattern, and deployment of sensors. Although the cost-based reconstruction seems very efficient, it is generally unavailable due to the uncertainty of those external factors. For example, to calculate the cost for delivering a data packet from the source to a mobile sink, the exact values of the external factors should be available such as the sending time at the source, receiving time at the sink, location of the sink at the receiving time, and hop count of each extended/shortened path. That is, the cost-based reconstruction can be adopted by the application where the external factors are strictly controlled. Therefore, the cost-based reconstruction may lead to inefficient end-to-end multipath reconstruction in general applications where those external factors cannot be strictly controlled. On the other hand, in the periodic reconstruction, the multipath is reconstructed every certain period of time or the distance that the sink moved after the multipath is reconstructed. This is very simple because it needs only to check time or distance. In addition, it can provide steady performance against the mobility pattern of a mobile sink and the data generation patterns of sensors. In this paper, we use the periodic approach for our computational simulation for providing relatively steady operation thereby clearly comparing the path management performance.

3.3.2. Neighbor List Update. In MPM, the mobile sink periodically updates its neighbor node list for identifying the opportunity of path shortening and the necessity of path extension. For neighbor list update, the sink broadcasts a beacon and the neighbor nodes respond to the beacon. During the beaconing process, the ordinary nodes just notify its existence to the sink whereas a path node includes the path ID and its hop distance in the response message for agent table update. The neighbor list update interval is an important tunable factor of MPM since it might influence the energy efficiency and data throughput. On the one hand, the frequent neighbor list update provides agile path management at the expense of the energy consumption of the sink and neighbor nodes for beaconing. On the other hand, the infrequent neighbor list update achieves energy savings but might suffer from the disconnection from the moment that the sink moves out of the agent's radio range to the next beaconing. In this context, the neighbor list update interval should be carefully configured with consideration of the mobility pattern of the mobile sink and the data packet generation patterns of sensor nodes.

4. Performance Evaluation

In this chapter, we present our simulation results to evaluate the performance of the proposed scheme. We implement the MPM in MATLAB. We also implement the multipath reconstruction (MPR) and single path management (SPM) schemes as the comparison group. In MPR, the end-to-end multipath between a source and a mobile sink is reconstructed whenever the sink moves outside the radio range of the agent. In SPM, new agents are continuously elected for constructing a single path along the moving path of the mobile sink after the multipath are constructed. In common, three schemes exploit the ideal multipath discovery method to minimize the effect of the multipath reconstruction overhead. In the ideal multipath discovery, a mobile sink is aware of the locations of every node. The sink constructs an ideal multipath toward the source without any signaling. The sink sends multiple copies of a message through all paths, and the source replies multiple copies of the ACK message along the reverse routes.

The default simulation setting is as follows. 1000 sensor nodes are uniformly deployed in the 500 m × 500 m square field. The radio range of all nodes is 25 m. Each node



Agent table

Path ID	Agent location	Hop distance	Path priority	Path state
A	x_{a1}, y_{a1}	3	2	Valid
B	x_{a2}, y_{a2}	2	0	Valid
C	x_{n2}, y_{n2}	5	1	Valid

Agent table

Path ID	Agent location	Hop distance	Path priority	Path state
A	x_{a1}, y_{a1}	3	2	Valid
B	x_{a2}, y_{a2}	2	0	Valid
C	x_{n3}, y_{n3}	6	1	Valid

Figure 5: Agent table update for path extending.

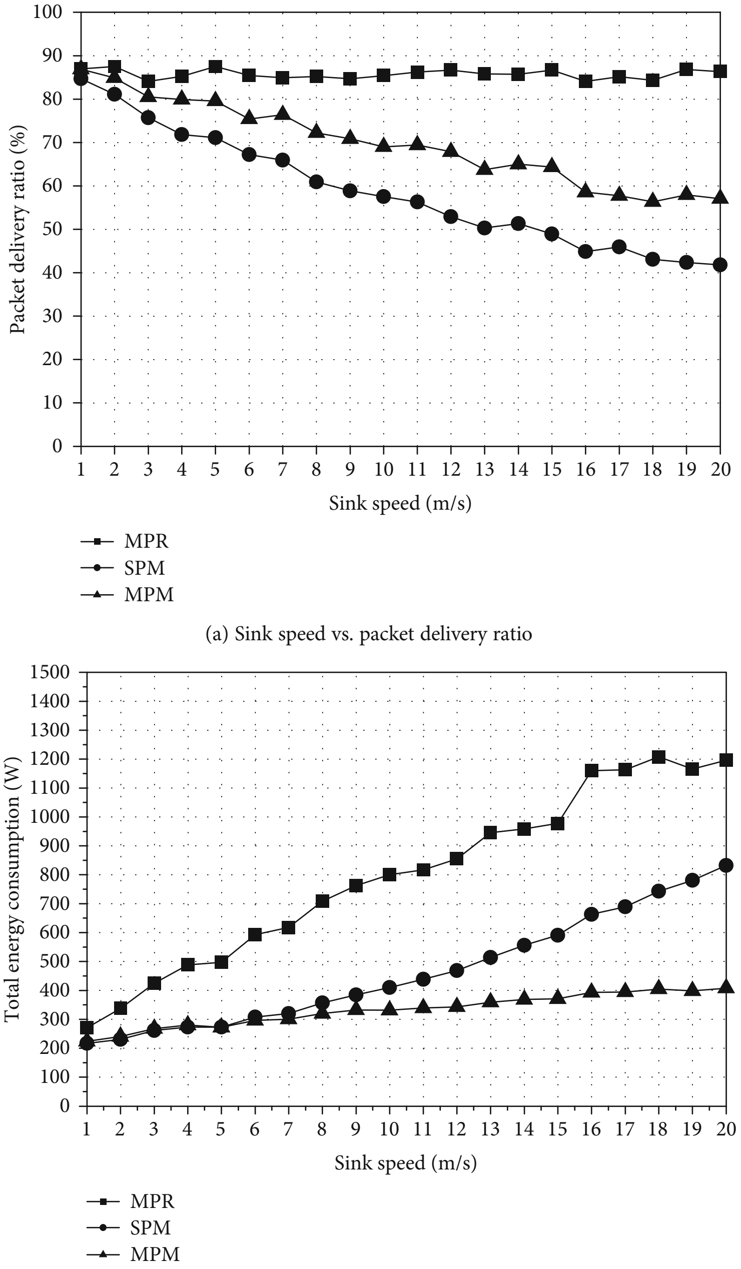
consumes 20 mW and 15 mW for transmitting and receiving a packet, respectively. There are a source and a mobile sink. Since the data delivery performance is significantly affected by the hop count, we set the initial distance between the source and the mobile sink about 350 m. Under the condition, the source is arbitrary elected among all sensor nodes, and similarly, the initial location of the sink is designated randomly. The source generates a data packet every 5 seconds. The mobility of the sink follows the random waypoint mobility model [21] and the speed of the sink is 5 m/s. The mobile sink updates the neighbor list every 0.5 seconds. After the neighbor list update, the multipath reconstruction (in case of MPR) or path management (in case of SPM and MPM) might be conducted. The number of paths is 3, and they are reconstructed every 50 seconds (periodic multipath reconstruction). The packet delivery success ratio of all links is 95% and the retransmission strategy is excluded. Each simulation lasts 1000 seconds and the results are the average of 20 simulations for complying 95% confidence interval.

We choose two metrics: the packet delivery ratio and total energy consumption. The packet delivery ratio is the proportion of the number of received data packets of the sink to the number of sent data packets of the source. The total energy consumption is the sum of the consumed energy of all nodes including the sink. These metrics are the typical pros and cons of multipath routing. As a guiding mention, our all simulation results say that the MPM mitigates serious degradation of packet delivery success ratio without considerable energy consumption.

Figure 6(a) shows the packet delivery ratio according to the speed of the mobile sink. We vary the sink speed from 1 m/s to 20 m/s for including various types of mobile sinks ranging from handheld devices to connected vehicles. The MPR shows the highest packet delivery ratio because it tends to construct the shortest multipath than others. Since MPR reconstructs an end-to-end multipath whenever the sink moves outside the agent, the results are relatively steady. On the other hand, the packet delivery ratio of SPM decreases as the sink speed increases. This is because the portion of the single path increases as the sink speed increases in SPM. In the case of the MPM, the packet delivery ratio is higher than SPM but lower than MPR. This is because the extended end-to-end paths of MPM are longer than the reconstructed paths of MPR although the number of multiple paths is the same. Figure 6(b) shows the total energy consumption according to the speed of the mobile sink. The noticeable tendency is that the total energy consumption of MPR increases rapidly as the sink speed increases. This is because the reconstruction frequency of MPR becomes higher as the sink speed increases. The total energy consumptions of SPM and MPM increased slowly as the speed of the mobile sink increased for constructing extended paths while MPM consumes more energy than SPM for extending multiple paths.

Figure 7(a) shows the packet delivery ratio according to the number of paths. We vary the number of paths from 2 to 5. All schemes show the increasing tendency because the path redundancy increases as the number of paths increases. The packet delivery ratios of MPR and MPM are higher than SPM but the one of MPM is slightly lower than MPR because the extended paths are longer than the reconstructed paths. The packet delivery ratio of SPM is lowest due to the extension of the single path. The difference between the SPM and others increases as the number of paths increases since other schemes use reconstructed or extended multipath while SPM uses the extended single path constantly. Figure 7(b) shows the total energy consumption according to the number of paths. The total energy consumption of the MPR increases sharply because the reconstruction overhead increases as the number of paths increases. On the other hand, the total energy consumptions of SPM and MPM increase gradually and almost the same. Although MPM manages more paths than SPM, the path shortening process makes the overall path management efficient.

Figure 8(a) shows the packet delivery ratio according to the packet generation interval varying from 1 to 10 seconds. As the packet generation interval increases, the source sends the decreased number of data packets but the routing paths are not affected. Therefore, the packet delivery ratios of all schemes are almost not changed. Figure 8(b) shows the total energy consumption according to the packet generation



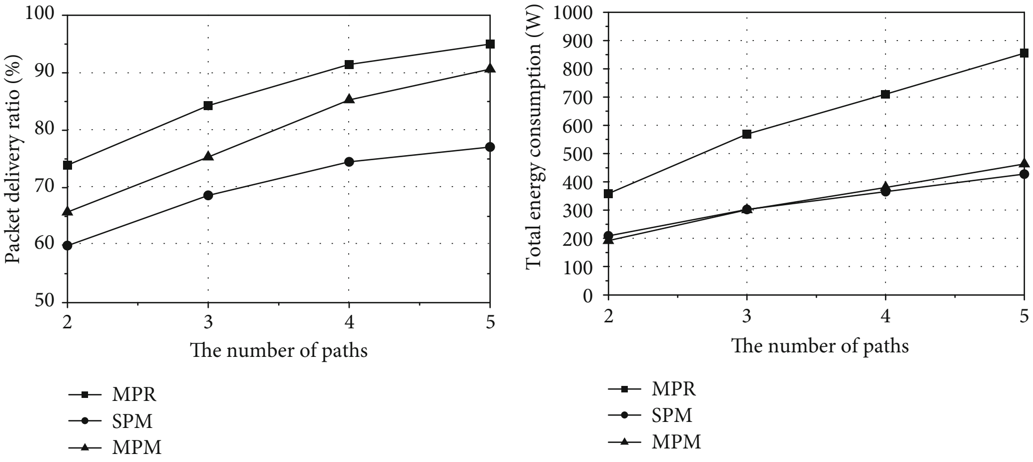
(a) Sink speed vs. packet delivery ratio

(b) Sink speed vs. total energy consumption

Figure 6: Simulation result according to the speed of the mobile sink.

interval. The total energy consumptions of all schemes decreased exponentially since the energy consumption for data delivery decreased as the data generation interval decreased. In addition, the total energy consumption of all schemes converged into a specific value which implies the cost for path management. Specifically, the convergence value of MPR is much higher than others which means MPR consumes much more energy for multipath reconstruction. Meanwhile, SPM and MPM consume almost similar energy for path management.

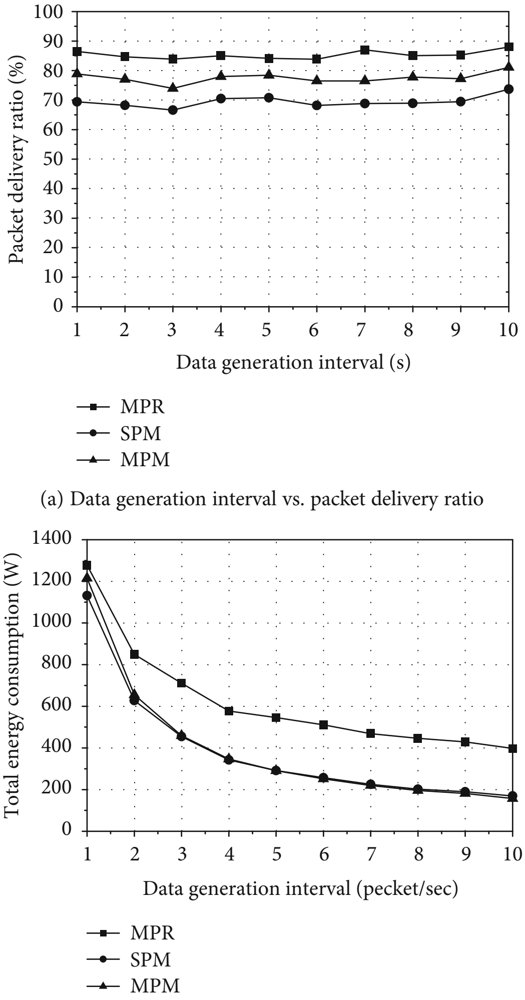
Figure 9(a) shows the packet delivery ratio according to the neighbor list update interval. The neighbor list update interval implies how often the mobile sink recognizes the disconnection with the agent. In other words, the disconnection duration increases as the neighbor list update interval increases. Therefore, the packet delivery ratio of all schemes decreased as the neighbor list update interval increased. The distinct feature is the peak points at some neighbor list update interval values, that is, 2.5 and 5seconds. This result is related to the packet generation interval which is designated by 5 seconds in the default simulation setting. In other words, if the value of the neighbor list update interval is an aliquot of the packet generation interval, the duration of the disconnection can be ignored. Figure 9(b) shows the total energy consumption according to the neighbor list update interval. In cases of the SPM and MPM, the total energy



(a) The number of paths vs. packet delivery ratio

(b) The number of paths vs. total energy consumption

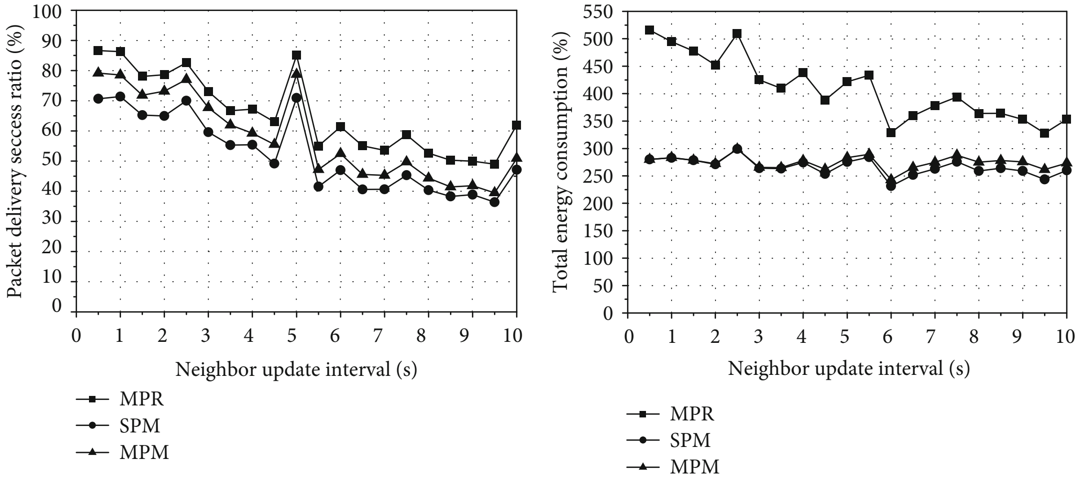
Figure 7: Simulation results according to the number of paths.



(a) Data generation interval vs. packet delivery ratio

(b) Data generation interval vs. total energy consumption

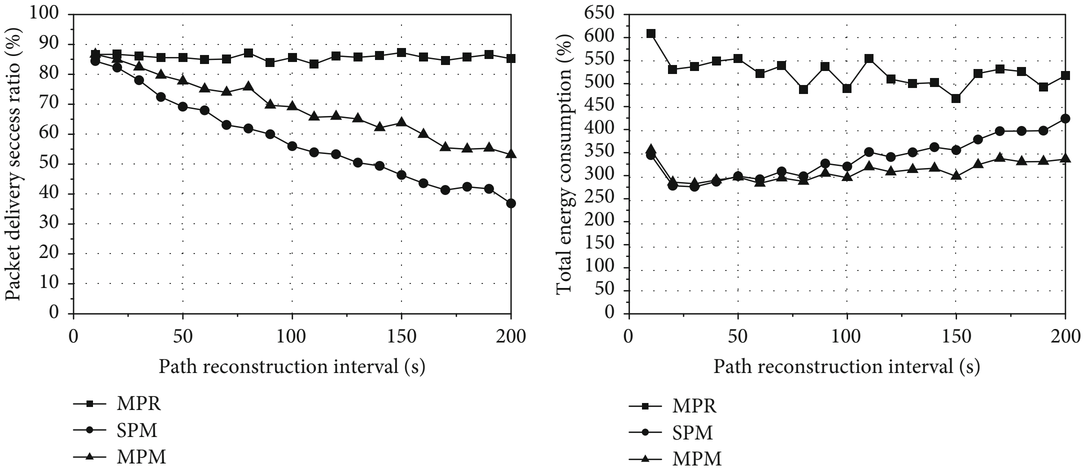
Figure 8: Simulation results according to the data generation interval.



(a) Neighbor update interval vs. packet delivery ratio

(b) Neighbor update interval vs. total energy consumption

Figure 9: Simulation results according to the neighbor update interval.



(a) Path reconstruction interval vs. packet delivery ratio

(b) Path reconstruction interval vs. total energy consumption

Figure 10: Simulation results according to the path reconstruction interval.

consumptions were almost steady which means the cost for neighbor list update and the local path management is very small compared to the end-to-end path reconstruction and data delivery. On the other hand, the total energy consumption of MPR decreased slowly as the neighbor list update interval increased. In the case of MPR, the end-to-end multipath reconstruction interval increases as the neighbor list update interval increases.

Figure 10(a) shows the packet delivery ratio according to the end-to-end multipath reconstruction interval which is the key factor in the periodic path reconstruction approach. The packet delivery ratio of MPR was almost steady because the end-to-end multipath reconstruction is performed whenever the sink leaves the agent regardless of the reconstruction interval. In cases of SPM and MPM, on the other hand, the packet delivery ratio decreased slowly whereas the MPM achieved a higher packet delivery ratio. It means that MPM manages multiple extended paths effectively even the reconstruction interval is significantly long. Figure 10(b) shows the total energy consumption according to the multipath reconstruction interval. In the case of MPR, the total energy consumption was almost not changed because the multipath reconstruction is performed not only periodically but also actively at the neighbor list update. On the other hand, SPM consumed a slightly increased amount of energy due to data delivery through the long extended routing path. In the case of MPM, the total energy consumption was almost not changed because it manages multiple routing paths very efficiently during the path reconstruction interval.

5. Conclusion

In this paper, we propose a novel multipath management scheme called MPM for supporting sink mobility in wireless sensor networks. Existing multipath routing protocols needs a significant amount of overhead for reconstructing an end-to-end multipath for supporting sink mobility. Meanwhile, the existing single path management schemes maintain the end-to-end connection without reconstruction by extending the constructed path. However, the single path management schemes cannot provide high-performance data delivery as multipath routing. To solve the problems, MPM extends multiple paths using the agent-based multipath management thereby maintaining the high-quality data delivery through multipath. Our simulation results show that MPM provides cost effectiveness compared to the existing multipath reconstruction and single path management schemes. Depending on the network parameter such as the speed of the mobile sink and the data generation interval of sensors, tuning neighbor list update interval and end-to-end multipath reconstruction interval influence the performance of MPM significantly.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

Acknowledgments

This work was supported by the Korea Institute of Science and Technology Information (KISTI).

References

[1] S. Yu, B. Zhang, C. Li et al., "Routing protocols for wireless sensor networks with mobile sinks: a survey," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 150–157, 2014.

[2] E. Hamida and G. Chelius, "Strategies for data dissemination to mobile sinks in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 6, pp. 31–37, 2008.

[3] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, 2014.

[4] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's INTRAnet of things to a future INTERnet of things: a wireless-and mobility-related view," *IEEE Wireless Communications*, vol. 17, no. 6, pp. 44–51, 2010.

[5] M. Radi, B. Dezfouli, K. A. Bakar, and M. Lee, "Multipath routing in wireless sensor networks: survey and research challenges," *Sensors*, vol. 12, no. 1, pp. 650–685, 2012.

[6] J. Ben-Othman and B. Yahya, "Energy efficient and QoS based routing protocol for wireless sensor networks," *Journal of Parallel and Distributed Computing*, vol. 70, no. 8, pp. 849–857, 2010.

[7] W. Lou and Y. Kwon, "H-SPREAD: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1320–1330, 2006.

[8] G. Wang, T. Wang, W. Jia, M. Guo, and J. Li, "Adaptive location updates for mobile sinks in wireless sensor networks," *The Journal of Supercomputing*, vol. 47, no. 2, pp. 127–145, 2009.

[9] J. Yim, J. Bang, Y. Nam, Y. Shin, and E. Lee, "Efficient Multipath Routing Protocol Against Path Failures in Wireless Sensor Networks," in *12th IFIP Wireless and Mobile Networking Conference (WMNC),*, vol. 2019, pp. 136–140, Paris, France, 2019.

[10] P. Mohanty and M. R. Kabat, "Energy efficient reliable multipath data transmission in WSN for healthcare application," *International Journal of Wireless Information Networks*, vol. 23, no. 2, pp. 162–172, 2016.

[11] M. Z. Hasan, H. Al-Rizzo, and F. Al-Turjman, "A survey on multipath routing protocols for QoS assurances in real-time wireless multimedia sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1424–1456, 2017.

[12] K. Jaiswal and V. Anand, "EOMR: an energy-efficient optimal multi-path routing protocol to improve QoS in wireless sensor network for IoT applications," *Wireless Personal Communications*, vol. 111, no. 4, pp. 2493–2515, 2020.

[13] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient broadcast: a robust data delivery protocol for large scale sensor networks," *Wireless Networks*, vol. 11, no. 3, pp. 285–298, 2005.

[14] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "TTDD: two-tier data dissemination in large-scale wireless sensor networks," *Wireless Networks*, vol. 11, no. 1-2, pp. 161–175, 2005.

[15] F. Yu, S. Park, E. Lee, and S.-H. Kim, "Elastic routing: a novel geographic routing for mobile sinks in wireless sensor networks," *IET Comuunications*, vol. 4, no. 6, pp. 716–727, 2010.

[16] S. Jain, K. K. Pattanaik, and A. Shukla, "QWRP: query-driven virtual wheel based routing protocol for wireless sensor networks with mobile sink," *Journal of Network and Computer Applications*, vol. 147, p. 102430, 2019.

[17] J. Wang, J. Cao, S. Ji, and J. H. Park, "Energy-efficient cluster-based dynamic routes adjustment approach for wireless sensor networks with mobile sinks," *The Journal of Supercomputing*, vol. 73, no. 7, pp. 3277–3290, 2017.

[18] S. Aswale and V. R. Ghorpade, "Geographic multipath routing based on triangle link quality metric with minimum inter-path interference for wireless multimedia sensor networks," *Journal of King Saud University-Computer and Information Sciences*, 2018.

[19] K. Sreeram, A. Unnisa, V. Poornima, and S. Chaudhari, "QoS aware multi-constrained node disjoint multipath routing for wireless sensor networks," in *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pp. 382–385, Coimbatore, India, 2019.

[20] C. Kim, Y. Yim, T. Yang, S. Kim, and S.-H. Kim, "Multipath management scheme for supporting sink mobility in wireless sensor networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 2017, pp. 1–6, San Francisco, CA, 2017.

[21] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 257–269, 2003.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8816655, 7 pages
https://doi.org/10.1155/2020/8816655

Research Article

New Signal Constellation Pairs for the ZTM-OFDM-IM System

Shuang Li[1] **and Seog Geun Kang**[2]

[1] *Department of Electrical and Electronic Engineering, Gyeongsang National University, Jinju, Republic of Korea*
[2] *Department of Semiconductor Engineering and the Engineering Research Institute, Gyeongsang National University, Jinju, Republic of Korea*

Correspondence should be addressed to Seog Geun Kang; sgkang@gnu.ac.kr

Received 23 June 2020; Revised 2 July 2020; Accepted 11 July 2020; Published 1 August 2020

Academic Editor: KI-IL Kim

In this paper, we investigate new signal constellation pairs for mapping active subcarriers of the zero-padded trimode orthogonal frequency division multiplexing with index modulation (ZTM-OFDM-IM) systems. In the presented system, one of a constellation pair is the same as the one used in the previous work, and the other is a constellation larger than the one used in the previous work. It increases the minimum Euclidean distance between the subblocks of the ZTM-OFDM-IM system with new constellation pairs under the constraint of the same spectral efficiency. Computer simulation in AWGN and frequency-selective fading channels shows that the new ZTM-OFDM-IM system has a much lower bit error rate than OFDM-IM and dual-mode OFDM-IM and slightly outperforms the system with conventional constellation pairs. Since the proposed constellation pairs prove the error performance improvement of the system, it is considered that a further study on generalized design of the constellation pair for the ZTM-OFDM-IM system is necessary in the future.

1. Introduction

Due to its large capacity, high spectral efficiency, and robustness against frequency-selective fading [1], orthogonal frequency division multiplexing (OFDM), a multicarrier modulation technique, has been widely adopted in various wireless communication systems and standards such as wireless fidelity (Wi-Fi) [2], long-term evolution (LTE) [3], and world interoperability for microwave access (WiMAX) [4] for next-generation wireless communications. It is also an attractive advantage of an OFDM system for wireless communications to effectively eliminate intersymbol interference (ISI) by inserting a cyclic prefix (CP) between successive symbols [5].

To accommodate the requirements for high spectral and energy efficiencies, a lot of studies on index modulation (IM) have been carried out. In the IM scheme, a part of indexed resources are activated and used for data transmission [6, 7]. A subcarrier-index modulated (SIM) OFDM [8] and an enhanced SIM-OFDM (ESIM-OFDM) scheme [9] introduce the concept of IM into the frequency domain to achieve diversity gain. However, the spectral efficiency of both systems is very low compared to the conventional OFDM scheme. To mitigate this problem, OFDM with IM (OFDM-IM) has been proposed to flexibly transmit IM bits by controlling the index of active subcarriers [10, 11]. In the OFDM-IM system, information bits are divided into symbol bits and index bits. The former are mapped to the activated subcarriers in the same way as the traditional OFDM modulation, and the latter indicate the pattern of the activated subcarriers in the OFDM subblock. Hence, the energy efficiency of the scheme can be significantly improved, while the spectral efficiency is limited [12]. To improve the spectral efficiency of the OFDM-IM system, dual-mode OFDM-IM (DM-OFDM-IM) has been proposed, where all subcarriers of a subblock are activated and mapped with two different constellations [13]. As the frequency spectrum can be fully utilized while maintaining the characteristics of the IM scheme, the DM-OFDM-IM generally outperforms the OFDM-IM. Though the DM-OFDM-IM scheme has a little loss of energy efficiency, it can enhance the data rate more than the OFDM-IM.

As a compromise between spectral efficiency and energy efficiency, the zero-padded trimode OFDM-IM (ZTM-OFDM-IM) system has been proposed in [14]. Here, the subcarriers are divided into several disjoint subblocks. A part of the subcarriers in each subblock is modulated by two distinguished constellations, and the other subcarriers are left blank, so that energy efficiency of the system can be increased. To avoid decreasing spectral efficiency, a larger constellation is usually exploited to map the active subcarriers.

In this paper, we introduce new constellation pairs to improve error performance of the ZTM-OFDM-IM system. Here, one of the two mappers used to map active subcarriers uses the same signal constellation as the one used in the previous work, while the other uses a larger constellation than the one used in [14]. This increases the minimum Euclidean distance (MED) between the subblocks of the ZTM-OFDM-IM system with new constellation pairs under the constraints of the same spectral efficiency. The proposed constellation pairs demonstrate the possibility of improving error performance of the system.

The rest of this paper is organized as follows. The transceiver models of the OFDM systems with the IM scheme are briefly introduced in Section II. In Section III, the new pairs of constellations are discussed in detail. Performance of the proposed ZTM-OFDM-IM system is analyzed in Section IV. Finally, some conclusions are provided in Section V.

2. The OFDM Systems with Index Modulation

There are three typical index modulation schemes for the OFDM system. The OFDM-IM and DM-OFDM-IM can be considered as special cases of the ZTM-OFDM-IM system. The typical transmitter structure of the ZTM-OFDM-IM is shown in Figure 1, where S/P and P/S represent serial-to-parallel and parallel-to-serial converters, respectively. IFFT is the N-point inverse fast Fourier transform. For an OFDM signal with N subcarriers $\mathbf{X} = [X(1), X(2), \cdots, X(N)]^T$ in the frequency domain, all subcarriers are divided into G disjoint subblocks to represent the signal as $\mathbf{X} = [\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(G)}]^T$, where $\mathbf{X}^{(g)} = [X_1^{(g)}, X_2^{(g)}, \cdots, X_n^{(g)}]^T$, $g = 1, 2, \cdots, G$, is the subcarriers in the gth subblock and $n = N/G$ is the number of subcarriers in a subblock. Thus, the OFDM signal can be rewritten as

$$\mathbf{X} = \left[X_1^{(1)}, X_2^{(1)}, \cdots, X_n^{(1)}, X_1^{(2)}, X_2^{(2)}, \cdots, X_n^{(2)}, \cdots, X_1^{(G)}, X_2^{(G)}, \cdots, X_n^{(G)}\right]^T. \tag{1}$$

The m input information bits are divided into G parallel data streams of $p = m/G$ bits. $p = p_M + p_I$ information bits for a group are divided into p_I index bits and p_M symbol bits, which are fed to the index selector and a pair of constellations for mappers, respectively. p_A from the p_M symbol bits are for signal mapping using the mapper A, while the remaining $p_B = p_M - p_A$ bits are for the mapper B. Since the constellations for signal mapping should be distinguishable, the signal points of the constellation A and those of the constellation B are not overlapped with each other. To prevent the spectral efficiency of the system from being reduced, the constellation size M_B for the mapper B is generally larger than M_A, where the constellation size implies the number of elements in the constellation.

According to the p_I index bits, the index selector determines one of 2^{p_I} index patterns expressed as

$$\mathbf{I} = \left[\mathbf{I}^{(1)}, \mathbf{I}^{(2)}, \cdots, \mathbf{I}^{(2^{p_I})}\right]^T, \tag{2}$$

where the ith pattern $\mathbf{I}^{(i)}$ indirectly transmits p_I bits given as

$$\boldsymbol{I}^{(i)} = \left[I^{(i)}(1), I^{(i)}(2), \cdots, I^{(i)}(p_I)\right], \tag{3}$$

for $i = 1, 2, \cdots, 2^{p_I}$. With the fixed index pattern, there are 2^{p_M} possible transmission signals of

$$\mathbf{S}_{\mathbf{I}^{(i)}} = \left[\mathbf{S}_{\mathbf{I}^{(i)}}^{(1)}, \mathbf{S}_{\mathbf{I}^{(i)}}^{(2)}, \cdots, \mathbf{S}_{I^{(i)}}^{(2^{p_M})}\right]^T. \tag{4}$$

Each $\mathbf{S}_{\mathbf{I}^{(i)}}^{(j)} \subset \mathbf{S}_{\mathbf{I}^{(i)}}$ composed of n subcarrier signals can be represented as

$$\mathbf{S}_{\mathbf{I}^{(i)}}^{(j)} = \left[\, S_{\mathbf{I}^{(i)}}^{(j)}(1) \quad S_{\mathbf{I}^{(i)}}^{(j)}(2) \quad \cdots \quad S_{\mathbf{I}^{(i)}}^{(j)}(n) \,\right], \tag{5}$$

for $j = 1, 2, \cdots, 2^{p_M}$. Hence, there are 2^p transmission signals for $\mathbf{X}^{(g)}$. In each subblock, k_A active subcarriers from n subcarriers are modulated by the mapper A, and the $k_B = k - k_A$ active subcarriers from the remaining $(n - k_A)$ subcarriers are modulated by the mapper B, where k is the total number of active subcarriers. Since inactive subcarriers are also included in the transmitted OFDM signal, the number of inactive subcarriers $(n - k)$ should be greater than zero. Therefore, the numbers of bits p_A and p_B modulated by mapper A and mapper B are

$$\begin{aligned} p_A &= k_A \log_2(M_A), \\ p_B &= k_B \log_2(M_B), \end{aligned} \tag{6}$$

respectively. Then, p_M symbol bits are

$$p_M = k_A \log_2(M_A) + k_B \log_2(M_B). \tag{7}$$

The index selector uses p_I of the p input bits to divide the index for a subblock into two index subsets. The number of bits needed to select an appropriate index mode is

$$p_I = \left\lfloor \log_2 \begin{pmatrix} n \\ k_A \end{pmatrix} + \log_2 \begin{pmatrix} n - k_A \\ k_B \end{pmatrix} \right\rfloor, \tag{8}$$

where $\lfloor x \rfloor$ denotes the greatest integer that is not greater than or equal to x.

Except for the subcarrier indexing method, the OFDM-IM and the DM-OFDM-IM have the same functional structures as the ZTM-OFDM-IM. In the OFDM-IM scheme, only one constellation is used to generate active subcarriers.

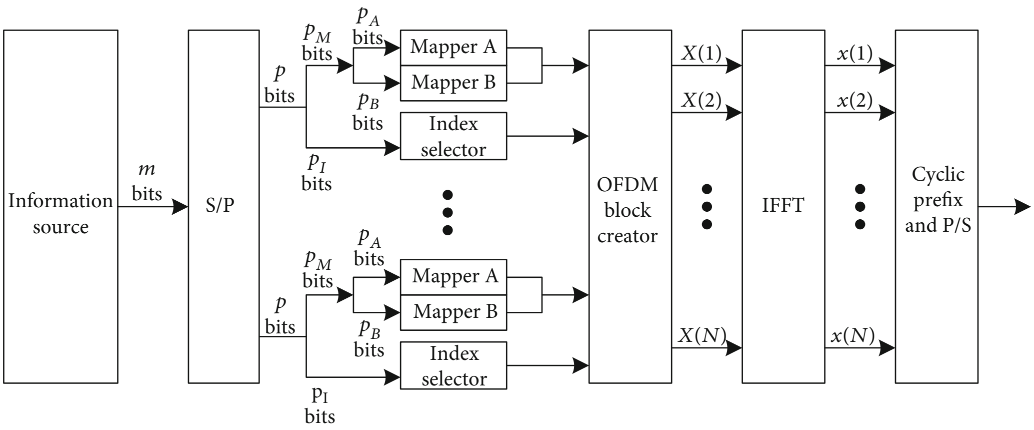


Figure 1: The transmitter structure of the ZTM-OFDM-IM system.

It is equivalent to the ZTM-OFDM-IM system with $k = k_A$, $k_B = 0$, and $M_B = 0$. Thus, it can be a special case of the ZTM-OFDM-IM with low spectral efficiency. The DM-OFDM-IM scheme exploits a pair of constellations to generate the transmitted signals. However, it is different from the ZTM-OFDM-IM in that all subcarriers are activated for signal transmission, that is, $n - k = 0$. Thus, it is considered as a ZTM-OFDM-IM system with low energy efficiency.

The OFDM system with an index modulation scheme also performs N-point IFFT to produce the signal $\mathbf{x} = [x(1), x(2), \cdots, x(N)]^T$ in the time domain. To eliminate the adverse effect of ISI caused by the delay spread of multipath fading channel, a CP of length L is added in front of the OFDM signal. Then, the spectral efficiency of the OFDM system is

$$\eta = \frac{Np}{(N+L)n}. \tag{9}$$

Suppose that the transmission environment is a frequency-selective Rayleigh fading channel of which the impulse response is $\mathbf{h} = [h(1), h(2), \cdots, h(v)]^T$ with the distribution of $\mathscr{CN}(0, 1/v)$, where v is the maximum delay spread. We assume that the length of CP should be greater than the maximum delay spread, that is, $L > v$ to completely eliminate the ISI. After removal of the CP and FFT operation in the receiver, the received OFDM signal in the frequency domain is $\mathbf{Y} = [Y(1), Y(2), \cdots, Y(N)]^T$. Then, the received signal in the gth subblock $\mathbf{Y}^{(g)} = [Y_1^{(g)}, Y_2^{(g)}, \cdots, Y_n^{(g)}]^T$ can be expressed as

$$\mathbf{Y}^{(g)} = \operatorname{diag}\left\{\mathbf{X}^{(g)}\right\}\mathbf{H}^{(g)} + \mathbf{W}^{(g)}, \tag{10}$$

where diag $\{\mathbf{X}^{(g)}\}$ is an $n \times n$ diagonal matrix of which the main diagonal elements are $\mathbf{X}^{(g)}$. $\mathbf{W}^{(g)} = [W_1^{(g)}, W_2^{(g)}, \cdots, W_n^{(g)}]^T$ represents complex additive white Gaussian noise (AWGN) of $\mathscr{CN}(0, N_0)$, that is, zero mean and variance of N_0. $\mathbf{H}^{(g)} = [H_1^{(g)}, H_2^{(g)}, \cdots, H_n^{(g)}]^T$ is a coefficient vector of the fading channel in the gth subblock.

The maximum likelihood (ML) detector is used to minimize the Euclidean distance between the estimated subblocks and the received subblocks. The estimation of the gth subblock can be determined by

$$X\wedge^{(g)} = \arg \min_{\mathbf{I}^{(i)}\subset\mathbf{I},\, \mathbf{S}^{(j)}_{\mathbf{I}^{(i)}}\subset\mathbf{S}_{\mathbf{I}^{(i)}}} \sum_{r=1}^{n} \left| Y_r^{(g)} - H_r^{(g)} S^{(j)}_{\mathbf{I}^{(i)}}(r)\right|^2, \tag{11}$$

where $S^{(j)}_{\mathbf{I}^{(i)}}(r)$ represents the jth transmission signal of the rth subcarrier with the fixed index pattern $\mathbf{I}^{(i)}$. A simple look-up table can be used to recover the information bits from the subblocks corresponding to the transmitted symbols and index patterns.

3. The Improved Constellation Pairs for ZTM-OFDM-IM

When an ML detector is exploited, error performance of the ZTM-OFDM-IM system depends largely on the MED between different OFDM subblocks. The Euclidean distance between two subblocks can be defined as

$$D\left(\mathbf{S}^{(j_1)}_{\mathbf{I}^{(i_1)}}, \mathbf{S}^{(j_2)}_{\mathbf{I}^{(i_2)}}\right) = \sqrt{\sum_{r=1}^{n} \left| S^{(j_1)}_{\mathbf{I}^{(i_1)}}(r) - S^{(j_2)}_{\mathbf{I}^{(i_2)}}(r)\right|^2}, \tag{12}$$

where $S^{(j_1)}_{\mathbf{I}^{(i_1)}}(r), S^{(j_2)}_{\mathbf{I}^{(i_2)}}(r) \in \mathbf{S}$ and $\mathbf{I}^{(i_1)}, \mathbf{I}^{(i_2)} \subset \mathbf{I}$. The minimum distance $d_{\min}$ normalized by the transmitted bit energy can be expressed as

$$d_{\min} = \min_{(i_1,j_1)\neq(i_2,j_2)} \sqrt{\frac{1}{E_b} D\left(\mathbf{S}^{(j_1)}_{\mathbf{I}^{(i_1)}}, \mathbf{S}^{(j_2)}_{\mathbf{I}^{(i_2)}}\right)}, \tag{13}$$

where $E_b = E_s(N+L)/m = (E_s(N+L))/(G(p_I + p_M))$ is the average energy per bit and E_s is the average symbol energy. Maximizing the MED between subblocks usually results in improved bit error rate (BER) performance. Therefore, when designing the constellation pair for subcarrier mapping in the ZTM-OFDM-IM, the MED between different subblocks should be greater than or equal to the MED between constellation points.

The constellation pair proposed in this paper is compared with the one in [14], where two different BPSK schemes shown in Figure 2 are adopted. The constellation A uses a pair of signal points located at $\{1,-1\}$, and the signal points of the constellation B are at $\{\sqrt{2}j,-\sqrt{2}j\}$, where $j=\sqrt{-1}$. In the proposed constellation pair shown in Figure 3, the constellation B exploits the signal points of 4-QAM, while the constellation A uses the same BPSK signal points.

Supposed that the number of subcarriers N, the length of CP L, and the maximum delay spread of the fading channel v are set to 128, 16, and 10, respectively. Assuming that the number of the disjoint group is 32, there are four subcarriers in a subblock consisting of two active subcarriers and two inactive subcarriers. In the proposed ZTM-OFDM-IM system, one of the active subcarriers is modulated by the mapper A and the other is modulated by the mapper B as shown in Figure 3. However, in the previous system [14], three of the four subcarriers are activated, two of them are modulated by the mapper A, and the other is modulated by the mapper B. The spectral efficiency of both systems is equal to 1.333 bit/s/Hz. According to (13), the normalized minimum distance of the proposed constellation pair shown in Figure 3 is $d_{\min}=2.667$, which is about 15.5% larger than $d_{\min}=2.309$ of the conventional constellation pair presented in Figure 2. Since $d_{\min}$ is one of the major factors for performance improvement, the proposed constellation pair possibly provides better BER performance. In Table 1, we compare the normalized MEDs of three index modulation-based OFDM systems with some specific parameters under the same spectral efficiency of 1.333 bit/s/Hz. It is considered that the ZTM-OFDM-IM with the proposed constellation pair has the largest $d_{\min}$.

When the spectral efficiency is 2.222 bit/s/Hz, a constellation pair shown in Figure 4 has been used in the previous ZTM-OFDM-IM system [14], where $a=(1+\sqrt{2})$, $b=-(1+\sqrt{2})$. The inner 4-QAM constellation is for the mapper A, and the outer 8 signal points are for the mapper B, that is, $M_A=4$ and $M_B=8$, respectively. Let d be the MED between a signal point in A and the nearest signal point in B. In addition, the MED from the origin to any signal point in the constellation pair is also equal to d. Then, the Euclidean distances d_A of the constellation A and d_B of the constellation B are equal to $d_A=d_B=\sqrt{2}d$. Such constraints can be met with the new constellation pair presented in Figure 5. Both constellation pairs use the same signal set for the mapper A. However, the outer signal set for the mapper B in Figure 5 is different from the one in Figure 4. Nevertheless, both pairs have the same parameter set of $(n,k,k_A,k_B)=(4,3,2,1)$ for designing the ZTM-OFDM-IM system as given in Table 2. It is noted that the proposed constellation pair has a larger normalized MED of 1.886 than the previous one. As presented in Table 2, the ZTM-OFDM-IM system with the proposed constellation pair has the largest $d_{\min}$. It implicitly implies that the system with a new constellation pair may have slightly improved error performance.

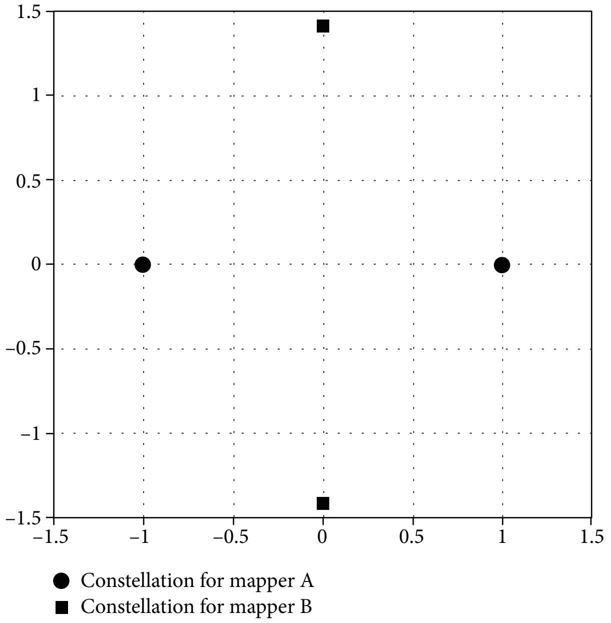


FIGURE 2: The constellation pair with $M_A=M_B=2$ in [14].

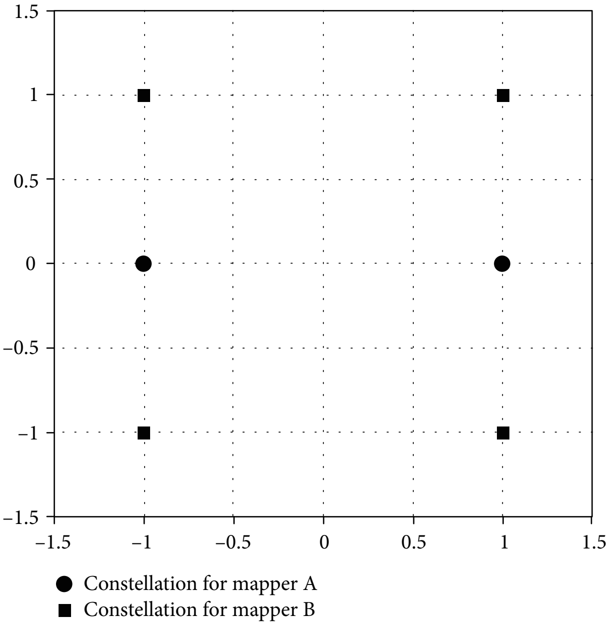


FIGURE 3: The new constellation pair with $M_A=2$ and $M_B=4$.

4. Performance Analysis

To evaluate performance of the proposed constellation pairs for the ZTM-OFDM-IM system, computer simulation has been carried out. The parameters presented in Tables 1 and 2 have been exploited to design the OFDM systems based on the IM scheme. We implement the frequency-selective Rayleigh fading channel as well as the AWGN channel using Matlab program as a transmission environment for simulation.

BERs of the OFDM systems with the IM scheme in the AWGN and frequency-selective Rayleigh fading channels are plotted in Figures 6 and 7 , respectively. Here, the

Table 1: The parameters for IM schemes when $\eta = 1.333$ bit/s/Hz.

Modulation	Constellation	M_A	M_B	n	k_A	k_B	P	$d_{\min}$
ZTM-OFDM-IM	Previous (Figure 2)	2	2	4	2	1	6	2.309
	Proposed (Figure 3)	2	4	4	1	1	6	2.667
DM-OFDM-IM	$\{1,-1\}, \left\{\sqrt{3}j,-\sqrt{3}j\right\}$	2	2	4	2	2	6	1.633
IM-OFDM-IM	16-QAM	$M = 16$		4	$k = 1$		6	1.461

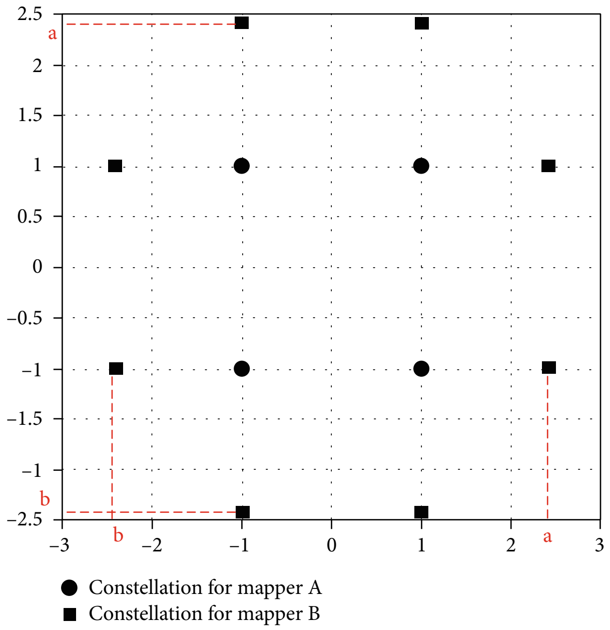


Figure 4: The previous constellation pair with $M_A = 4$ and $M_B = 8$.

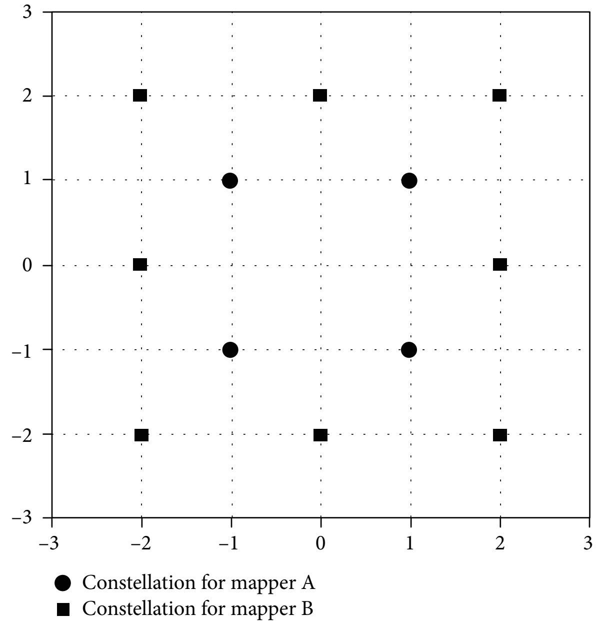


Figure 5: The new constellation pair with $M_A = 4$ and $M_B = 8$.

parameters presented in Table 1 are used. To make the systems have the same spectral efficiency of 1.333 bit/s/Hz, the subcarriers in the previous ZTM-OFDM-IM system generated by the mapper A and the mapper B shown in Figure 2 are $k_A = 2$ and $k_B = 1$, respectively, while those in the proposed ZTM-OFDM-IM system produced by the mappers shown in Figure 3 are $k_A = k_B = 1$. In the DM-OFDM-IM system, four subcarriers are divided into $k_A = k_B = 2$, that is, no inactive subcarrier is exploited. In the case of the OFDM-IM system, only one subcarrier is mapped using 16-QAM.

As a result, the ZTM-OFDM-IM system with the recommended constellation pair shows better error performance than the previous system. When the signal-to-noise ratio (SNR) is lower than 8.5 dB in the AWGN environment, it can be observed that the DM-OFDM-IM system has better error performance than other systems as shown in Figure 6. However, since BER is higher than 3.0×10^{-4} in this case, it is considered that the DM-OFDM-IM is not suitable for application to a communication system. Specifically, for the BER of 10^{-6} as a reference, the proposed system has about 1.0 dB improvement in SNR over the conventional one. And the error performance of about 0.6 dB and 2.5 dB is improved compared to the DM-OFDM-IM and OFDM-IM systems, respectively. It is analyzed that such performance improvement of the proposed ZTM-OFDM-IM system is caused by the increased MED between subblocks.

In the Rayleigh fading channel, both ZTM-OFDM-IM systems have almost the same BER. However, it can be observed that those systems still have a performance gain of about 3.0 dB and 5.5 dB, respectively, compared to the DM-OFDM-IM and OFDM-IM systems at a reference BER of 10^{-5}. It is due to the fact that the former systems have much increased MEDs compared to the latter systems as shown in Table 1.

When the spectral efficiency is increased to 2.222 bit/s/Hz, the simulation results in the AWGN and Rayleigh fading channels are plotted in Figures 8 and 9 , respectively. Since the constellation pairs shown in Figures 4 and 5 are applied to the ZTM-OFDM-IM system, mapper A and mapper B are changed to $M_A = 4$ and $M_B = 8$, respectively, as presented in Table 2. In the DM-OFDM-IM system, mapper A and mapper B exploit the constellations shown in Figure 2, so $M_A = M_B = 4$, and the OFDM-IM system generates subcarrier signals using 16-QAM.

Similar to Figure 6, the ZTM-OFDM-IM system with the proposed constellation pair has about 1.8 dB and 2.6 dB SNR gain in the AWGN channel, respectively, compared to the DM-OFDM-IM and OFDM-IM systems at the reference BER of 10^{-6}. However, the difference in BERs of the previous and the new ZTM-OFDM-IM system is reduced to 0.3 dB.

Table 2: The parameters for IM schemes when $\eta = 2.222$ bit/s/Hz.

Modulation	Constellation	M_A	M_B	n	k_A	k_B	P	$d_{\min}$
ZTM	Previous (Figure 4)	4	8	4	2	1	10	1.812
	Proposed (Figure 5)	4	8	4	2	1	10	1.886
DM	Figure 2 in [13]	4	4	4	2	2	10	1.371
IM	16-QAM	$M = 16$		4	$k = 2$		10	1.333

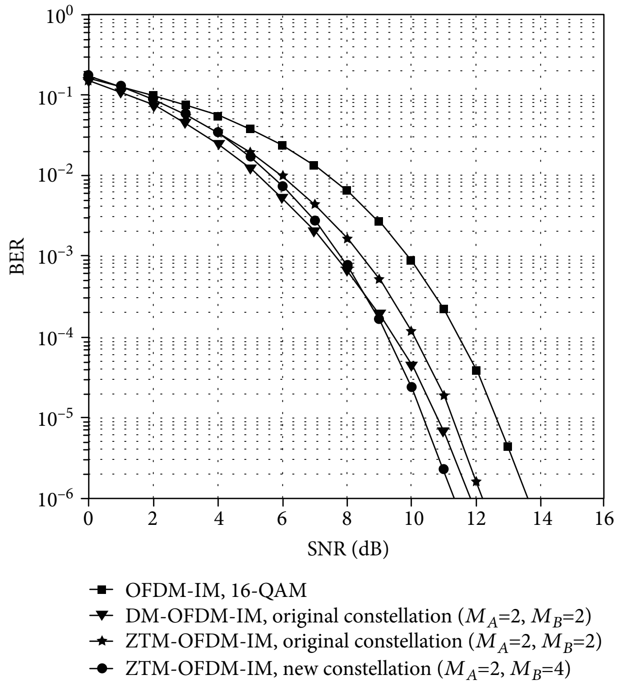


Figure 6: BERs in the AWGN channel when $\eta = 1.333$ bit/s/Hz.

As the MED gain of the new constellation pair over the previous pair decreases, the amount of performance improvement also decreases. The simulation results on the frequency-selective Rayleigh fading channel also show that the ZTM-OFDM-IM system with the new constellation pair has the best error performance as shown in Figure 9. Hence, maximizing the MED between different subblocks is crucial for improving performance of the OFDM system based on index modulation schemes.

5. Conclusions

This paper introduces new signal constellation pairs which improve the error performance of the ZTM-OFDM-IM system. In the presented system, one of the two mappers for mapping active subcarriers uses the same signal constellation as the one used in the previous work, and the other exploits a constellation larger than the one used in [14]. This causes the MED between the subblocks of the ZTM-OFDM-IM system with new constellation pairs to be increased under the constraint of the same spectral efficiency. It is a major factor to improve the bit error performance of the proposed system.

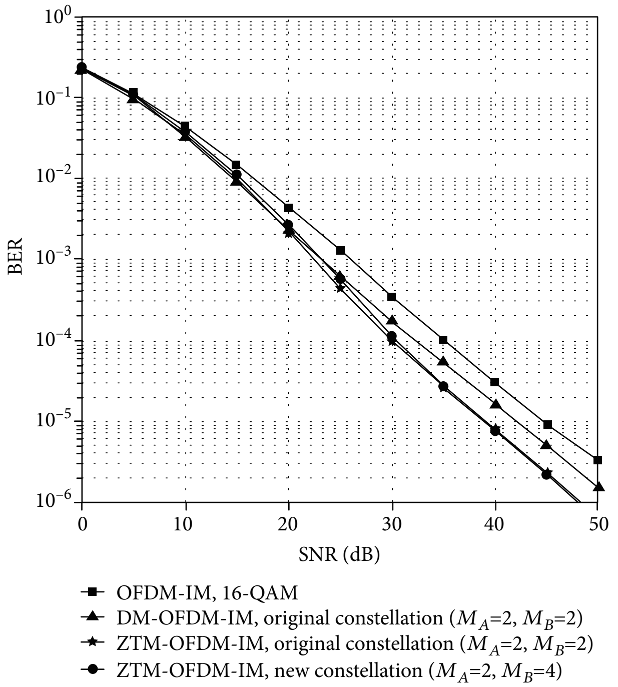


Figure 7: BERs in the Rayleigh fading channel when $\eta = 1.333$ bit/s/Hz.

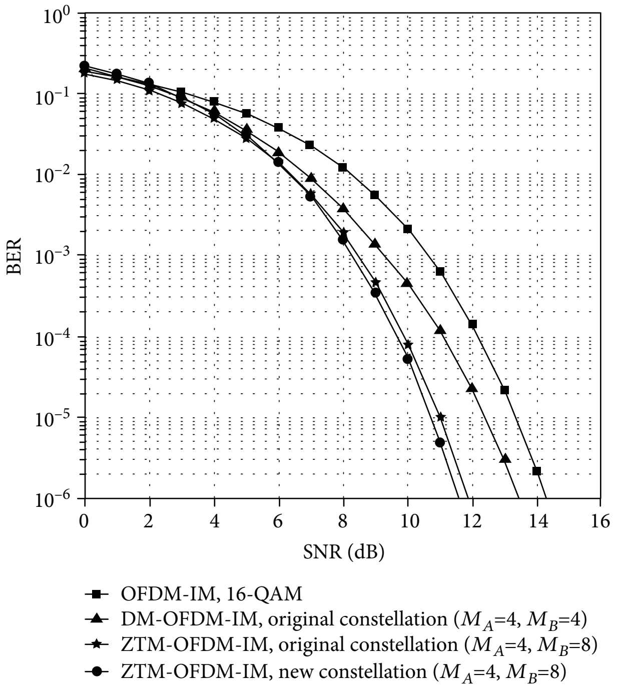


Figure. 8: BERs in the AWGN channel when $\eta = 2.222$ bit/s/Hz.

As a result of simulation in ideal AWGN and frequency-selective Rayleigh fading channel environments, the ZTM-OFDM-IM system with the proposed constellation pairs has much lower BER than the typical OFDM system with IM scheme such as OFDM-IM and DM-OFDM-IM. In addition, the new ZTM-OFDM-IM system shows slightly better

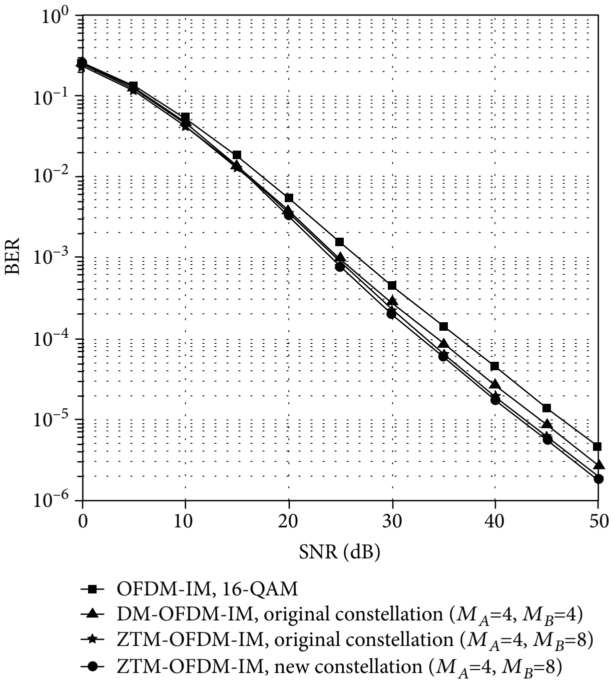


Figure 9: BERs in the Rayleigh fading channel when $\eta = 2.222$ bit/s/Hz.

performance than the same system with the conventional constellation pairs. The new constellation pairs proposed in this paper increase the MED between subblocks, demonstrating the possibility of improving error performance of the ZTM-OFDM-IM system. Hence, it is considered that a further study on generalized design of the constellation pairs for the mappers of the ZTM-OFDM-IM system is necessary in the future. In addition, another investigation on the IM-based OFDM system is necessary to further improve spectral and energy efficiencies while suppressing system complexity.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] J. Joung, C. K. Ho, and S. Sun, “Spectral efficiency and energy efficiency of OFDM systems: impact of power amplifiers and countermeasures,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 2, pp. 208–220, 2014.

[2] F. Adib and D. Katabi, “See through walls with Wi-Fi!,” *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 75–86, 2013.

[3] D. Astely, E. Dahlman, A. Furuskär, Y. Jading, M. Lindström, and S. Parkvall, “LTE: the evolution of mobile broadband,” *IEEE Communications Magazine*, vol. 47, no. 4, pp. 44–51, 2009.

[4] A. Ghosh, D. R. Wolter, J. G. Andrews, and R. Chen, “Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential,” *IEEE Communications Magazine*, vol. 43, no. 2, pp. 129–136, 2005.

[5] A. Peled and A. Ruiz, “Frequency domain data transmission using reduced computational complexity algorithms,” in *ICASSP '80. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 964–967, Denver, Colorado, USA, USA, Apr. 1980.

[6] E. Basar, M. Wen, R. Mesleh, M. Di Renzo, Y. Xiao, and H. Haas, “Index modulation techniques for next-generation wireless networks,” *IEEE Access*, vol. 5, pp. 16693–16746, 2017.

[7] T. Mao, Q. Wang, Z. Wang, and S. Chen, “Novel index modulation techniques: a survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 315–348, 2019.

[8] R. Abu-Alhiga and H. Haas, “Subcarrier-index modulation OFDM,” in *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 177–181, Tokyo, Japan, Sep. 2009.

[9] D. Tsonev, S. Sinanovic, and H. Haas, “Enhanced subcarrier index modulation (SIM) OFDM,” in *2011 IEEE GLOBECOM Workshops*, pp. 728–732, Houston, TX, USA, Dec. 2011.

[10] E. Basar, U. Aygolu, E. Panayirci, and H. V. Poor, “Orthogonal frequency division multiplexing with index modulation,” *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5536–5549, 2013.

[11] M. Wen, B. Ye, E. Başar, Q. Li, and F. Ji, “Enhanced orthogonal frequency division multiplexing with index modulation,” *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4786–4801, 2017.

[12] R. Fan, Y. J. Yu, and Y. L. Guan, “Generalization of orthogonal frequency division multiplexing with index modulation,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5350–5359, 2015.

[13] T. Mao, Z. Wang, Q. Wang, S. Chen, and L. Hanzo, “Dual-mode index modulation aided OFDM,” *IEEE Access*, vol. 5, pp. 50–60, 2017.

[14] T. Mao, Q. Wang, J. Quan, and Z. Wang, “Zero-padded orthogonal frequency division multiplexing with index modulation using multiple constellation alphabets,” *IEEE Access*, vol. 5, pp. 21168–21178, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 4380462, 14 pages
https://doi.org/10.1155/2020/4380462

Research Article

OEDDBOS: An Efficient Data Distributing Strategy with Energy Saving in Sensor-Cloud Systems

Qifei Zhao,[1] Gaocai Wang ,[2] Ying Peng,[2] and Yuting Lu[2]

[1] *College of Electrical Engineering, Guangxi University, Nanning, 53004 Guangxi, China*
[2] *School of Computer and Electronic Information, Guangxi University, Nanning, 53004 Guangxi, China*

Correspondence should be addressed to Gaocai Wang; wanggcgx@163.com

Received 19 January 2020; Revised 25 May 2020; Accepted 12 June 2020; Published 7 July 2020

Academic Editor: KI-IL Kim

Sensor-cloud is a developing technology and popular paradigm for various applications. It integrates wireless sensor into a cloud computing environment. On the one hand, the cloud offers extensive data storage and analytical and processing capabilities not available in sensor nodes. On the other hand, data distribution (such as time synchronization and configuration files) is always an important topic in such sensor-cloud systems, which leads to a rapid increase in energy consumption by sensors. In this paper, we aim to reduce the energy consumption of data dissemination in sensor-cloud systems and study the optimization of energy consumption with time-varying channel quality when multiple nodes use the same channel to transmit data. Suppose that there is a certain probability that the nodes send data for competing channel. And then, they decide to distribute data in terms of channel quality for saving energy after getting the channel successfully whether or not. Firstly, we construct the maximization problem of average energy efficiency for distributing data with delay demand. Then, this maximization problem transferred an optimal stopping problem which generates the optimal stopping rule. At last, the thresholds of the optimal transmission rate in each period are solved by using the optimal stopping theory, and the optimal energy efficiency for data distribution is achieved. Simulation results indicate that the strategy proposed in this paper can to some extent improve average energy efficiency and delivery ratio and enhance energy optimization effect and network performance compared with other strategies.

1. Introduction

With the rapid development of sensor networks, different wireless sensor networks are quickly becoming popular. Wireless sensors equipped with a variety of wireless interfaces such as Wi-Fi, Bluetooth, and cellular networks have powerful wireless communication capabilities. In particular, by combining wireless sensor networks and the cloud, the concept of sensor-cloud systems appeared [1–6]. The cloud provides extensive data storage and analytical and processing capabilities, while wireless sensors are responsible for collecting data. Despite the fact that the cloud has helped to break through lots of wireless sensor network limitations, there are still other challenges to be solved. The major problem is that data (time synchronization and configuration files) to be distributed and transmitted can significantly affect the sensor-cloud systems' performance due to limited energy supply. Sensor-cloud systems have been deployed in many applications fields such as agriculture, military, healthcare, environmental monitoring, and manufacturing, in particular, for remote mountainous area or a large area of forest, where the power cannot be recharged or the sustained power cannot be provided. A sensor node can perform various types of data communication with others and distribute shared data to others within its transmission range, for example, sensed data is sent to some sinks and the cloud. The sinks distribute synchronous data to all sensors and so on. On the other hand, distributing a great deal of data will rapidly exhaust the power and impact the normal operation of the sensor or the sink, and especially at certain place where timely supply of power cannot be realized. Therefore, data distributing strategies with energy saving and performance guarantee is an important subject in the study of energy consumption and performance optimization in such sensor-cloud systems.

In the nodes' communication of sensor-cloud systems, the wireless channel's quality changes randomly with time because of its essential attributes, multipath propagation, and environmental interference and so on. If one node (sensor/sink) is selected to disseminate data when the channel quality is in good condition, the energy consumption generated in data dissemination will be effectively reduced. In some applications, multiple nodes use the same channel to distribute data, but only one node is allowed to use it in a given time period which will cause transmission collisions and data distribution failure if multiple (more than one) nodes use one channel to distribute data at the same time. In order to reduce energy waste caused by transmission collisions, the sending nodes must detect the channel service condition by successfully receiving a response signal from the receiving nodes within a given time when they distribute data to the receiving nodes [7, 8]. And the channel service quality is evaluated on the basis of the power of the signal. When only one node is detected to use the channel, the sending node decides whether or not to continue to distribute data in terms of the channel service quality, in order to avoid the increase in the energy consumption of the distribution data when the channel service quality is in poor condition. This process is named channel competition as the sending node detects the channel service condition. It is the key to save energy as the sending node selects the optimal channel service quality to distribute data in the context of successfully competing for the channel. In [7], the authors use the optimal stopping theory to study the energy efficiency issue when multiple mobile nodes compete for the same channel to distribute data. But they do not consider the delay requirements of data distribution. Therefore, the energy optimization effect under the delay constraint scenario is poor. In addition, it has a fixed time length of the data transmission. The transmission energy efficiency will vary if the channel quality changes during data transmission.

In this paper, we consider the data distribution delay requirements in sensor-cloud systems. And it is also consistent with the realistic situation that the random varying channel quality has a certain holding time. The channel variation model is used in [8] as well. The sending node determines the length of the distribution time according to the amount of data to be distributed and the channel holding time. In [9], the authors used the game theory to study the optimal probability of multiple mobile nodes competing for the same channel, and then reducing the energy waste of transmission collisions. But it does not take into account the impact of channel quality on distribution energy consumption, which will affect the energy saving of the data distribution.

Therefore, this paper proposes an optimal energy efficiency distribution strategy for data distributing in sensor-cloud systems when the node competes for the same channel. And the maximum distribution delay constraint of the data distribution is also considered. In sensor-cloud systems, the sending node participates in channel competition with a certain probability, observes the channel quality after successfully competing to the channel, then decides whether or not to distribute data based on the channel quality condition. The optimal energy efficiency moment is selected for the data distribution so as to realize energy saving. Optimal energy efficiency means that the average amount of the data distribution per unit of energy consumption reaches a maximum. The problem of stopping observation and distributing data through the sending node constantly observing the channel quality to select a good channel quality is an optimal stopping rule [10, 11] problem. In this stop-rule problem, the sending node continuously observes the variation of the channel quality and obtains the stopping time when the expected reward of energy efficiency is maximized. Then, it takes the action of distributing data to achieve the goal of maximizing the expected reward.

The paper is organized as follows. Section 2 reviews the related research work. Section 3 presents the sensor-cloud model and optimization issues. Section 4 introduces the optimal energy efficiency data dissemination strategy based on the optimal stopping theory. Section 5 shows simulation results and analysis. At last, we make a conclusion and point out some possible directions for further work in Section 6.

2. Related Research Work

Data distribution is always a hot topic in distributing networks, and its applications are applied widely in various networks, such as ad hoc networks, social networks, and wireless sensor networks. Researchers pay extensive attention to data distribution applications and their research goals include reducing distribution energy consumption, increasing throughput, and reducing delay.

At present, the integration of wireless sensor networks and the cloud have been proposed in many architectures and research frameworks [1, 2, 12]. The majority of these works study a three-layer architecture with the focus on publish/subscribe communication between the sinks and cloud layers, trying to push communication between sensors and sinks layers. In [13–15], the authors mainly focus on improving the sensor-cloud performance by raising an optimization problem.

In [13], the authors put forward a three-tier architecture named CEB (cloud, edge, and beneath). To enhance scalability and energy efficiency, CEB uses two algorithms in optimizing data transmission rates between cloud and edge layers as well as edge and sensor layers, respectively. The optimization is performed with respect to energy consumption of sensor nodes. In [14], the authors focused on push-pull hybrid communication between the sensor and cloud layers through the edge layer. They come up with an optimization algorithm for sensor and sink nodes concerning different objectives. The purpose is to find the optimal transmission rate of each node. The optimization is conducted in terms of data yield, bandwidth, and energy consumption. In [15], the paper discusses how to solve the problem of energy constraints of sensor-cloud infrastructure so as to improve its performance and prolong its lifetime, which is really necessary in practice. On this basis, a sensor-cloud optimization strategy is proposed to improve its lifetime and energy efficiency. Formulated as integer linear programming, the strategy can determine the optimal number of push or pull periods for data collection or requesting and thus maximize the lifetime of sensors layer. Furthermore, it optimizes sensor-cloud lifetime with regard to energy consumption of sensors and sinks under the different push and pull mechanisms.

On the other hand, another research work has been mentioned in different literature, such as in [7–9, 16]. In [7], the authors assumed that multiple sending terminals take part in channel competition with equal probability and transmit data within a given time. They use the optimal stopping theory to deduce the optimal transmission rate of the data dissemination by the sending terminal so as to achieve greater amount of data dissemination per unit of energy consumption and higher energy efficiency. In [9], the authors studied the energy optimization problem of data dissemination from the perspective of the probability of participating in wireless network channel competition. Game theory is applied to derive the optimal probability that multiple sending terminals participate in channel competition to reduce the energy consumption of data transmission collision and data dissemination. In [16], the authors studied the optimal control from the perspective of flood data dissemination in mobile social networks. What is more, through the dynamic programming, the time dissemination problem of the optimal control signal is solved and the total network cost is minimized.

In [8], the authors study the distributed opportunity scheduling problem that multiple sending terminals compete for the same channel for data dissemination in mobile ad hoc networks. Based on the optimal stopping theory, the paper proposed an optimal stopping problem in order to maximize network throughput. And it also calculates the optimal transmission rate of the sending terminal by a backward induction method, thereby maximizing the throughput of the entire network. However, this paper does not consider the situation of data transmission delay and assumes that the sending terminal always has enough data to be distributed at any time. In [17], the authors study the data dissemination problem when the channel quality changes randomly in mobile networks. In this paper, the optimal transmission rate threshold is solved based on the optimal stopping theory to achieve the maximum network throughput. To this end, it also uses noncooperative games to develop the users' best response strategy.

In general, the researchers addressed the issue of data dissemination in different network backgrounds, such as ad hoc networks and social networks, and used different optimization methods, such as the optimal stopping theory, game theory, and heuristic algorithm. But the final goal is to reduce data dissemination energy consumption and optimize network performance. However, some researches do not consider the problems of data transmission delay, while others do not take into account random variation of channel quality.

In [18, 19], we study the energy consumption of data transmission for that the channel quality changes randomly in wireless networks and get an optimal transmission rate using the optimal stopping theory to solve a constant data generated rate and variable data generated rate. Therefore, based on our previous work, we apply our approaches and models to study the energy efficiency issues with delay constraint data in sensor-cloud systems in which some similar issues exist. Energy consumption is also a serious concern for users and researchers, and the characteristics of channel quality changes with time. So, in this paper, our goal is to improve network energy efficiency while ensuring sensor-cloud systems' performance. The main contributions of this paper include the following aspects. (1) Considering the case where the channel quality changes randomly with time and having a certain holding time, an energy efficiency model and an optimal energy efficiency model in which multiple nodes compete for the same channel for data dissemination are constructed in sensor-cloud systems. (2) Construct the optimal stopping rule problem based on optimal average energy efficiency to obtain the approximate optimal stopping rule. And within the maximum transmission delay, use the optimal stopping theory to solve the optimal transmission rate threshold for data transmission during channel quality maintenance period.

3. Network Model and Problem Description

3.1. Network Model. In a sensor-cloud system, such as the sink node sends a Q_d data with the maximum delay D_m. to sensors at intervals of T_{diss}. Sensors which are beyond the transmission range of the sink node cannot directly receive the data, and other sensors receiving the data need to distribute data to them. The data distribution model in a sensor-cloud system is shown in Figure 1.

The conflicting of the transmission and the failure of the data distribution will appear if multiple sinks use the same channel to disseminate data and multiple (more than one) sensors are enclosed in the transmission interference range. However, the data transmission will be successful if only one sink uses the channel to transmit data at the same time. The channel will be idle if there is no data to be transmitted. In order to detect the status of the channel, the sending sink transmits data to the receiving sensor for duration of τ. And the transmission is successful if the acknowledgment signal from the receiving sensor is returned successfully in time τ. This successful transmission is regarded as a successful competition channel of a sending node, and τ is the channel competition period. Assume that the channel quality is held on for a time T. That means the sending sink competes successfully with the channel and obtains the opportunity to continue to disseminate data for a period T. Indeed, the period T is called a period of one-channel competition and data dissemination. Obviously, the channel competition period τ satisfies $\tau \ll T$. Since the channel quality changes randomly with time, the sending sink has competed successfully the channel and detected the channel quality according to the acknowledgment signal power value sent back by the receiving nodes. The data will be disseminated if the channel quality is good. Otherwise, the opportunity is given up. The other nodes have to wait for the next cycle T to participate in the channel competition again if the sending node chooses to

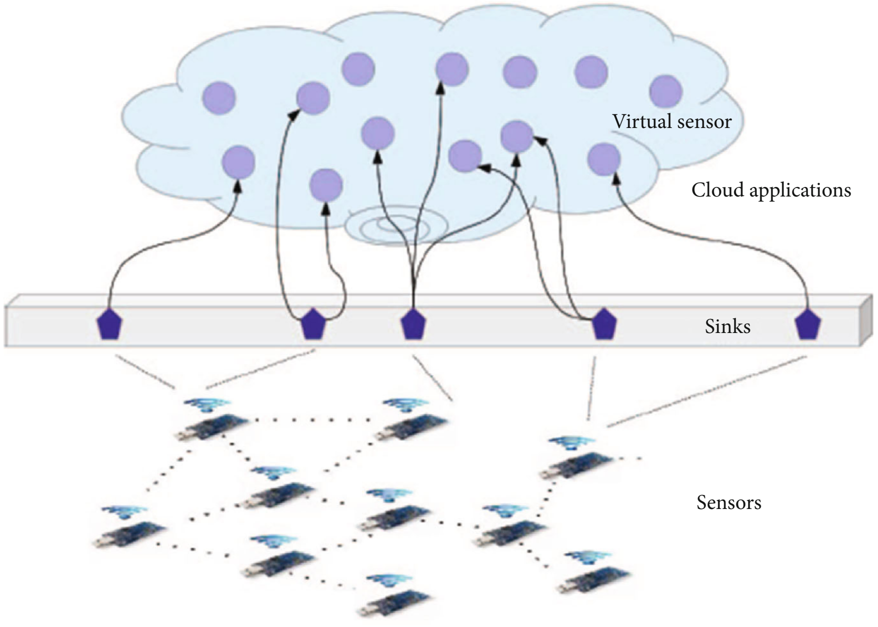


Figure 1: A data distribution model for sensor-cloud system.

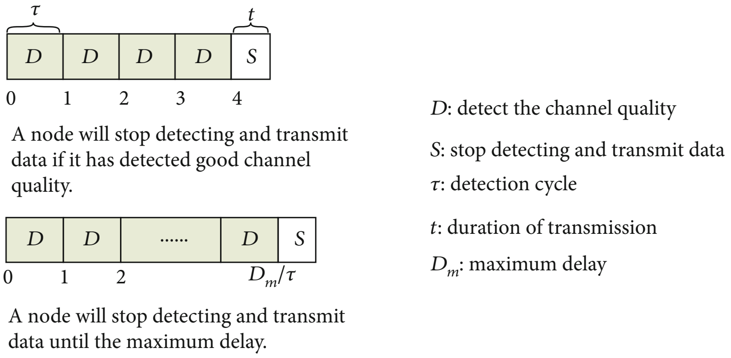

τ
t
D
D
D
D
S
0
1
2
3
4
A node will stop detecting and transmit data if it has detected good channel quality.
D
D
......
D
S
0
1
2
D_m/τ
A node will stop detecting and transmit data until the maximum delay.
D: detect the channel quality
S: stop detecting and transmit data
τ: detection cycle
t: duration of transmission
D_m: maximum delay

Figure 2: One round of channel competition and data dissemination process.

distribute data. On the other hand, the sending node could not participate in the channel competition for the remaining time of the current round if it gives up the channel, but the other nodes could continue to compete for the channel. One round of channel competition and data dissemination process is shown in Figure 2.

3.2. Problem Description. It is assumed that the number of sending nodes using the same channel is K within the transmission interference range. These nodes participate in the channel competition with an equal probability p, and the value is $1/K$. At the beginning of each round of channel competition, K sending nodes compete for the channel with probability p. If there is only one node competing for the channel, then the channel competition turns out to be successful, which is called the first round channel competition success, and the sending node is represented by A. On the condition that the current channel quality is good, A continues to disseminate data, while the others wait to compete for the channel in the next round. Or node A stops disseminating data, quits this round of channel competition, and waits to participate in the next round, while the remaining $K-1$ sending nodes continue the competition. When there is only one node competing for the channel, the second round is declared successful, and that node is represented by B. Then, B performs the same operation as A. Obviously, node A successfully competes for the first round and continues to disseminate data if the channel quality is good in the current cycle. Otherwise, each node that successfully occupies the channel has to give up using the channel which is thereby successfully competed for K times. Therefore, the probability of successful channel competition in ith round is

$$\rho_i^s = \binom{1}{K-i+1} p(1-p)^{K-i}, \tag{1}$$

The total times of expected channel competition is $b_i = 1/\rho_i^s$ in the ith round successful competition process. The number of competition expectations for each sending node is $b_i \cdot p$ in ith competition. And in each round of K time competition, competition expectations total $\sum_{i=1}^{K} b_i p$. With the transmission power consumption of the sending node being P and the duration of each competition being τ, the competition energy consumption is $P \cdot \tau$. Therefore, the expected competition power consumption of the sending node is $\sum_{i=1}^{K} b_i pP\tau$ in each round of channel competition. The sending node which wins the channel competition disseminates data in the 1st round if the channel quality is good. The expected duration of competition is $b_1\tau$ and the time length of the remaining transmission is $T - b_1 \cdot \tau$.

Under normal circumstances, data dissemination is subject to a certain deadline in practical applications and the data not yet transmitted within the deadline will be discarded. This paper assumes that the maximum data delay is D_m. If a sending node successfully competes for the channel and carries the data to be distributed which is going to reach the maximum delay D_m, then it distributes data regardless of the channel quality to avoid discarding the data due to timeout. Obviously, the competing expectations of the sending node add up to $b_1 \cdot p$, and accordingly the expected competition energy is $b_1 \cdot p \cdot P \cdot \tau$. The expected competition time is $b_1 \cdot \tau$; therefore, the time length of the remaining transmission is $T - b_1 \cdot \tau$.

According to the Shannon formula, when the sending node's transmission power consumption is given, the channel quality is better and the transmission rate is bigger. That means the more data is disseminated by the sending node at the same time, and the more data is transmitted by using the same power. Defining the energy efficiency is the amount of data bits that can be disseminated in per unit of joule energy. Our goal is to obtain the maximum or optimal energy efficiency when the sending node competes successfully to the channel for data dissemination and to improve the energy efficiency of data dissemination with guarantee of data transmission delay.

4. Optimal Energy Efficiency Data Dissemination Strategy Based on Optimal Stopping Theory

4.1. Construction of Optimal Energy Efficiency Problems. Assume that the sending node successfully obtains the channel in the ith competition of one round channel contention and set the channel transmission rate as R. Then, the energy efficiency of the node for data dissemination is

$$\gamma = \frac{R\tilde{t}}{P\tilde{t} + E_{\text{total}}}. \tag{2}$$

Here, R is the transmission rate, and E_{total} is the total competing and reference energy consumption before the sending node gets ith transmission opportunity. And it is called the total detection energy consumption, which is calculated from the next round of channel competition after the last successful data transmission and includes competing and reference energy consumption in this round. The reference energy consumption P_0 is the basic energy consumed by each current device of the sending node in the active state (including the idle and transmission state). $\tilde{t}$ represents the actual transmission time which is determined by the amount of data to be distributed and the transmission rate R. It is defined as follows:

$$\tilde{t} = \begin{cases} Q_d/R, & Q_d/R \le T - b_1 \cdot \tau, \\ T - b_1 \cdot \tau, Q_d/R > T - b_1 \cdot \tau. \end{cases} \tag{3}$$

In order to reduce energy consumption, each sending node maximizes its own energy efficiency to achieve optimal energy efficiency. Therefore, the optimal energy efficiency γ_{op} of the sending sensor can be defined as follows:

$$\gamma_{\text{op}} = \max \frac{R\tilde{t}}{P\tilde{t} + E_{\text{total}}}. \tag{4}$$

Each sending node competes for the channel with probability p. And after obtaining successfully the channel, it decides to disseminate data immediately or give up the use of the channel according to the current channel transmission rate and the amount of data to be disseminated. The dissemination data will be discarded if the sending node has the expectation to obtain greater energy efficiency. Otherwise, the data is disseminated immediately. The sending node, based on the amount of data to be disseminated, continuously observes the quality condition of the successfully competing channel and makes a decision whether or not to use the channel for data dissemination. Obviously, this problem is the optimal stopping rule problem that the sending node chooses the optimal time to stop observing and distribute data for obtaining the optimal energy efficiency. The optimal stopping theory provides the theoretical analysis base and practical solutions.

4.2. Optimal Stopping Rule Problem of Optimal Energy Efficiency. After the sending node receives the data with the maximum delay of D_m, the node starts to compete for the channel and observes the quality of the channel being successfully competed for. If the channel quality is up to the optimal energy efficiency standard, it will disseminate data; otherwise, it will wait for the next round of the competition. If the sending node carrying the data keeps observing until the maximum delay D_m reaches the deadline, it must disseminate data after occupying the channel so as to avoid the data discarding due to the excessive delay. Therefore, the sending node participates in channel competition with $M = \lfloor D_M/T \rfloor$ rounds. If the sending node obtains the channel and distributes the data in the Nth $(1 \le N < M)$ round, its energy efficiency is

$$\gamma_N = \frac{R_N \tilde{t}_N}{P\tilde{t}_N + \sum_{n=1}^{N} \left(\sum_{i=1}^{K} b_i \cdot p \cdot P \cdot \tau + P_0 \cdot T \right)}. \tag{5}$$

In (5), R_N represents the transmission rate obtained by the sending node in the Nth round; $\tilde{t}_N$is the actual time length of data dissemination at this moment, which is defined in (3); $\sum_{i=1}^{K} b_i pP\tau$ represents the expected competing energy consumption. P_0 is the reference power consumption of the sending node and $P_0 \cdot T$ is the reference energy consumption. Since the sending node has to disseminate data after successfully competing for the channel in the Mth round, the energy efficiency of the Mth round shall be

$$\gamma_M = \frac{R_M \tilde{t}_M}{P\tilde{t}_M + \sum_{n=1}^{M-1}\left(\sum_{i=1}^{K} b_i p P\tau\right) + b_1 p P\tau + M P_0 T}. \quad (6)$$

It is called one data dissemination, which refers to the process from the sending node competing for the channel to winning the competition and disseminating data. Indeed, one data distribution includes one or more rounds of channel competition and one data dissemination. N_y is defined as the number of channel competition rounds of data dissemination in the yth. That is, the time is the period when channel competition stops and data dissemination starts, which is referred to the stop time. If the sending node repeatedly disseminates data for Y times, the corresponding stop time sequence is $\{N_1, N_2, \cdots, N_y, \cdots, N_Y\}$. The transmission rate sequence is $\{R_{N_1}, R_{N_2}, \cdots, R_{N_y}, \cdots, R_{N_Y}\}$. The actual transmission time length sequence is $\{\tilde{t}_{N_1}, \tilde{t}_{N_2}, \cdots, \tilde{t}_{N_y}, \cdots, \tilde{t}_{N_Y}\}$. And the detected total energy consumption sequence is $\{E_{\text{total},N_1}, E_{\text{total},N_2}, \cdots, E_{\text{total},N_y}, \cdots, E_{\text{total},N_Y}\}$. Among them, R_{N_y}, $\tilde{t}_{N_y}$, and E_{total,N_y} represent the transmission rate, actual transmission time length, and total energy consumption of the data dissemination after the sending node competing for the channel N_y times in the yth round, respectively. Based on (5) and (6), the average energy efficiency of the sending node is

$$\chi = \sum_{y=1}^{Y} \gamma_{N_y} = \frac{\sum_{y=1}^{Y} R_{N_y} \tilde{t}_{N_y}}{\sum_{y=1}^{Y}\left(P\tilde{t}_{N_y} + E_{\text{total},N_y}\right)}, \quad (7)$$

$$\tilde{t}_{N_y} = \begin{cases} Q_d/R_{N_y}, Q_d/R_{N_y} \le T - b_1 \cdot \tau, \\ T - b_1 \cdot \tau, \quad Q_d/R_{N_y} > T - b_1 \cdot \tau, \end{cases} \quad (8)$$

$$E_{\text{total},N_y} = \sum_{n=1}^{N_y} (E_{c,n} + P_0 \cdot T), \quad (9)$$

$$E_{c,n} = \begin{cases} \sum_{i=1}^{K} b_i \cdot p \cdot P \cdot \tau, 1 \le n < M, \\ b_1 \cdot p \cdot P \cdot \tau, \quad n = M. \end{cases} \quad (10)$$

According to the law of large numbers, (7) is transformed into the following:

$$\chi = \frac{E\left[R_N \tilde{t}_N\right]}{E\left[P\tilde{t}_N + E_{\text{total},N}\right]}. \quad (11)$$

Here, $E[\bullet]$ and N, respectively, represent the mathematical expectation and the number of competition rounds before data dissemination, which is also the stop time of the channel competition. With the data to be disseminated being subject to the maximum delay D_m, the maximum stop time of the channel contention is represented by M. The stop time set for channel contention is defined as $\Omega = \{N : 1 \le N \le M\}$. So the biggest problem about the average energy efficiency in this paper is

$$\max_{N\in\Omega} \frac{E\left[R_N \tilde{t}_N\right]}{E\left[P\tilde{t}_N + E_{\text{total},N}\right]}. \quad (12)$$

According to (11) and (12), the optimal average energy efficiency (the average value of the maximum amount of data per unit energy transmission) is

$$\chi* = \sup_{N\in\Omega} \frac{E\left[R_N \tilde{t}_N\right]}{E\left[P\tilde{t}_N + E_{\text{total},N}\right]}. \quad (13)$$

Furthermore, formula (13) is transformed by an equation transformation into the following:

$$\sup_{N\in\Omega}\left(E\left[R_N \tilde{t}_N\right] - \chi * \left(E\left[P\tilde{t}_N + E_{\text{total},N}\right]\right)\right) = 0. \quad (14)$$

Formula (14) is an average energy efficiency maximization problem concerning the maximization of expected reward $E[\Phi_N(\chi)]$ about χ, where the reward function $\Phi_N(\chi)$ is

$$\Phi_N(\chi) = R_N \tilde{t}_N - \chi\left(P\tilde{t}_N + E_{\text{total},N}\right). \quad (15)$$

Therefore, the average energy efficiency maximization problem of (12) is converted into an optimal stopping problem.

$$\begin{aligned}\psi(\chi) &= \max_{N\in\Omega} E[\Phi_N(\chi)] \\ &= \max_{N\in\Omega}\left(E\left[R_N \tilde{t}_N\right] - \chi\left(E\left[P\tilde{t}_N + E_{\text{total},N}\right]\right)\right),\end{aligned} \quad (16)$$

The aim of (16) is to gain the optimal stopping time $N* = N(\chi *)$ of the channel competition, so as to obtain the optimal average energy efficiency $\chi *$. So, we have

$$N* = \arg\ \sup_{N\in\Omega} \frac{E\left[R_N \tilde{t}_N\right]}{E\left[P\tilde{t}_N + E_{\text{total},N}\right]}. \quad (17)$$

The average energy efficiency maximization problem of formula (17) is a limited-range optimal stopping problem which follows an optimal stopping rule (as shown in literature [4]). This paper solves the optimal stopping problem by inverse induction. And the sending node obtains an optimal transmission rate threshold when it stops competing for the channel and disseminates data in each period T.

4.3. Optimal Stopping Rule of Optimal Energy Efficiency. The sending node takes part in the channel competition in each period T, observes the channel quality after obtaining the channel, and then determines whether the current moment is the optimal energy efficiency moment for disseminating data. And the decision is made dependent on the expected value of the future channel quality. The sending node's reward for stopping channel competition and disseminating data is $\Phi_n(\chi) = R_n\tilde{t}_n - \chi(P\tilde{t}_n + E_{\text{total},n})$ when it successfully competes for the channel at time n. If it gives up disseminating data, the sending node will expect to receive reward.

$$E[\Phi_{n+1}(\chi) \mid F_n] = E\left[R_{n+1}\tilde{t}_{n+1} - \chi\left(P\tilde{t}_{n+1} + E_{\text{total},n+1}\right) \mid F_n\right]. \tag{18}$$

Here, $n = 1, 2, \cdots, M-1$ and F_n represents that the transmission rate sequence values $R_1, \cdots, R_n$. According to the optimal stopping rule [5, 12], if the actual reward $\Phi_N(\chi)$ of the data dissemination is equal to or greater than the expected reward $E[\Phi_N(\chi) \mid F_n]$ after the sending sensor achieves the channel competition success at time n, the data will be disseminated. Instead, the sending node gives up using the channel and continues to take part in the next competition round. Therefore, the condition that the sending node disseminates data after obtaining the channel at time n satisfies

$$\Phi_n(\chi) \geq E[\Phi_{n+1}(\chi) \mid F_n], n = 1, 2, \cdots, M-1. \tag{19}$$

Proposition 1. When $N^*(\chi) = \min\{M > n \geq 1 : \Phi_n(\chi) \geq E[\Phi_{n+1}(\chi)]|F_n]\}$ and the approximate stopping rule satisfies $R_n \geq R_{th,n}(\chi)$, the rule is optimal, where

$$R_{th,n}(\chi) = \begin{cases} \dfrac{\int_{R_{th,n+1}(\chi)}^{R_{\max}} \left((r - \chi P)\tilde{t}_{n+1}\right) dF_R(r) - \chi(E_{c,n+1} + P_0 T)}{T - b_1\tau} + \chi P, 1 \leq n < M, \\ 0, \quad n = M, \end{cases} \tag{20}$$

$$E_{c,n+1} = \begin{cases} \sum_{i=1}^{K} b_i \cdot p \cdot P \cdot \tau, 1 \leq n < M-1, \\ b_1 \cdot p \cdot P \cdot \tau, \quad n = M-1, \end{cases} \tag{21}$$

$$\tilde{t}_{n+1} = \begin{cases} Q_d/R_{n+1}, Q_d/R_{n+1} \leq T - b_1 \cdot \tau, \\ T - b_1 \cdot \tau, Q_d/R_{n+1} > T - b_1 \cdot \tau. \end{cases} \tag{22}$$

Proof. According to (15), there is

$$E[E_{c,n+1} \mid F_n] = \begin{cases} \sum_{i=1}^{K} b_i pP\tau, 1 \leq n < M-1, \\ b_1 pP\tau, \quad n = M-1, \end{cases} \tag{23}$$

$E[E_{c,n+1} \mid F_n]$ represents the expected competition energy consumption of the sending sensor at $n+1$. Its value is

$$E[E_{c,n+1} \mid F_n] = \begin{cases} \sum_{i=1}^{K} b_i pP\tau, \quad 1 \leq n < M-1, \\ b_1 pP\tau, \quad n = M-1. \end{cases} \tag{24}$$

Since the data has the maximum delay D_m, the sending node does not need to consider the transmission rate after successfully competing to the channel at time M, i.e., the transmission rate threshold of M is 0. Then there is

$$E\left[(R_M - \chi P)\tilde{t}_M \mid F_{M-1}\right] = \int_0^{R_{\max}} \left((r - \chi P)\tilde{t}_M\right) dF_R(r), \tag{25}$$

where $\tilde{t}_M$ is defined in (7). $F_R(r)$ is the cumulative dissemination function of the transmission rate, and $R_{\max}$ represents the maximum transmission rate. Therefore, the transmission rate of the data dissemination after the transmission node successfully competes for the channel at time $M-1$ satisfies

$$R_{M-1}\tilde{t}_{M-1} \geq \int_0^{R_{\max}} \left((r - \chi P)\tilde{t}_M\right) dF_R(r) - \chi(b_1 pP\tau + P_0 T) + \chi P\tilde{t}_{M-1}. \tag{26}$$

According to (7), we can see that $\tilde{t}_{M-1} \leq T - b_1\tau$.

As a result, the transmission rate threshold at which the sending node stops the channel competition and starts disseminating data at $M-1$ is

$$R_{th,M-1}(\chi) = \frac{\int_0^{R_{\max}} \left((r - \chi P)\tilde{t}_M\right) dF_R(r) - \chi(b_1 pP\tau + P_0 T)}{T - b_1\tau} + \chi P. \tag{27}$$

So there is

$$E\left[(R_{M-1} - \chi P)\tilde{t}_{M-1} \mid F_{M-2}\right] = \int_{R_{th,M}(\chi)}^{R_{\max}} \left((r - \chi P)\tilde{t}_{M-1}\right) dF_R(r). \tag{28}$$

$\tilde{t}_{M-1}$ is defined in (7). According to (19), the transmission rate threshold at which the sending node stops channel competition and starts to disseminate data at $M-2$ is

$$R_{th,M-2}(\chi) = \frac{\int_{R_{th,M-1}(\chi)}^{R_{\max}} \left((r-\chi P)\tilde{t}_{M-1}\right) dF_R(r) - \chi\left(\sum_{i=1}^{K} b_i pP\tau + P_0 T\right)}{T - b_1\tau} + \chi P. \tag{29}$$

Similarly, the transmission rate threshold obtained from $M-3$ to 1 is $R_{th,M-3}(\chi), \cdots, R_{th,1}(\chi)$.

In summary, the transmission rate threshold of the transmission sensor at time n $(1 \le n \le M)$ is defined as (20). When the transmission rate satisfies $R_n \ge R_{th,n}(\chi)$, (19) is monotonous. Therefore, the myopic stopping rule in Proposition 1 is optimal.

4.4. Optimal Energy Efficiency Data Dissemination Strategy Based on Optimal Stopping Theory-OEDDBOS. According to the optimal myopia stopping rule in Subsection 4.3, the optimal stopping rule for optimal stopping problem (16) is

$$N(\chi *) = \min\left\{M \ge n \ge 1 : R_n \ge R_{th,n}(\chi *)\right\}. \tag{30}$$

Here, $R_{th,n}(\chi *)$ is the optimal transmission rate threshold of the sending sensor at n, which is defined in (20). Next, we solve the optimal average energy efficiency $\chi *$ according to (13).

Given the cumulative dissemination function $F_G(g)$ of the random gain variable G of the channel, the cumulative dissemination function of the random variable R of the channel transmission rate is derived as $F_R(r)$. The probability that the sending node competes successfully to the channel is $1/K$ at time n. Therefore, the probability that the sending node disseminates data at time 1 is $(1-F_R(R_{th,1}(\chi *)))/K$, while the probability that data cannot be disseminated is $F_R(R_{th,1}(\chi *) + (1-F_R(R_{th,1}(\chi *)))(K-1)/K$. If the sending node does not disseminate data at time 1, then it continues to participate in channel competition at time 2, and so on. The probability that the sending node disseminates data at time n is

$$\theta_n = \left(\prod_{i=1}^{n-1}\left(F_R(R_{th,i}(\chi *)) + \frac{(1-F_R(R_{th,i}(\chi *))(K-1)}{K}\right)\right) \cdot \frac{(1-F_R(R_{th,n}(\chi *)))}{K}. \tag{31}$$

The transmission rate expected value and the expected time length of the sending node data dissemination at the time n are

$$\begin{gathered}\bar{R}_n = E[R_n] = \int_{R_{th,n}(\chi *)}^{R_{\max}} \frac{r}{(1-F_R(R_{th,n}(\chi *)))} dF_R(r), \\ \bar{t} = E\left[\tilde{t}_n\right] = \begin{cases} \dfrac{Q_d}{\bar{R}_n}, & \dfrac{Q_d}{\bar{R}_n} \le b_1\tau, \\ b_1\tau, & \dfrac{Q_d}{\bar{R}_n} > b_1\tau. \end{cases}\end{gathered} \tag{32}$$

The expected value of transmission data, the expected energy consumption, and the expected detection total energy consumption of the sending node from 1 to M are

$$E\left[R_N\tilde{t}_N\right] = \sum_{n=1}^{M} \bar{R}_n \bar{t}_n \theta_n, \tag{33}$$

$$E\left[P\tilde{t}_N\right] = \sum_{n=1}^{M} P\bar{t}_n \theta_n \tag{34}$$

$$E[E_{\text{total},N}] = \sum_{n=1}^{M}\left(\sum_{x=1}^{n} E_{c,x} + nP_0T\right) \cdot \theta_n, \tag{35}$$

where $E_{c,x}$ is defined in (7). Finally, the optimal average energy efficiency $\chi *$ is obtained by solving (13). The solution process for $\chi *$ can be described in detail as follows.

Step 1. Initialize the average energy efficiency χ_0.

Step 2. Calculate the transmission rate threshold $R_{th,M}$ (χ_0) in terms of formula (20), and let $n = M-1$.

Step 3. If $n \ge 1$, then run Step 4; otherwise, run Step 5.

Step 4. Calculate $R_{th,n}(\chi_0)$ in terms of formula (20), and let $n = n-1$, then return to Step 3.

Step 5. Calculate the expected value of transmission data $E[R_N\tilde{t}_N]$, the expected energy consumption $E[P\tilde{t}_N]$, and the expected detection total energy consumption of the sending node (N) $E[E_{\text{total},N}]$ in terms of formulas (33), (34), and (35). Calculateχ_{new} in terms of formula (13). If $|1/\chi_{\text{new}} - 1/\chi_0| > \varepsilon$ (a given error), then$\chi_0 = \chi_{\text{new}}$, return to Step 2; otherwise, $\chi * = \chi_{\text{new}}$.

Step 6. End.

Therefore, the flowchart of OEDDBOS is shown in Figure 3.

The above algorithm uses the Newton iteration method to calculate the optimal average energy efficiency $\chi *$ according to the given initial value χ_0. This iterative method quadratically convergences to the optimal value and the result tends to be stable after three times iterations. According to the optimal average energy efficiency $\chi *$, an optimal transmission rate threshold $R_{th,n}(\chi *)$ for the data dissemination is obtained after the sending sensor successfully competes for the channel at n in the optimal stopping rule (30).

The optimal rate threshold $R_{th,n}(\chi *)$ is a threshold for disseminating data after the sending sensor obtains a channel using opportunity at n. And it is also the rate threshold when the average value of the data dissemination per unit of energy consumption is the largest. This threshold controls effectively the timing at which the sending node disseminates data. The sending node participates in the channel competition in each round T. The optimal stopping rule strategy is to verify

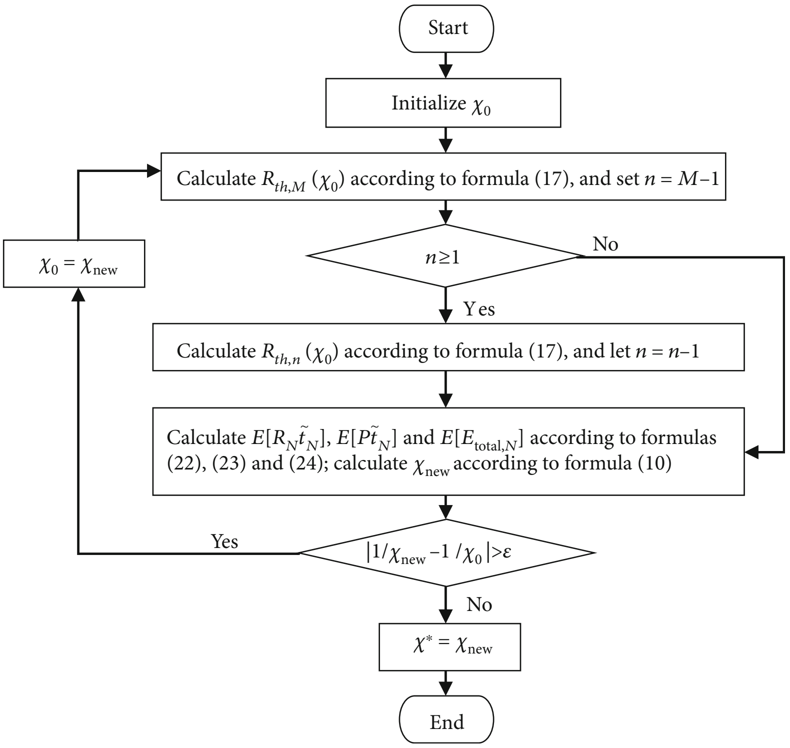

Start
Initialize χ_0
Calculate $R_{th,M}$ (χ_0) according to formula (17), and set $n = M-1$
$n\geq1$
No
Yes
$\chi_0 = \chi_{\text{new}}$
Calculate $R_{th,n}$ (χ_0) according to formula (17), and let $n = n-1$
Calculate $E[R_N\tilde{t}_N]$, $E[P\tilde{t}_N]$ and $E[E_{\text{total},N}]$ according to formulas (22), (23) and (24); calculate χ_{new} according to formula (10)
Yes
$|1/\chi_{\text{new}} -1 /\chi_0|>\varepsilon$
No
$\chi^* = \chi_{\text{new}}$
End

Figure 3: The flowchart of OEDDBOS.

whether the current transmission rate reaches the optimal transmission rate threshold $R_{th,n}(\chi *)$ after winning the channel competition. The sending node disseminates data if the current transmission rate value is greater than or equal to the optimal threshold. Otherwise, it abandons the use of the channel and continues the channel competition in the next round. The sending node performs continuously channel competition and data dissemination according to a given optimal stopping strategy and increases the average data volume per unit of energy consumption and the data transmission rate, so as to achieve the energy optimization effect under the premise of guaranteeing the data dissemination throughput.

5. Experimental Results and Analysis

This section compares the optimal energy efficiency data dissemination strategy based on optimal stopping theory proposed in this paper with other data dissemination strategies in related studies, then analyzes and evaluates the average energy efficiency and the average transmission rate of each strategy. The strategies being compared in this paper include the following three.

(1) Energy-Efficient Optimization for Distributed Opportunistic Scheduling Strategy (EEODOS). K sending nodes participate once in the channel competition with a probability $1/K$ every other period τ. And the competition is successful when only one node competes for the channel. Then, the node decides to disseminate data immediately or to abandon dissemination and continue to compete for the channel according to whether the current transmission rate is greater than or equal to the optimal transmission rate threshold. The duration of time that the sending node disseminates data is T_{trans}

(2) Energy Efficient Data Dissemination Strategy Based on Game Theory (EEDDBG). K sending nodes participate in channel competition for one time with an optimal probability every other period τ. And it is successful for competition when only one node competes for the channel. Then, the node disseminates data. Indeed, the duration of the dissemination depends on the size of the transmission rate and the capacity of the data packets to be distributed, where the optimal probability relies on the number K of nodes participating in the competition

(3) Randomly Competing Data Dissemination Strategy (Random). K sending nodes participate in a channel competition with a probability $1/K$ every other period τ. And the competition succeeds when only one node competes for a channel. Then, the node disseminates data in the remaining time of the holding time of the current channel quality. The duration of data dissemination depends on the transmission rate and the capacity of the data packets to be distributed. The fading of wireless channels is a small-scale fading. And its fading model is usually modeled as the Rayleigh distribution or the Rician distribution [20]. In simulation, the sending node and the receiving node have the same channel condition cumulative

Table 1: Simulation experiment parameter values.

Parameter	Description	Value
W	Bandwidth (Hz)	10^7
N_0	Noise power spectral density (W/Hz)	10^{-7}
σ^2	Channel gain variance correlation value	1
P	Transmission power (W)	0.1
g	Channel gain	$0 \sim 4$
A	Peak of main signal amplitude	1
K	Number of sending sensor (s)	5
T_{diss}	Cycle of base station dissemination data (s)	4
Q_d	Amount of data distributed by the base station at each time (kB)	144
D_m	The maximum delay of data (s)	10
T	Channel quality holding time (s)	$0.7 \times D_m$
τ	Channel competition period (s)	$0.001 \times T$
T_{trans}	Transmission time length of EEODOS (s)	$3 \times \tau$
Pe	Reference power consumption (W)	$0.1 \times P$

distribution function. According to the channel gain probability density function in [15], combined with the Shannon formula [(4)], the cumulative distribution functions of the transmission rate r against the maximum rate $R_{\max}$ under the Rayleigh and Rician distributions are as follows:

$$F_{\text{R\_Rayleigh}}(r) = \frac{\exp\left(-\left(\left(2^{r/W}-1\right)^2 \cdot (N_0 W)^2\right)/2\sigma^2 P^2\right)}{\exp\left(-\left(\left(2^{R_{\max}/W}-1\right)^2 \cdot (N_0 W)^2/2\sigma^2 P^2\right)\right)},$$

$$F_{\text{R\_Rayleigh}}(r) = \frac{Q_1\left(A/\sigma, \left(2^{r/W}-1\right)N_0 W/\sigma P\right)}{Q_1\left(A/\sigma, \left(2^{R_{\max}/W}-1\right)N_0 W/\sigma P\right)}, \tag{36}$$

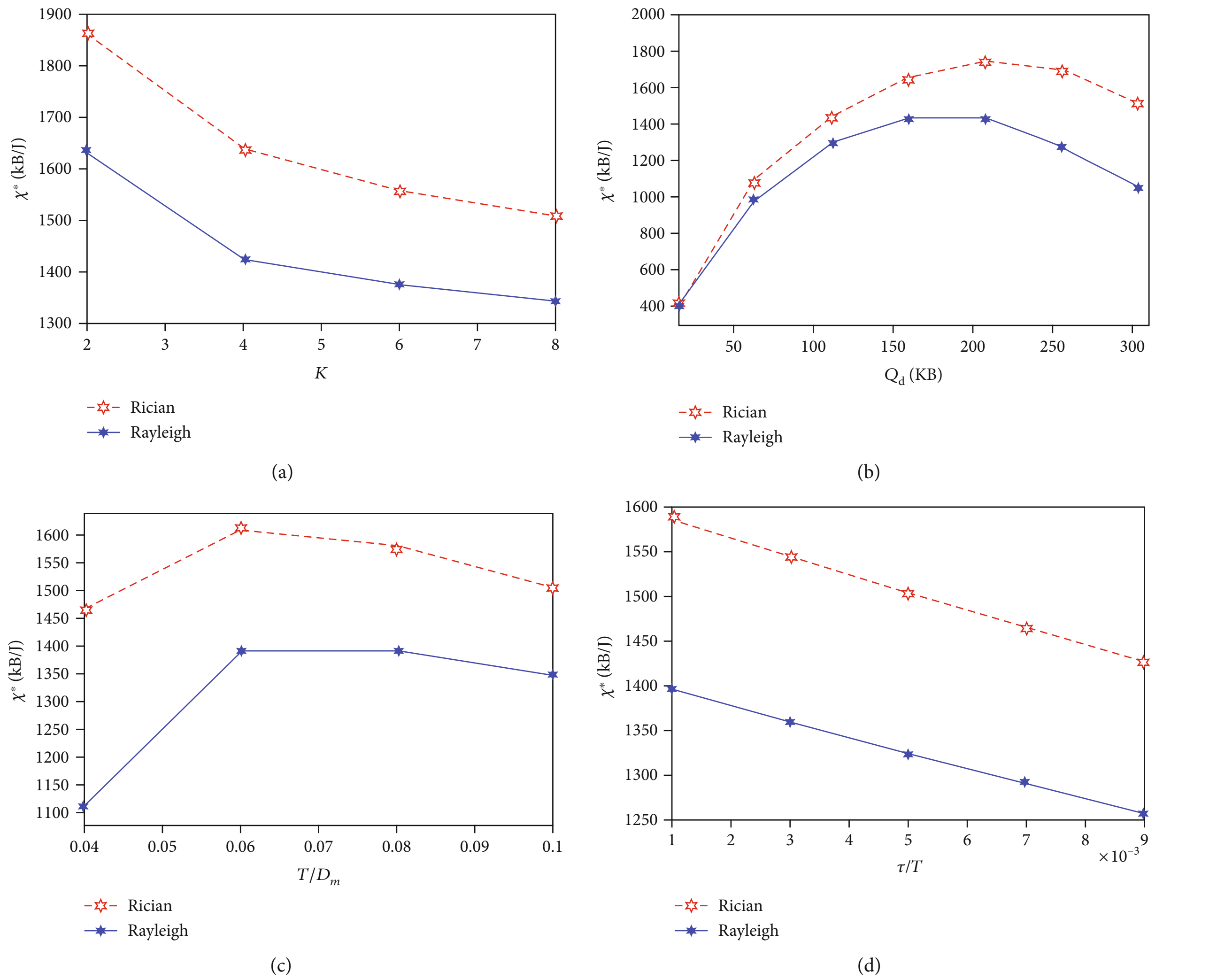
where σ^2 is the channel gain variance correlation value. W is the channel bandwidth. N_0 is the noise power spectral density. A is the peak of the main signal amplitude. And $Q_1(\bullet)$ is the first type of the Marcum Q-function. The value of each parameter in the simulation experiment is shown in Table 1.

According to the analysis in Section 4, the sending node's transmission rate threshold $R_{th,n}(\chi *)$ is closely related to the value of parameters such as the amount of data to be disseminated Q_d, the number of nodes of participating in competing K, the channel holding period T, the channel contention period τ, reference power consumption P_0, and so on. Figure 4 shows the relationship between the optimal average energy efficiency $\chi *$ and the parameters for the Rayleigh and Rician distributions.

First of all, when the amount of the data to be distributed Q_d is small, the ratio of the competition energy consumption to the transmission energy consumption of the transmission node is large, and the value $\chi *$ is small. If the value of Q_d is too large, the sending node continuously reduces the transmission rate threshold in order to ensure data transmission delay, reducing the value of $\chi *$. Secondly, as the number of nodes K increases, the values of the transmission rate threshold and $\chi *$ decrease. And then, as the channel retention period T is extended, the available transmission time after successful acquisition of the channel increases. And the ratio of the competition energy consumption to the transmission energy consumption within the period decreases, while the value of $\chi *$ increases. Next, the smaller the competition period τ, the lesser the competition energy consumption is and the greater the value of $\chi *$. Finally, P_0 is the continuous power consumption of the sending node in the open state. So the larger the value of P_0 is, the smaller the value of $\chi *$ is.

5.1. Average Energy Efficiency. Energy efficiency is the ratio of the total amount of data disseminated by the sending node to the total energy consumption. And the average energy efficiency refers to the average energy efficiency of each sending node, which represents the average amount of data dissemination per unit of energy consumption as well as the data dissemination efficiency by the unit energy consumption. The higher the average energy efficiency is, the greater amount of data dissemination per unit of energy consumption is achieved and the more energy is saved. Figure 5 compares the average energy efficiency of each strategy when the Rayleigh and Rician distributions have different parameters.

In Figure 5, OEDDBOS-1 shows the OEDDBOS situation under the Rician distribution, as OEDDBOS-2 is under the Rayleigh distribution. And the other three strategies are the same. From Figure 5, OEDDBOS achieves the greatest average energy efficiency. That is, its saving effect is the best. OEDDBOS obtains the optimal transmission rate threshold of the sending node by taking advantage of the optimal stopping theory at each period T, which is based on the amount of data to be distributed Q_d, the number of sensors participating in the competition K, the channel holding period T, the channel competition period τ, and the value of the reference power consumption P_0. It is proven that the highest average energy efficiency can be achieved with this threshold. Based on the game theory, EEDDBG obtains the optimal probability of the sending node participating in the competition. This strategy aims to reduce the conflict energy consumption of the channel competition. However, it does not take into account the random variation of channel quality and its average energy efficiency is related to the probability distribution of channel transmission rate. Random has neither obtained the optimal probability of participating in the competition, nor selected the transmission rate when the channel quality is in good condition. In addition, its average energy efficiency is comparatively lower. On the basis of the optimal stopping theory, EEODOS obtains the optimal transmission rate threshold. However, the maximum transmission delay of the data is not considered at the threshold, resulting in some data being discarded because of exceeding the deadline and thereby lacking the transmittable data when there is a higher transmission rate threshold. EEODOS does not also consider the length of the channel quality holding time. The sending node participates in the channel competition all the time after the data transmission is completed if



(a)

(b)

(c)

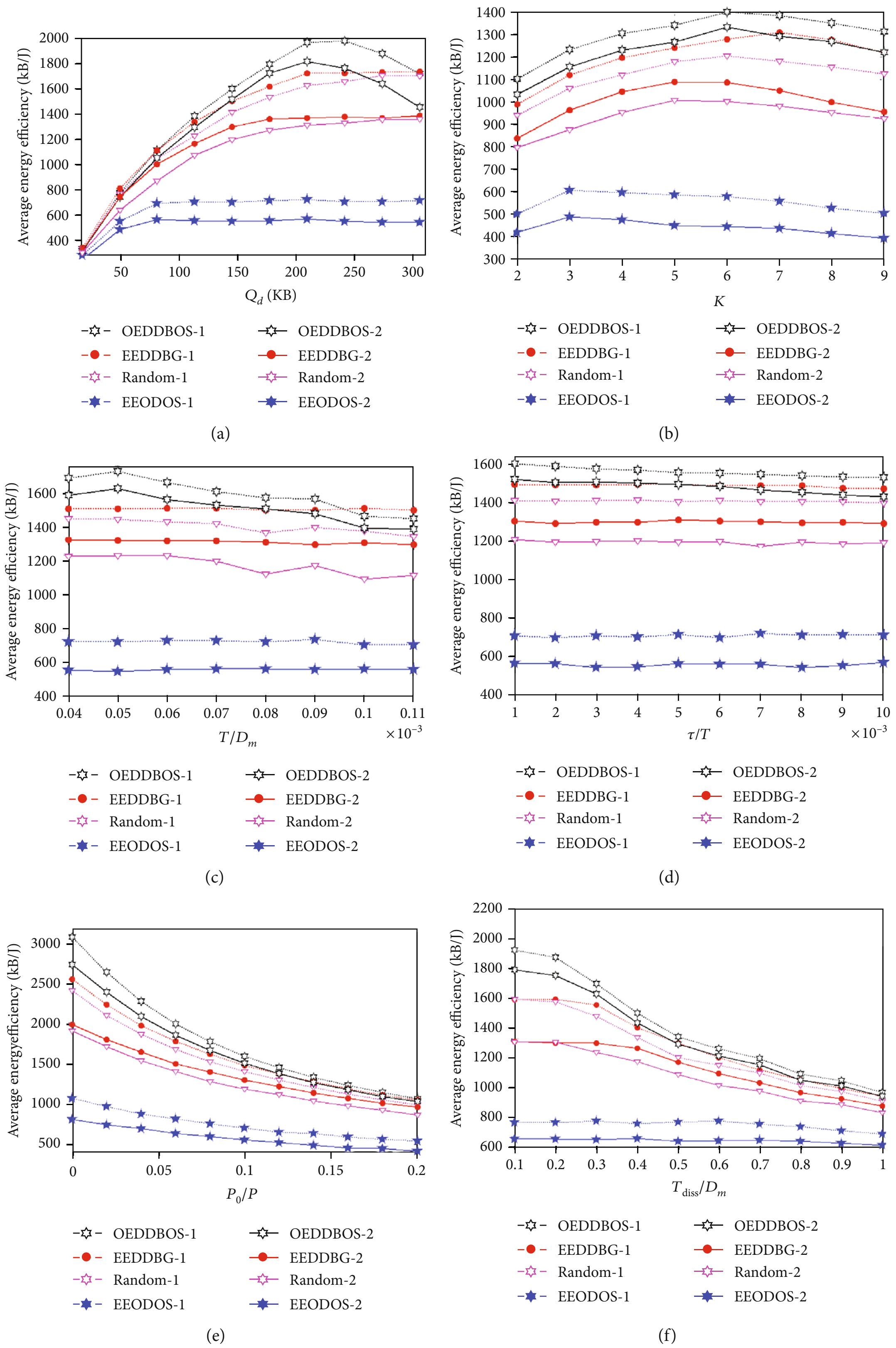
(d)

FIGURE 4: The relationship between the optimal average energy efficiency $\chi *$ and the parameters. (a) The relationship between $\chi *$ and K. (b) The relationship between $\chi *$ and Q_d. (c) The relationship between $\chi *$ and T. (d) The relationship between $\chi *$ and τ.

the transmission rate is not reaching to the threshold. And its competitive energy consumption greatly increased. Meanwhile, the time length of the data transmission is given, which will result in the data transmission rate changing during the dissemination process, affecting the average energy efficiency.

5.2. Average Transmission Rate. The transmission rate is the ratio of the amount of data being successfully disseminated by the sending node to the amount of data to be disseminated. Because data has the maximum delay requirement, the data that exceeds the delay will be discarded. The average transmission rate is the average of the transmission rate of each transmission node. It represents the proportion of the data that is disseminated within the delay requirement in the network. And the average transmission rate is also the probability of successful data dissemination. The greater the average transmission rate, the less data discarded due to timeout. Figure 6 shows the mean comparison of the average transmissibility for each strategy as the Rayleigh and Rician distributions have different parameters.

The description of Figure 6 has the same meaning as Figure 5. The mean value refers to the average value of the average transmission rate when the parameter values are within a certain range. The value ranges for each parameter are the same as shown in Figure 5. From Figure 6, OEDDBOS has a higher average transmission rate. The OEDDBOS selects the time when the channel quality is good (i.e., the transmission rate is high) to transmit data according to the amount of data to be disseminated Q_d and to reduce the amount of data that is discarded due to the delay. As long as the sending node has the amount of data to be disseminated, it will participate in the channel competition and disseminate the data after obtaining the channel, thereby increasing the transmission rate. In Random, the sending node participates in the channel competition with a period of T. The sending node which obtains the channel occupies the remaining time of T, regardless of the amount of data to be disseminated. This will cause some data to be discarded before getting a transmission opportunity due to beyond the delay. The average transmission rate when the number of



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5: The comparison of the average energy efficiency of different strategy. (a) Average energy efficiency of different quantities of data distributed Q_d. (b) Average energy efficiency of different numbers of sending nodes K. (c) Average energy efficiency of different channel holding periods T. (d) Average energy efficiency of different channel competition periods τ. (e) Average energy efficiency of different reference power consumptions P_0. (f) Average energy efficiency of different cycles to distributed data T_{diss}.

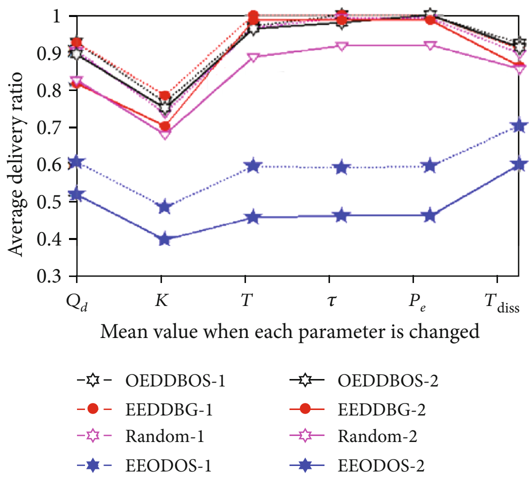


Figure 6: The mean comparison of the average transmissibility for five strategies.

sending node K changes is the smallest. Because when the K increases, the transmission opportunity of each transmission node decreases, thus the transmission rate decreases. The average transmission mean value is smaller when the data volume Q_d is variable. As Q_d keeps increasing, the amount of data that can be disseminated by each sending sensor is limited. And some data are discarded due to excessive delays, resulting in a decrease in the transmission rate. The change in the channel holding period T, the competition period τ, and the reference power consumption P_0 does not affect the mean value of the average transmission rate but mainly affects the average energy efficiency. The average transmission rate is small when the data dissemination period T_{diss} changes. Because T_{diss} keeps declining, the amount of data to be disseminated continues to increase, and the amount of data that is discarded beyond the delay increases, thus the transmission rate decreases.

6. Conclusion

In sensor-cloud systems, multiple sensors/sinks usually use the same channel to transmit data at the same time in the transmission interference range, which will bring transmission collisions and result in data transmission failure. Aiming at reducing energy consumption for data dissemination in sensor-cloud systems, this paper mainly focuses on the energy consumption optimization problem with time-varying channel quality when multiple sensors use the same channel to disseminate data. The sensor sends data with a certain probability to compete for channel. The sensor decides whether or not to distribute data in terms of channel quality for saving energy after getting channel successfully. We construct the maximization problem of average energy efficiency for distributing data with delay demand. Then, this maximization problem is transferred to an optimal stopping problem which generates the optimal stopping rule. At last, the thresholds of the optimal transmission rate in each period are solved by using the optimal stopping theory, and the optimal energy efficiency for data distribution is achieved. Simulation results indicate that the strategy proposed in this paper can to some extent improve average energy efficiency and delivery ratio and enhance energy optimization effect and network performance compared with other strategies.

Data Availability

The figure's data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This study was funded by the National Natural Science Foundation of China under Grant Nos. 61562006 and 61772233 and in part by the Natural Science Foundation of Guangxi Province under Grant Nos. 2013GXNSFGA019006 and 2016GXNSFBA380181.

References

[1] A. Alamri, W. S. Ansari, M. M. Hassan, M. S. Hossain, A. Alelaiwi, and M. A. Hossain, "A survey on sensor-cloud: architecture, applications, and approaches," *International Journal of Distributed Sensor Networks*, vol. 9, no. 2, 2013.

[2] Z. Jiandian, T. Wang, J. Weijia, P. Shaoliang, and G. Wang, "A survey on Sensor-Cloud," *Journal of Computer Research and Development*, vol. 54, no. 5, pp. 925–939, 2017.

[3] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, "A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4831–4843, 2019.

[4] T. Wang, J. Zhou, A. Liu, M. Z. A. Bhuiyan, G. Wang, and W. Jia, "Fog-based computing and storage offloading for data synchronization in IoT," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4272–4282, 2019.

[5] T. Wang, L. Yang, W. Fang et al., "A comprehensive trustworthy data collection approach in sensor-cloud systems," *IEEE Transactions on Big Data*, p. 1, 2019.

[6] M. Yuriyama and T. Kushida, "Sensor-cloud infrastructure - physical sensor management with virtualized sensors on cloud computing," in *2010 13th International Conference on Network-Based Information Systems*, pp. 1–8, Takayama, Japan, 2010.

[7] A. Garcia-Saavedra, P. Serrano, and A. Banchs, "Energy-efficient optimization for distributed opportunistic scheduling," *IEEE Communications Letters*, vol. 18, no. 6, pp. 1083–1086, 2014.

[8] H. Chen and J. S. Baras, "Distributed opportunistic scheduling for wireless ad-hoc networks with block-fading model," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 11, pp. 2324–2337, 2013.

[9] A. Antonopoulos and C. Verikoukis, "Multi-player game theoretic MAC strategies for energy efficient data dissemination," *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 592–603, 2014.

[10] Y. Chow, H. Robbins, and D. Siegmund, *Great Expectations: Theory of Optimal Stopping*, Houghton Mifflin, Boston, MA, 1971.

[11] T. Ferguson, *Optimal stopping and applications*, Math. Dept., Univ. Calif, Los Angeles, CA, USA, 2006, https://www.math.ucla.edu/~tom/Stopping/Contents.html.

[12] P. Boonma and J. Suzuki, "Toward interoperable publish/subscribe communication between wireless sensor networks and access networks," in *2009 6th IEEE Consumer Communications and Networking Conference*, pp. 1–6, Las Vegas, NV, USA, 2009.

[13] Y. Xu, S. Helal, M. Thai, and M. Scmalz, "Optimizing push/-pull envelopes for energy-efficient cloud-sensor systems," in *Proceedings of the 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems - MSWiM '11*, pp. 17–26, New York, NY, USA, 2011.

[14] D. H. Phan, J. Suzuki, S. Omura, and K. Oba, "Toward sensor-cloud integration as a service: Optimizing three-tier communication in cloud integrated sensor networks," in *Proceedings of the 8th International Conference on Body Area Networks*, pp. 355–362, Brussels, Belgium, 2013.

[15] L. B. Saad and B. Tourancheau, "Lifetime optimization of sensor-cloud systems," in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, Paris, 2015.

[16] C. Pin-Yu, S.-M. Cheng, and C. Kwang-Cheng, "Optimal control of epidemic information dissemination over networks," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2316–2328, 2014.

[17] D. Zheng, W. Ge, and J. Zhang, "Distributed opportunistic scheduling for ad hoc networks with random access: an optimal stopping approach," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 205–222, 2009.

[18] G. Wang, Y. Peng, and Q. Zhao, "Optimal energy efficiency data dissemination strategy based on optimal stopping theory in mobile network," in *Computational Data and Social Networks*, vol. 11280 of Lecture Notes in Computer Science, , Springer.

[19] Y. Peng, G. Wang, and N. Wang, "Energy-efficient transmission strategy by using optimal stopping approach for mobile networks," *Mobile Information Systems*, vol. 2016, Article ID 8981251, 16 pages, 2016.

[20] M. I. Poulakis, A. D. Panagopoulos, and P. Constantinou, "Channel-aware opportunistic transmission scheduling for energy-efficient wireless links," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 192–204, 2013.