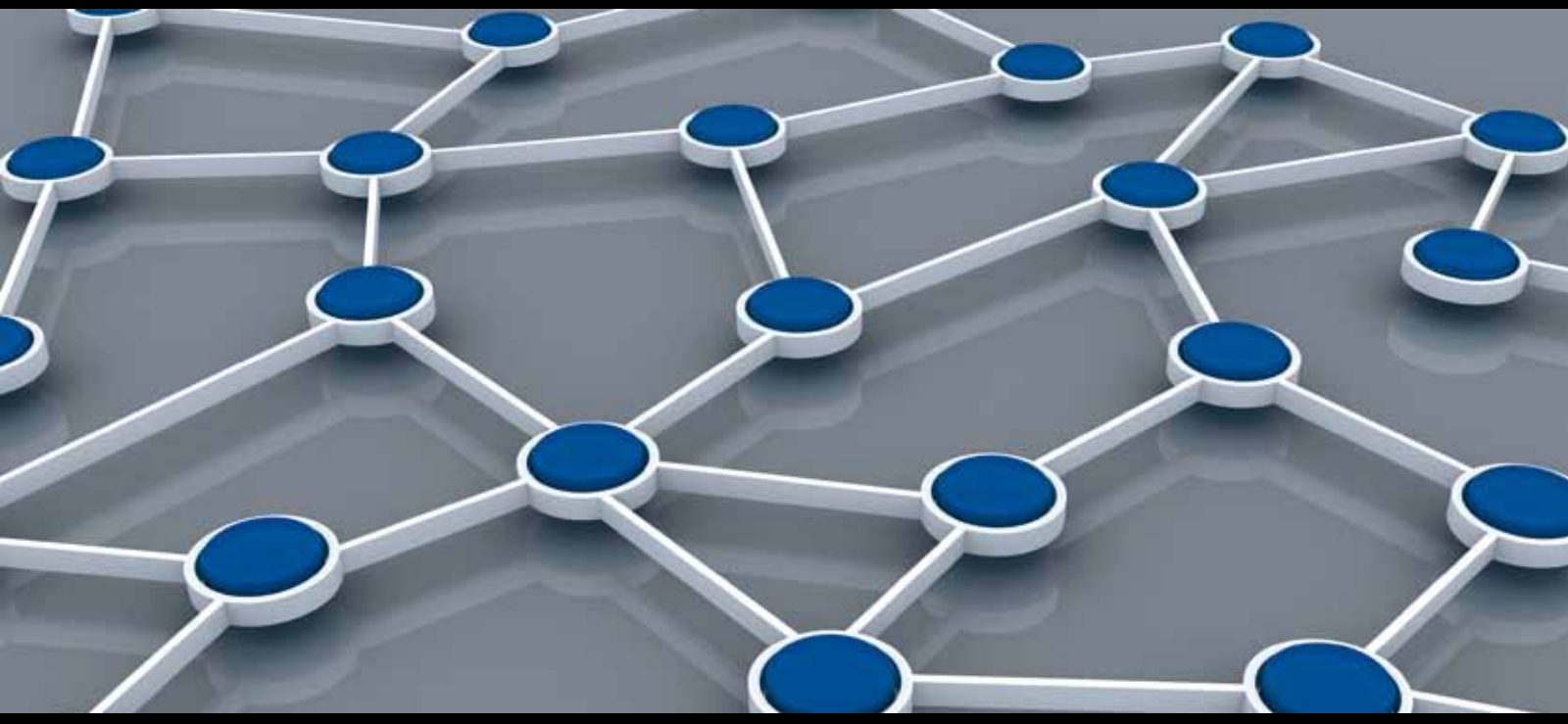
International Journal of Distributed Sensor Networks

Mobile Sensing and Data Management for Sensor Networks

Guest Editors: Jianwei Niu, Lei Shu, Zhangbing Zhou, and Yan Zhang



Mobile Sensing and Data Management for Sensor Networks

International Journal of Distributed Sensor Networks

Mobile Sensing and Data Management for Sensor Networks

Guest Editors: Jianwei Niu, Lei Shu, Zhangbing Zhou, and Yan Zhang

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 898169, 3 pages
http://dx.doi.org/10.1155/2013/898169

Editorial
Mobile Sensing and Data Management for Sensor Networks

Jianwei Niu,[1] Lei Shu,[2] Zhangbing Zhou,[3,4] and Yan Zhang[5]

[1] *State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China*
[2] *Guangdong University of Petrochemical Technology, Maoming 525000, China*
[3] *China University of Geosciences, Beijing 100191, China*
[4] *Institut Télécom, France*
[5] *Simula Research Laboratory, University of Oslo, 1325 Lysaker, Norway*

Correspondence should be addressed to Jianwei Niu; niujianwei@buaa.edu.cn

Received 14 August 2013; Accepted 14 August 2013

With the rapid advent of the Internet of Things, sensor cloud, mobile Internet, and Web 3.0, more and more mobile devices, such as smart phones, Google glasses, and RFID, plus deployed various sensor networks, can sense and collect sensory data anytime and anywhere. We are moving toward the era of worldwide sensor networks, in which a huge amount of heterogeneous sensory data will be created every day and require advanced data management. In this setting, efficiently gathering, sharing, and integrating these spatial temporal data, and then deriving valuable knowledge timely, are a big challenge in this context. Furthermore, in the mobile environment, data management means a collection of centralized and distributed algorithms, architectures, and systems to store, process, and analyze the immense amount of spatial temporal data, where these data are cooperatively gathered through collections of mobile sensing devices which move in space over time. This special issue on mobile sensing and data management for sensor networks is intended to provide a forum for presenting, exchanging, and discussing the most recent advances in sensing and data management techniques.

To prolong the life time of each node in MSNs, energy model and conservation should be considered carefully when designing the data communication mechanism. The limited battery volume and high workload on channels worsen the life times of the busy nodes. In the paper "*An probability-based energy model on cache coherence protocol with mobile sensor network*," the authors propose a new energy evaluating methodology of packet transmissions in MSNs, which is based on redividing network layers and describing the synchronous data flow with matrix form.

Mobile cloud computing (MCC) enables mobile devices to outsource their computing, storage, and other tasks onto the cloud to achieve more capacities and higher performance. In the paper "*Adaptive computing resource allocation for mobile cloud computing*," the authors propose a novel MCC adaptive resource allocation model to achieve the optimal resource allocation in terms of the maximal overall system reward by considering both cloud and mobile devices. The adaptive resource allocation is modeled as a semi-Markov decision process (SMDP) to capture the dynamic arrivals and departures of resource requests.

Computation offloading is a popular approach for reducing energy consumption of mobile devices by offloading computation to remote servers. The paper "*An energy-efficient multisite offloading algorithm for mobile devices*" proposes an Energy-Efficient Multisite Offloading (EMSO) algorithm. It formulates the multiway partitioning problem as the 0-1 integer linear programming (ILP) problem, which is solved through the proposed EMSO algorithm adopting the multiway graph partitioning-based technique.

The mixed wireless sensor networks that are composed of a mixture of mobile and static sensors are the tradeoff between cost and coverage. To provide the required high coverage, the mobile sensors have to move from dense areas to sparse areas. The paper "*An energy-efficient motion strategy for mobile sensors in mixed wireless sensor networks*" presents a centralized algorithm to assist the movement of mobile sensors. The management node of the WSN collects

the geographical information of all of the static and mobile sensors. The management node executes the algorithm to get the best matches between mobile sensors and coverage holes.

With the advance of embedded sensing devices, pervasive urban sensing (PUS) with probe vehicles is becoming increasingly practical. A probe vehicle is equipped with onboard sensing devices that detect urban information as the probe vehicle drive across the road network. In the paper "*Pervasive urban sensing with large-scale mobile probe vehicles*," the authors present the framework of Pervasive Urban Sensing with probe vehicles, and showcase two cases of urban sensing with probe vehicles.

Rotating machinery is widely used in modern industry. It is one of the most critical components in a variety of machinery and equipment. Along with the continuous development of science and technology, the structures of rotating machinery become of larger scale, of higher speed, and more complicated, which results in higher probability of concurrent failure in practice. In the paper "*Concurrent fault diagnosis for rotating machinery based on vibration sensors*", the authors develop an integrated method using artificial immune algorithm and evidential theory to achieve concurrent fault diagnosis for rotating machinery.

The paper "*Enhanced mobile multiple-input multiple-output underwater acoustic communications*" focuses on mobile multiple-input multiple-output (MIMO) underwater acoustic communications (UAC) over double-selective channels subject to both intersymbol interference and Doppler scaling effects. Under the assumption that the channels between all the transmitter and receiver pairs experience the same Doppler frequency, a variation of the recently proposed generalization of the sparse learning via iterative minimization (GoSLIM) algorithm is employed to estimate the frequency modulated acoustic channels.

Often, a large number of wireless sensor nodes are deployed to detect target signal that is more accurate than the traditional single radar detection method. Each local sensor detects the target signal in the region of interests and collects relevant data, and it sends the respective data to the data fusion center (DFC) for aggregation processing and judgment making whether the target signal exists or not. The paper "*Weight-based clustering decision fusion algorithm for distributed target detection in wireless sensor networks*" proposes a novel Weight-based Clustering Decision Fusion Algorithm (W-CDFA) to detect target signal in wireless sensor networks.

Sensor network positioning systems have been extensively studied recently. How to acquire the anchor's position is a challenge. To address this issue, in the paper "*Efficient deterministic anchor deployment for sensor network positioning*," the authors design an efficient mapping algorithm between anchors and their positions (MD-SKM) to avoid the complicated artificial calibration and propose a best feature matching (BFM) method to further relax the restriction of MD-SKM where three or more calibrated anchors are needed.

In vehicular networks, the multihop message delivery from information source to moving vehicles presents a challenging task due to many factors, including high mobility, frequent disconnection, and real-time requirement for applications. In the paper "*Moving target oriented opportunistic routing algorithm in vehicular networks*," the authors propose a moving target oriented opportunistic routing algorithm in vehicular networks for message delivery from information source to a moving target vehicle. In order to adapt the constantly changing topology of networks, the forwarding decisions are made locally by each intermediate vehicle based on the trajectory information of the target vehicle.

With the increase of the storage capacity, computing, and wireless networking of the vehicular embedded devices, the vehicular networks bring a potential to enable new applications for drivers and passengers in the vehicles. In the paper "*RoadGate: mobility-centric roadside units deployment for vehicular networks*," the authors study the problem of deploying the RSUs to provide the desired connectivity performance while minimizing the number of the deployed RSUs. Besides, the authors analyze a realistic vehicle trace, observe the mobility pattern, and propose a graph model to characterize it. Based on the graph model, the gateway deployment problem is transformed into a vertex selection problem in a graph. A heuristic algorithm *RoadGate* is then proposed to search greedily the optimal positions.

Optimizing the path planning to reduce the time and cost is an essential consideration in modern society. Using dynamic path planning to adjust and update the path information in time is a challenging approach to reduce road congestion and traffic accidents. In the paper "*Data processing and algorithm analysis of vehicle path planning based on wireless sensor network*," the authors present a data analysis algorithm that determines an efficient dynamic path for vehicle repair-scrap sites and navigates more flexibly to avoid obstacles. The key idea is to design the wireless sensor network that helps to obtain data from different devices.

The paper "*Trajectory-based optimal area forwarding for infrastructure-to-vehicle data delivery with partial deployment of stationary nodes*" proposes a trajectory-based optimal area forwarding (TOAF) algorithm tailored for multihop data delivery from infrastructure nodes (e.g., Internet access points) to moving vehicles (infrastructure-to-vehicle) in vehicular ad hoc networks (VANETs) with partial deployment of stationary nodes. It focuses on reducing the delivery-delay jitter and improving the low reliability of infrastructure-to-vehicle communication.

The design and analysis of routing algorithms are important issues in WSNs. In the paper "*An overlapping clustering approach for routing in wireless sensor networks*," the authors propose a k-connected overlapping clustering approach with energy awareness, namely, k-OCHE, for routing in WSNs. The basic idea of this approach is to select a cluster head by energy availability (EA) status. The k-OCHE scheme adopts a sleep scheduling strategy of CKN, where neighbors will remain awake to keep it k-connected, so that it can balance energy distributions well.

WSNs are important parts of Internet of Things or cyber-physical systems. Data query processing is very important for WSNs. In the paper "*Continuous top-k contour regions querying in sensor networks*", the authors propose a Continuous Top-k Contour Regions Querying algorithm which can

continuously obtain the top-k contour regions and does not lose the rate of precision. This technique takes full advantage of the kth value of top-k result in current round as the threshold to suppress the nodes whose readings do not belong to the top-k result in next round.

In WSNs, homogeneous or heterogeneous sensor nodes are deployed at a certain area to monitor our curious target. The sensor nodes report their observations to the base station (BS), and the BS should implement the parameter estimation with sensors' data. Best linear unbiased estimation (BLUE) is a common estimator in the parameter estimation. In some soft real-time applications, we expect that the estimation can be completed before the deadline with a probability. In the paper "*Energy-efficient soft real-time scheduling for parameter estimation in WSNs*," the authors proposed an energy-efficient scheduling algorithm especially for the soft real-time estimations in WSNs. Through the proper assignment of sensors' state, an energy-efficient estimation is achieved before the deadline with a probability.

WSNs have limited energy and transmission capacity, so data compression techniques have extensive applications. For multivariate stream on a sensor node, some data streams are elected as the base functions according to the correlation coefficient matrix, and the other streams from the same node can be expressed in relation to one of these base functions using linear regression. By designing an incremental algorithm for computing regression coefficients, in the paper "*A self-adaptive regression-based multivariate data compression scheme with error bound in wireless sensor networks*," the authors propose a multivariate data compression scheme based on self-adaptive regression with infinite norm error bound.

The drive-thru Internet is an effective mean to provide Internet access service for WSNs deployed on vehicles. In these networks, vehicles often experience different link qualities due to different relative positions to the access point. This makes fair and efficient system design a very challenging task. In the paper "*Amortized fairness for drive-thru internet*," the authors propose a novel amortized fairness MAC protocol. Basically, vehicles with lower link quality can defer their fairness requests and let the lost fairness be "amortized" in the future when their links become the high quality. The inner and inter-AP correlations revealed from our extensive field studies are fully exploited, and a link quality prediction algorithm is proposed. Based on the predicted link quality, the optimal amortized fairness MAC is formulated as a convex programming problem, which can be solved with the desired precision in polynomial time.

Jianwei Niu
Lei Shu
Zhangbing Zhou
Yan Zhang

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 512368, 10 pages
http://dx.doi.org/10.1155/2013/512368

Research Article

Spatial TinyDB: A Spatial Sensor Database System for the USN Environment

Dong-Oh Kim,[1] Lei Liu,[2] In-Su Shin,[3] Jeong-Joon Kim,[3] and Ki-Joon Han[3]

[1] *Cloud Computing Research Department, Electronics and Telecommunications Research Institute, Daejeon 305-700, Republic of Korea*
[2] *Domestic Financial Department, China Banking Regulatory Commission Henan Office, 6 Cuizhu Street, High & New Technology Industries Development Zone, Zhengzhou 450000, China*
[3] *Division of Computer Science & Engineering, Konkuk University, Seoul 143-701, Republic of Korea*

Correspondence should be addressed to Jeong-Joon Kim; jjkim9@db.konkuk.ac.kr

Received 13 January 2013; Accepted 19 July 2013

Academic Editor: Lei Shu

For the Ubiquitous Sensor Network (USN) environment, which generally uses spatial as well as aspatial sensor data, a sensor database system to manage these data is essential. For this reason, sensor database systems such as TinyDB and Cougar are being developed by researchers. However, as most of these systems do not support spatial data types and spatial operators for managing spatial sensor data, they are not suitable for the USN environment. Therefore, in this paper, we design and implement Spatial TinyDB which is a spatial sensor database system that extends TinyDB to support spatial data types and spatial operators for the efficient management of spatial sensor data. In particular, Spatial TinyDB provides memory management and filtering functions to reduce system overload caused by sensor data streams. Finally, we prove that Spatial TinyDB is superior by comparing its actual performance, in terms of execution time, accuracy, and memory usage, with that of TinyDB.

1. Introduction

With the development of sensor technologies for sensing various types of data (such as temperature, humidity, and pressure) and with advances in wireless communication technologies (resulting in technologies such as CDMA, WiFi, and WiBro), there is increasing interest in, and research on, the application of technologies related to ubiquitous sensor networks (USNs) in areas such as ecosystem monitoring, home automation, and car theft detection [1].

A USN is a communication network in which various types of sensor nodes interconnected by means of wireless communication schemes manage sensed data [2, 3]. A sensor node consists of sensing, processing, storage, and communication modules. However, it has limited hardware and software capacities. That is, it has limitations with regard to its capacity to process sensed data, the space available to store sensed data, the distance of data transmission, and the amount of electric power available. In particular, the sensor node consumes more power for data transmission than for data processing.

A geosensor can obtain its location, either directly or indirectly, via RFID readers, GPSs, CCTVs, and so forth, and can generate various forms of related stream data [4]. The use of geosensors is increasing, particularly in the USN environment, as they are utilized in the provision of diverse services related to u-GIS, u-LBS, u-Logistics, u-Transportation, u-Medicine, u-Disaster Prevention, and so forth. In this sense, geosensors are leading the ubiquitous age using both aspatial and spatial data simultaneously. Accordingly, there is active research into the efficient management of spatial sensor data collected by geosensors in the USN environment [5].

Recently, various sensor database systems, including TinyDB [6] and Cougar [7], have been developed for the efficient management of sensor data in the USN environment due to the fact that existing sensor database systems do not support spatial data types and spatial operators. However, these recently developed systems cannot efficiently manage spatial sensor data from geosensors. In addition, SE TinyDB [8]—a spatial extension of TinyDB—provides its own spatial operators but cannot support the international standards recommended by the Open Geospatial Consortium (OGC).

In this paper, we design and implement Spatial TinyDB which is a spatial sensor database system that provides various spatial data types and spatial operators for the efficient management of spatial sensor data in the USN environment. In particular, Spatial TinyDB extends TinyDB—an existing sensor database system—to facilitate the efficient management of spatial sensor data. It also conforms to the Simple Feature Specification for SQL [9], a standard proposed by the OGC, in extending spatial data types and spatial operators for interoperability.

This paper is organized as follows. Following the introduction given in Section 1, Section 2 analyzes related studies on the TinyDB and SE TinyDB sensor database systems. Section 3 describes the overall structure of Spatial TinyDB and outlines each of the managers used in Spatial TinyDB. Section 4 verifies the superiority of Spatial TinyDB by means of performance evaluations. Finally, we give concluding remarks in Section 5.

2. Related Works

In this section we analyze TinyDB (used in the implementation of Spatial TinyDB proposed in this paper) and look at SE TinyDB—a spatial extension of TinyDB.

2.1. TinyDB. TinyDB [6, 10] is a query processing system used in TinyOS to extract information from wireless sensor networks. TinyOS [11]—an open source system developed at the University of California Berkeley using nesC language—is the most widely used representative sensor operating system. It modularizes the system into component units and each component is connected to other components through function calls known as interfaces. Thus, application developers can develop applications using components as libraries and connect the components through interfaces.

The characteristics of TinyDB are as follows. First, it provides a metadata catalog for describing the types of sensors in a sensor network. Second, it supports a query language that easily describes the data desired by users. Third, it can form a network by itself for query processing. Fourth, it can process multiple queries for multiple sensor nodes simultaneously. Fifth, a TinyDB sensor network can be extended by simply downloading a standard TinyDB code onto a new node and resetting the node.

Query processing in TinyDB is as follows. First, an input query from the server PC is transmitted to the network as an optimized query. The node that receives the transmitted query then acquires data from its neighboring nodes based on a routing tree and executes aggregate operations within the network. Finally, the result is returned to the server PC.

In order to enable users to extract the data desired without doing any programming, TinyDB provides a simple interface that is similar to SQL. Table 1 shows the query statement format supporting the SQL-like interface and a typical query example that can be used in TinyDB.

As shown in Table 1, clauses SELECT, WHERE, GROUP BY, and HAVING in the query statement format are similar to those of standard SQL. TRIGGER ACTION executes triggered actions such as SOUND and LED when the conditions of the WHERE statement are satisfied. EPOCH DURATION is used to set the time interval between epochs, in millisecond (ms). The example query in Table 1 gets the mean temperature of sensor nodes whose temperatures exceed 100°C. Table 2 lists the data types and operators supported by TinyDB.

As can be deduced from Table 2, TinyDB does not support spatial data types and spatial operators. As a result, it has difficulty in managing spatial data efficiently.

Table 1: Query statement format and query example.

Query statement format	SELECT select-list ⟨FROM sensors⟩ WHERE where-clause ⟨GROUP BY gb-list⟩⟨HAVING having-list⟩ ⟨TRIGGER ACTION command-name ⟨(param)⟩⟩ ⟨EPOCH DURATION⟩
Query example	SELECT AVG (temp) FROM sensors WHERE temp > 100

Table 2: Data types and operators.

Data types	Operators
int8, int16, int32, uint8, uint16, uint32, string	SUM, AVERAGE, MIN, MAX, COUNT

Table 3: Spatial operators in SE TinyDB.

Operators	Explanation
DISTANCE	DISTANCE [ID, X, Y]
INBOX	INBOX [$X_{\min}$, $Y_{\min}$, $X_{\max}$, $Y_{\max}$]
BEYONDBOUNDARY	BEYONDBOUNDARY [$X_{\min}$, $Y_{\min}$, $X_{\max}$, $Y_{\max}$, CMD (par)]

2.2. SE TinyDB. SE TinyDB [8] was designed and developed to process both spatial and aspatial queries in sensor nodes. SE TinyDB extends traditional TinyDB by adding the spatial operators DISTANCE, INBOX, and BEYONDBOUNDARY. Table 3 lists the spatial operators used to extend TinyDB to develop SE TinyDB.

In Table 3, the DISTANCE operator returns the coordinate distance between sensor nodes, while the INBOX operator returns the location of the sensor nodes within a specified rectangle. The BEYONDBOUNDARY operator returns the location of sensor nodes outside a specified rectangle.

Table 4 gives examples of spatial queries that use the spatial operators in SE TinyDB.

In Table 4, Example 1 returns the ID (nodeid), temperature (temp), and location coordinates (lat, lon) of those sensor nodes less than 200 units away from a specified point (Point (500, 500)). Example 2 returns the ID (nodeid), temperature (temp), and location coordinates (lat, lon) of those sensor nodes located within a specified rectangle (0, 0, 500, 500). Finally, Example 3 returns the ID (nodeid), temperature

Table 4: Examples of spatial query in SE TinyDB.

Example 1	SELECT nodeid, temp, lat, lon FROM sensors WHERE DISTANCE [nodeid, 500, 500] < 200
Example 2	SELECT nodeid, temp, lat, lon FROM sensors WHERE INBOX [0, 0, 500, 500]
Example 3	SELECT nodeid, temp, lat, lon FROM sensors WHERE BEYONDBOUNDARY [0, 0, 500, 500]

(temp), and location coordinates (lat, lon) of those sensor nodes located outside a specified rectangle (0, 0, 500, 500).

The three spatial operators shown in Table 3 were incorporated into SE TinyDB to facilitate the processing of queries on a specific area or moving trajectory. However, SE TinyDB does not support various spatial data types and spatial operators related to the Simple Feature Specification for SQL, the standard recommended by the OGC.

3. Spatial TinyDB

In this section, we explain the overall structure of Spatial TinyDB and look at the various managers comprising it.

3.1. System Structure. The proposed Spatial TinyDB extends TinyDB to facilitate the efficient management of spatial sensor data in USN environments. Figure 1 depicts the overall structure of Spatial TinyDB.

As depicted in Figure 1, Spatial TinyDB consists of a spatial data manager, a spatial query processing manager, a spatial data stream manager, a spatial interface manager, and a spatial data communication manager. In Figure 1, the light-blue boxes signify the extended TinyDB components, while the pink boxes signify newly added components.

The spatial data manager manages spatial data types in conformance with the OGC standards and converts spatial attribute information into spatial schema. The spatial query processing manager provides spatial relation operators and spatial analysis operators in line with the OGC standards. In addition, it provides spatial trajectory operators for processing the moving trajectories of sensor nodes. The spatial data stream manager provides memory sharing functions among spatial queries in processing spatial queries and filtering functions for reducing input load. The spatial interface manager receives spatial queries, parses them, and then displays the final results. Finally, the spatial data communication manager manages communication and sessions between the server PC and sensor nodes in a wireless sensor network and between sensor nodes.

3.2. Spatial Data Manager. In this subsection, we look at the spatial data type management module and the spatial schema conversion module forming the spatial data manager.

Table 5: Spatial data types and examples.

Spatial data types	Examples
Point	POINT (10 10)
LineString	LINESTRING (10 10, 20 20, 30 40)
Polygon	POLYGON (10 10, 10 20, 20 20, 20 15, 10 10)
PolyhedralSurface	POLYHEDRALSURFACE ((10 10, 10 20, 20 20, 20 10, 10 10), (20 10, 40 20, 20 20, 20 10))
MultiPoint	MULTIPOINT (10 10, 20 20)
MultiLineString	MULTILINESTRING ((10 10, 20 20), (15 15, 30 15))
MultiPolygon	MULTIPOLYGON ((10 10, 10 20, 20 20, 20 15, 10 10), (60 60, 70 70, 80 60, 60 60))

3.2.1. Spatial Data Type Management Module. The spatial data type management module provides the spatial data types recommended in the "Simple Features Specification for SQL" Standard Specifications [9] of the OGC in order to support spatial queries. Table 5 shows spatial data types and examples supported in the spatial data type management module.

As shown in Table 5, the spatial data type management module supports seven spatial data types: Point, LineString, Polygon, PolyhedralSurface, MultiPoint, MultiLineString, and MultiPolygon.

3.2.2. Spatial Schema Conversion Module. The spatial schema conversion module converts spatial attribute information sent by the spatial data communication manager into spatial schema according to spatial schema mapping rules. Figure 2 illustrates the spatial schema conversion process.

As depicted in Figure 2, the spatial schema conversion module converts the spatial attribute information Location, Lines, and Boundary into Point, LineString, and Polygon, respectively, according to spatial schema mapping rules.

3.3. Spatial Query Processing Manager. In this subsection, we look at the spatial relation operator module, the spatial analysis operator module, and the spatial trajectory operator module forming the spatial query processing manager.

3.3.1. Spatial Relation Operator Module. The spatial relation operator module provides the spatial relation operators recommended in "Simple Features Specification for SQL" Standard Specifications [9] of the OGC, in order to support spatial query processing in the server PC and sensor nodes. Table 6 shows the spatial relation operators provided in the spatial relation operator module.

As can be seen in Table 6, the spatial relation operator module supports eight spatial relation operators: Equals, Disjoint, Touches, Within, Overlaps, Crosses, Intersects, and Contains. A spatial relation operator receives two spatial objects, Geometry A and Geometry B, as the input and returns True or False as the output.

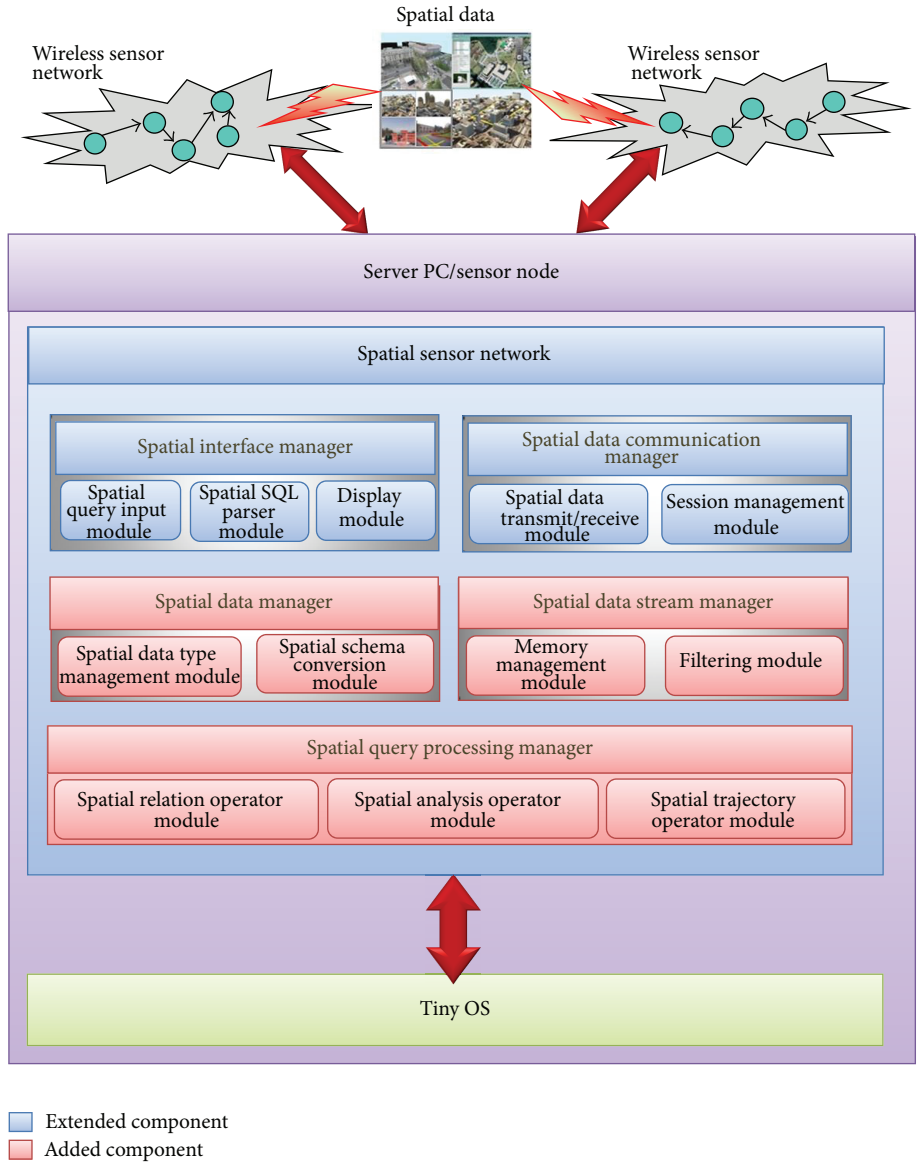


Figure 1: Structure of Spatial TinyDB.

3.3.2. Spatial Analysis Operator Module. The spatial analysis operator module provides the spatial analysis operators recommended in the "Simple Features Specification for SQL" Standard Specifications [9] of the OGC, in order to process spatial queries in the server PC and sensor nodes. Table 7 lists the spatial analysis operators provided in the spatial analysis operator module.

As shown in Table 7, the spatial analysis operator module supports six spatial analysis operators: Distance, Intersection, Difference, Union, Buffer, and ConvexHull. Each spatial analysis operator receives two spatial objects, Geometry A and Geometry B, as the input and returns a new spatial object as the output.

3.3.3. Spatial Trajectory Operator Module. The spatial trajectory operator module provides spatial trajectory operators for processing the moving trajectories of sensor nodes [12–14]. Table 8 lists the spatial trajectory operators provided in the spatial trajectory operator module.

As listed in Table 8, the spatial trajectory operator module supports five spatial trajectory operators: Enter, Insides, Leaves, Meets, and Passes. A spatial trajectory operator receives two spatial objects, Geometry A and Geometry B, as the input and returns True or False as the output.

3.4. Spatial Data Stream Manager. In this subsection, we look at the memory management module and the filtering module comprising the spatial data stream manager.

3.4.1. Memory Management Module. The memory management module provides data sharing functions among various spatial queries executed in Spatial TinyDB in order to reduce the system load caused by spatial data streams [5, 15].

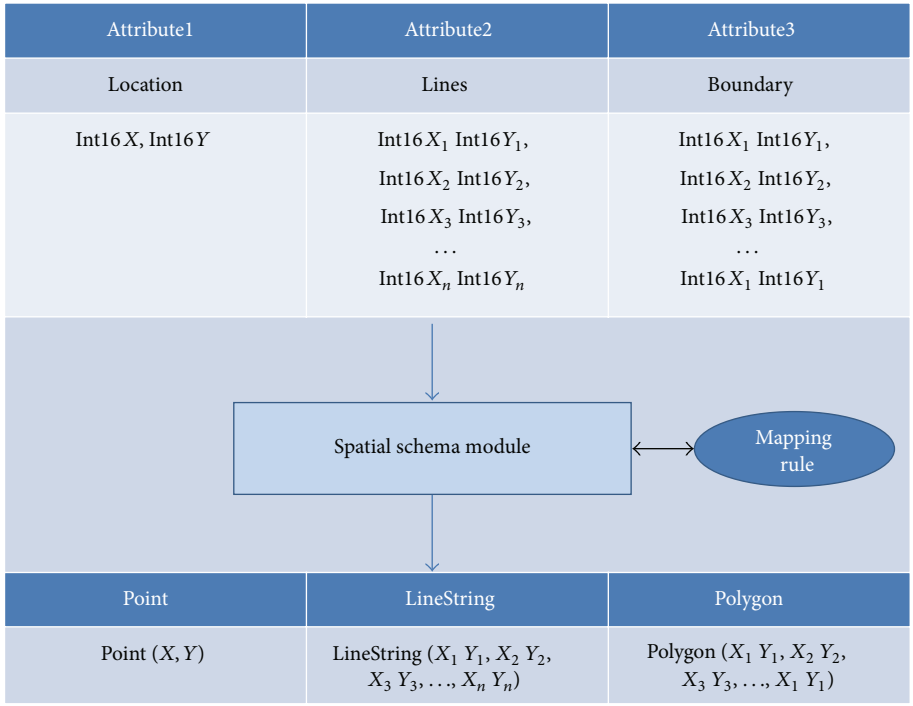


Figure 2: Spatial schema conversion process.

Table 6: Spatial relation operators.

Spatial relation operators	Explanation
Equals (Geometry A, Geometry B)	Return whether or not Object A is equal to Object B
Disjoint (Geometry A, Geometry B)	Return whether or not Object A is apart from Object B
Touches (Geometry A, Geometry B)	Return whether or not the boundary of Object A meets the boundary of Object B
Within (Geometry A, Geometry B)	Return whether or not Object A is included in Object B
Overlaps (Geometry A, Geometry B)	Return whether or not Object A and Object B overlap each other
Crosses (Geometry A, Geometry B)	Return whether or not Object A crosses Object B
Intersects (Geometry A, Geometry B)	Return whether or not Object A intersects Object B
Contains (Geometry A, Geometry B)	Return whether or not Object A contains Object B

Table 7: Spatial analysis operators.

Spatial analysis operators	Explanation
Distance (Geometry A, Geometry B)	Return the distance between Object A and Object B
Intersection (Geometry A, Geometry B)	Return the intersection of Object A and Object B
Difference (Geometry A, Geometry B)	Return the difference between Object A and Object B
Union (Geometry A, Geometry B)	Return the union of Object A and Object B
Buffer (Geometry A, Double L)	Return a spatial object whose boundary is larger by L from the boundary of Object A
ConvexHull (Geometry A)	Return the smallest convex polygon that can contain Object A

When Spatial TinyDB executes two or more spatial queries simultaneously, the spatial queries share one memory area instead of having their own respective memory areas. In the shared memory, a reference counter is set for each data tuple. If a spatial query processes a data tuple, the reference counter corresponding to the data tuple is reduced by 1, and if the reference counter becomes 0, the corresponding data tuple is deleted from the memory.

For example, if Spatial Query 1 saves a specific tuple in the depository and Spatial Query 2 and Spatial Query 3 share the tuple, the tuple's reference counter becomes 3. If Spatial Query 1 processes the tuple, the tuple is not deleted from the depository, but its reference counter is reduced by 1. If both Spatial Query 2 and Spatial Query 3 process the tuple, the reference counter becomes 0 and the corresponding tuple is deleted.

3.4.2. Filtering Module. The filtering module provides filtering functions that solve the overload problem by reducing the volume of the input data stream, while minimizing loss of accuracy [16]. It carries out filtering of the input data stream using filtering conditions such as difference in the distance of location coordinates, time range at specific times, and IDs of sensor nodes. It then delivers only the filtered data stream to the spatial query processing manager. Figure 3

Table 8: Spatial trajectory operators.

Spatial trajectory operators	Explanation
Enter (Geometry A, Geometry B)	Return whether or not Object A enters Object B from outside
Insides (Geometry A, Geometry B)	Return whether or not Object A stays inside Object B
Leaves (Geometry A, Geometry B)	Return whether or not Object A goes out of Object B from inside
Meets (Geometry A, Geometry B)	Return whether or not Object A only touches the boundary of Object B
Passes (Geometry A, Geometry B)	Return whether or not Object A enters Object B from outside and then goes out of Object B from inside

Table 9: Spatial query examples.

Query types	Examples
Spatial relation operators	SELECT nodeid, temp, loc FROM sensors WHERE Contains (polygon (0 0, 40 0, 40 40, 0 40, 0 0), loc)
Spatial analysis operators	SELECT nodeid, temp, loc FROM sensors WHERE Contains (Intersection (Polygon (0 0, 0 40, 40 40, 40 0, 0 0), Polygon (30 30, 30 70, 70 70, 70 30, 30 30), loc)
Spatial trajectory operators	SELECT nodeid, temp, loc FROM sensors WHERE Passes (loc, Polygon (0 0, 0 40, 40 40, 40 0, 0 0))

gives an example of spatial data filtering in accordance with a filtering condition that consists of IDs of sensor nodes and the difference in the distance of location coordinates.

As illustrated in Figure 3, if "ID is from 1 to 5 and the difference in the distance of location coordinates is less than 100" is the filtering condition, the sensor node filters out the input data stream with IDs from 1 to 5 and with the difference in distance less than 100 between the previous location and the current location of the same object, and it delivers only the filtered data stream to the spatial query processing manager.

3.5. Spatial Interface Manager. In this subsection, we explain the spatial query input module, the spatial SQL parser module, and the display module forming the spatial interface manager.

3.5.1. Spatial Query Input Module. The spatial query input module receives spatial queries from the user and delivers them to the spatial SQL parser module. In addition, it chooses between text and graphic modes in displaying the results of a spatial query and receives the parameters (execution cycle, query ID, query condition, etc.) of a spatial query.

In Spatial TinyDB, the format of the spatial query statements is the same as that used in TinyDB. However, it can also use spatial relation operators, spatial analysis operators, and spatial trajectory operators, in addition to spatial data types, when building a spatial query. Table 9 shows spatial query examples that are impossible in TinyDB but possible in Spatial TinyDB.

In Spatial TinyDB, various spatial queries can be built by using spatial data types, spatial relation operators, spatial analysis operators, and spatial trajectory operators, as depicted in Table 9.

3.5.2. Spatial SQL Parser Module. The spatial SQL parser module parses a spatial SQL statement received from the spatial query input module (by executing lexical and syntactic analyses) and tests the validity of the spatial SQL statement based on parsed information. It also does error processing when a spatial SQL statement is incorrect. In addition, for a spatial query for which spatial SQL parsing has been completed normally, the module creates spatial attribute and spatial operator information and delivers them to the spatial data communication manager.

3.5.3. Display Module. The display module displays the status of query execution, final query results, and so forth, on the screen of the server PC. In particular, the module displays final query results received from the spatial query processing manager in two modes—text and graphic. In addition, it can display spatial SQL query statements received from the spatial SQL parser module and error messages for errors occurring while a spatial query is being executed. Further, the display module allows the user to reset the query execution cycle during the execution of a spatial query and to control operations such as pausing and restarting a spatial query.

3.6. Spatial Data Communication Manager. In this subsection, we look at the session management module and the spatial data transmit/receive module forming the spatial data communication manager.

3.6.1. Session Management Module. The session management module creates, maintains, and deletes sessions between the server PC and the sensor nodes and between the sensor nodes within a wireless sensor network. A session is created when a new spatial query is started and if an existing spatial query is finished, the corresponding session is deleted for the flexible management of sessions. In addition, the module resets the effective duration of a session when the query execution cycle is set.

3.6.2. Spatial Data Transmit/Receive Module. The spatial data transmit/receive module transmits and receives data between the server PC and the sensor nodes and between the sensor nodes within a wireless sensor network. The spatial data transmit/receive module on the server PC sends spatial attribute and spatial operator information received from the spatial interface manager to each sensor node in the wireless sensor network and again sends the final results of a spatial query received from the wireless sensor network to the spatial interface manager. The spatial data transmit/receive module

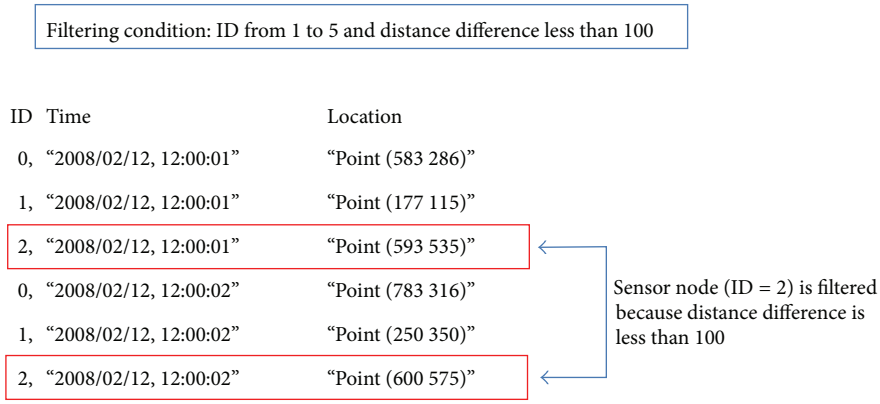


Figure 3: Example of spatial data filtering.

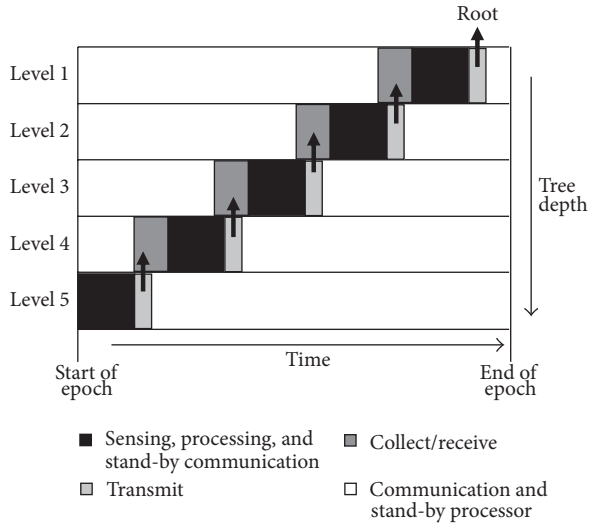


Figure 4: Spatial data transmit/receive process.

on a sensor node sends spatial data received from other sensor nodes to the query processing manager and also sends spatial data sensed by the node to other sensor nodes.

In particular, when spatial data that is to be transmitted exceed the size limit, the module divides the spatial data into smaller pieces before sending. Further, when spatial data sending fails, the module resends the data automatically. Furthermore, the module uses efficient spatial data transmit/receive methods suitable for the limited resources of sensor nodes. Figure 4 shows the spatial data transmit/receive process.

As illustrated in Figure 4, sensor nodes stay mainly in the waiting mode in order to conserve power. In addition, when a sensor node has completed operations such as data sensing, transmitting, and receiving within a certain amount of time, it is switched to the waiting mode automatically.

4. Performance Evaluation

In this section, we analyze the results of performance evaluation on the execution time, accuracy, and memory use of Spatial TinyDB proposed in this paper and TinyDB. Especially, the performance of Spatial TinyDB is evaluated both with and without spatial data filtering.

Table 10: Parameter values and query examples for performance evaluation.

Parameters	Values	
Spatial query execution cycle	256 ms	
Size limit of data	25 Byte	
The number of sensor nodes	1000, 2000, 3000, 4000	
The number of spatial queries	120	
Filtering conditions	Distance difference less than 100	
	TinyDB	Spatial TinyDB
Query examples	SELECT nodeid, temp, x, y FROM sensors WHERE temp > 40 AND x > 0 AND x < 400 AND y > 0 AND y < 400	SELECT nodeid, temp, Loc FROM sensors WHERE temp > 40 AND Contains (Polygon (0 0, 0 400, 400 400, 400 0, 0 0), Loc)

4.1. Performance Evaluation Environment. The proposed Spatial TinyDB was implemented using TinyOS 1.1.15 under Cygwin 2.5.7 as the operating system, and the development tools were nesC 1.2.8 and g++ 3.4.3 provided in TinyOS. In addition, the Java GUI was based on the Microsoft Windows XP Professional environment, and JAVA 1.4 was used as the development tool.

For the performance evaluation, we set parameters such as spatial query execution cycle, size limit of data in each transmission, number of sensor nodes, number of spatial queries executed simultaneously, and filtering condition. Table 10 lists the parameters and values set for the performance evaluation, along with the example queries used in the performance evaluation.

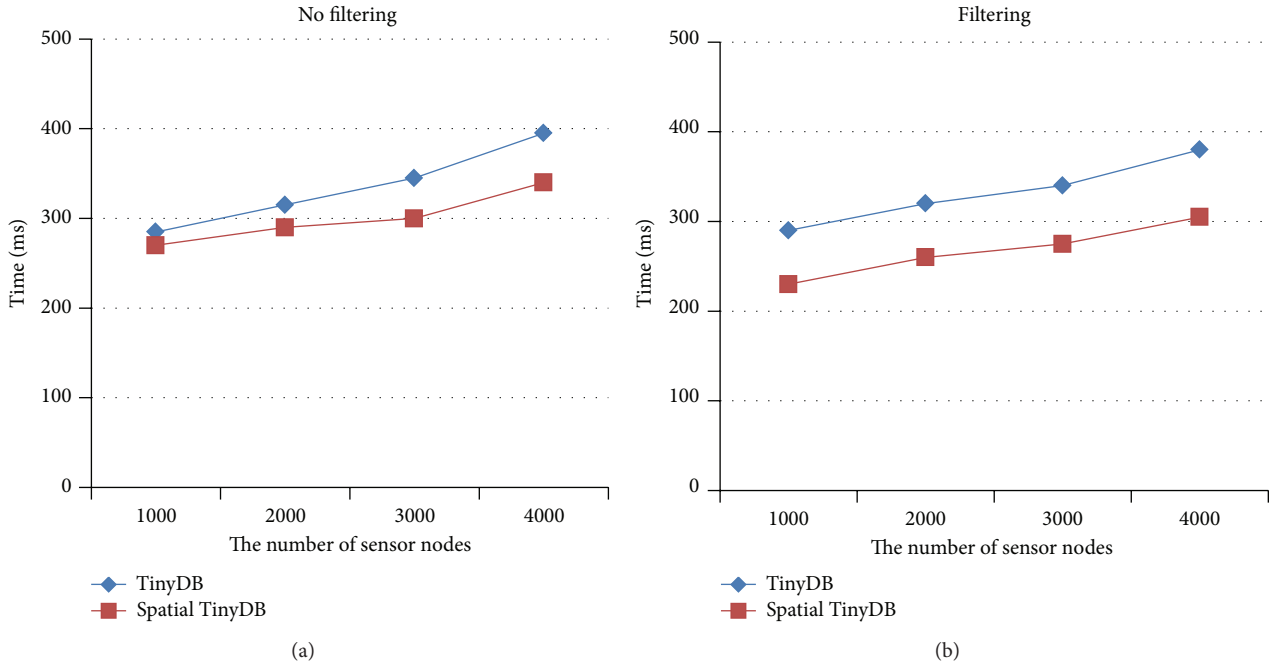


Figure 5: Measured results of the execution time.

4.2. Execution Time. Figure 5 graphically illustrates the measured results of the execution time of Spatial TinyDB and TinyDB versus the number of sensor nodes.

As illustrated in Figure 5(a) (no filtering used), Spatial TinyDB executed approximately 12% faster than TinyDB on average. This resulted from the fact that our Spatial TinyDB processes spatial queries faster because it supports spatial operators. When filtering was used (Figure 5(b)), the execution time of Spatial TinyDB was 21% faster than that of TinyDB on average. This resulted from the fact that the system load was reduced as input data were filtered before execution in Spatial TinyDB.

4.3. Accuracy. Figure 6 shows the measured results of the accuracy of Spatial TinyDB and TinyDB versus the number of sensor nodes.

As depicted in Figure 6(a) (no filtering used), Spatial TinyDB showed the same accuracy as TinyDB since there was no loss of input data in either of the two cases. In Figure 6(b), however, when filtering was used, Spatial TinyDB was approximately 7% less accurate than TinyDB on average. This resulted from the fact that input data filtering causes loss of some data in Spatial TinyDB.

4.4. Memory Usage. Figure 7 shows the measured results of the memory usage in Spatial TinyDB and TinyDB versus the number of sensor nodes.

As illustrated in Figure 7(a) (no filtering used), Spatial TinyDB showed memory usage of approximately 6% less than TinyDB on average. This resulted from the fact that memory is shared by multiple spatial queries in Spatial TinyDB. In addition, in Figure 7(b), when filtering was used, Spatial TinyDB showed memory usage approximately 11% less than that of TinyDB on average. This resulted from the fact that the spatial data stream manager reduced the volume of the input data stream not only through memory sharing but also through input data filtering.

5. Conclusions

In this paper, we propose Spatial TinyDB which is a spatial sensor database system that extends TinyDB to facilitate the efficient management of spatial sensor data in USN environments. The proposed Spatial TinyDB supports spatial data types, spatial relation operators, spatial analysis operators, and spatial trajectory operators that are in compliance with international standards. In addition, Spatial TinyDB provides memory management functions to facilitate sharing of necessary data among various spatial queries, and filtering functions to solve the overload problem by reducing the volume of the input data stream while minimizing loss of accuracy.

The results of performance evaluations indicate that Spatial TinyDB is superior to TinyDB in terms of execution time and memory use but shows slightly lower performance than TinyDB in terms of accuracy when filtering is used.

Acknowledgments

This work (Grants no. C0027296) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012.

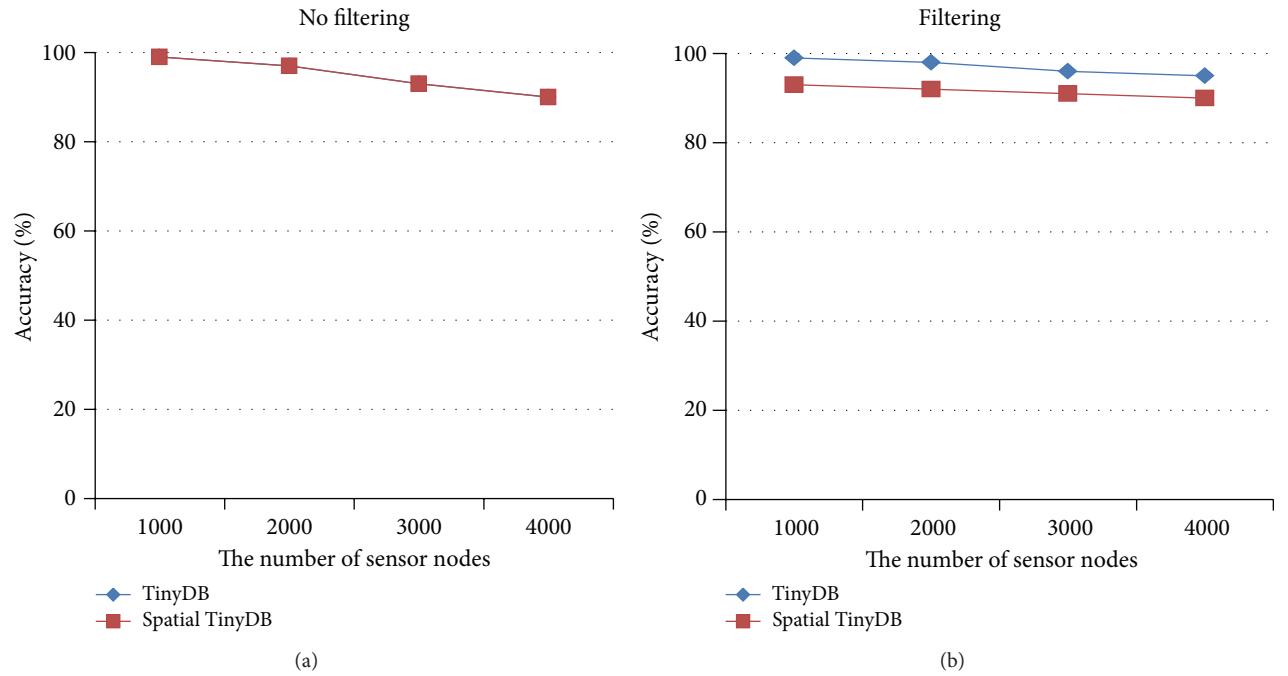


Figure 6: Measured results of the accuracy.

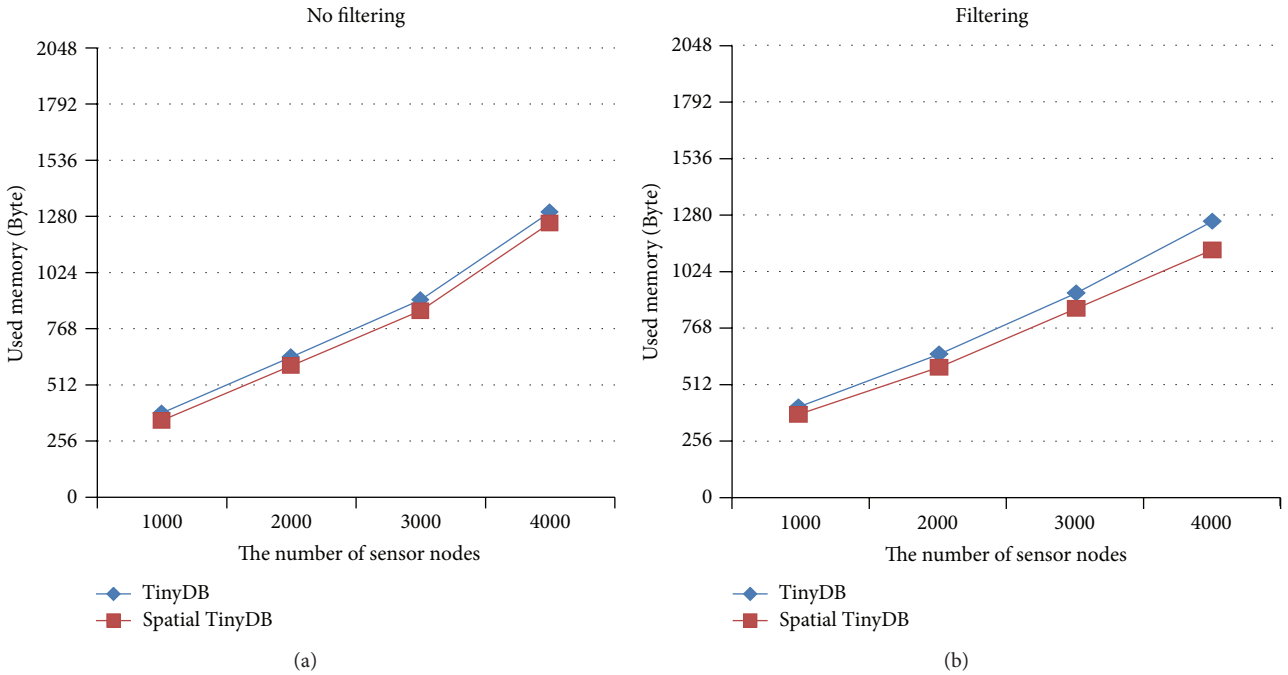


Figure 7: Measured results of the memory usage.

References

[1] M. Inoue, "A model and system architecture for ubiquitous sensor network businesses," in *Proceedings of the ITU-T Kaleidoscope Academic Conference on Innovations for Digital Inclusion*, pp. 1–8, September 2009.

[2] P. Andreou, D. Zeinalipour-Yazti, A. Pamboris, P. K. Chrysanthis, and G. Samaras, "Optimized query routing trees for wireless sensor networks," *Information Systems*, vol. 36, no. 2, pp. 267–291, 2011.

[3] E. Taslidere, F. S. Cohen, and F. K. Reisman, "Wireless sensor networks-a hands-on modular experiments platform for enhanced pedagogical learning," *IEEE Transactions on Education*, vol. 54, no. 1, pp. 24–33, 2011.

[4] S. Nittel, A. Labrinidis, and A. Stefanidis, "Introduction to advances in geosensor networks," in *GeoSensor Networks*, pp. 1–6, 2008.

[5] J. Park, K. Kim, S. Ahn, and B. Hong, "Continuous query processing on combined data stream: sensor, location and

identification," in *Proceedings of the 7th International Conference on Information Technology*, pp. 518–522, April 2010.

[6] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: an acquisitional query processing system for sensor networks," *ACM Transactions on Database Systems*, vol. 30, no. 1, pp. 122–173, 2005.

[7] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *SIGMOD Record*, vol. 31, no. 3, pp. 9–18, 2002.

[8] P. D. Felice, M. Lanni, and L. Pomante, "Design and evaluation of a spatial extension of TinyDB for wireless sensor networks," *International Journal of Computers and Their Applications Manuscript*, vol. 17, pp. 172–193, 2010.

[9] Open Geospatial Consortium, OpenGIS Implementation Specification for Geographic Information-Simple Feature Access-Part 1: Common Architecture, Version 1.2.1, 2010.

[10] P. Levis and H. Wei, "TinyDB: design, code and implementation," 2006, http://csl.stanford.edu/~pal/pubs/tinyos-programming.pdf.

[11] P. Levis, S. Madden, J. Polastre et al., "TinyOS: an operating system for wireless sensor networks," in *Ambient Intelligence*, pp. 115–148, 2005.

[12] M. Erwig and M. Schneider, "Developments in spatio-temporal query languages," in *Proceedings of the 10th International Workshop on Database and Expert Systems Applications*, pp. 441–449, 1999.

[13] J. H. Lee, K. H. An, and J. H. Park, "Design of query language for location-based services," in *Web and Wireless Geographical Information Systems*, vol. 3833 of *Lecture Notes in Computer Science*, pp. 11–18, 2005.

[14] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel approaches in query processing for moving objects," in *Proceedings of the 26th International Conference on Very Large Data Bases*, pp. 395–406, 2000.

[15] K. Križanović, Z. Galić, and M. Baranović, "Spatio-temporal data streams: an approach to managing moving objects," in *Proceedings of the 33rd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO '10)*, pp. 744–749, May 2010.

[16] D. Maier, P. A. Tucker, and M. Garofalakis, "Filtering, punctuation, windows and synopses," in *Stream Data Management*, chapter 3, pp. 35–58, 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 471962, 16 pages
http://dx.doi.org/10.1155/2013/471962

Research Article

Enhanced Mobile Multiple-Input Multiple-Output Underwater Acoustic Communications

Kexin Zhao,[1] Jun Ling,[2] and Jian Li[1]

[1] *Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA*
[2] *MathWorks Inc., Natick, MA 01760, USA*

Correspondence should be addressed to Kexin Zhao; kexinzhao@ufl.edu

Received 12 January 2013; Revised 22 April 2013; Accepted 15 May 2013

Academic Editor: Lei Shu

This paper focuses on mobile multiple-input multiple-output (MIMO) underwater acoustic communications (UAC) over double-selective channels subject to both intersymbol interference and Doppler scaling effects. Temporal resampling is implemented to effectively convert the Doppler scaling effects to Doppler frequency shifts. Under the assumption that the channels between all the transmitter and receiver pairs experience the same Doppler frequency, a variation of the recently proposed generalization of the sparse learning via iterative minimization (GoSLIM) algorithm, referred to as GoSLIM-V, is employed to estimate the frequency modulated acoustic channels. GoSLIM-V is user parameter free and is easy to use in practical applications. This paper also considers turbo equalization for retrieving the transmitted signal. In particular, this paper reviews the linear minimum mean-squared error (LMMSE) based soft-input soft-output equalizer involved in the turbo equalization scheme and adopts a fast implementation of the equalizer that achieves negligible detection performance degradation compared to its direct implementation counterpart. The effectiveness of the considered MIMO UAC scheme is demonstrated using both simulated data and measurements recently acquired during the MACE10 in-water experiment.

1. Introduction

Achieving reliable underwater acoustic communications (UAC) with high data rate is difficult owing to the unique challenges imposed by the underwater acoustic environment [1, 2]. In typical UAC, the difference in the propagation time between the earliest and latest arrivals could span tens to hundreds of symbol periods [3], which translates into long channel impulse response (CIR) and severe intersymbol interference (ISI) at the receiver side. Moreover, the presence of Doppler effects, owing to the relative motions between the transmitter and receiver platforms and the dynamic underwater acoustic medium, induces temporal scaling (stretching or compression) to the transmitted signals [4]. Doppler-induced scaling effects impair the reliability of UAC, especially in the case of a phase-coherent detection scheme [3]. Furthermore, the scarcely available bandwidth permitted by the acoustic channel imposes an upper bound on the attainable symbol rate [2]. Therefore, the pursuit of high data rate in UAC leverages the multiple-input multiple-output (MIMO) scheme, which offers enhanced reliability and/or increased data rates compared to its single-input counterpart [5–7]. The focus of the present paper is on effective mobile MIMO UAC over double-selective acoustic channels suffering from both ISI and Doppler scaling effects.

Converting the double-selective channel into an ISI channel via temporal resampling is an effective way to tackle mobile UAC difficulties [4]. Although the Doppler scaling effects can be largely mitigated via such a temporal resampling process, the residual Doppler still causes frequency shift on the received measurements. Coherent UAC requires the receiver to acquire knowledge of the underlying channel after temporal resampling via channel estimation [7]. Channel estimation could be conducted either in the training-directed mode, using known training sequences, or in the decision-directed mode, using the detected payload symbols [5, 6]. A preferable tool to characterize a channel subject to both ISI and Doppler frequency shift is the scattering function (SF), which essentially decouples the acoustic channel into a bank of paths that experience different delays and Doppler

frequencies [8]. The major concern in SF-based channel estimation is that the problem becomes over parameterized with too many degrees of freedom. It is practically more beneficial to look for a channel model with the smallest number of parameters, but one that still sufficiently reflects the defining characteristics of the acoustic channel of interest. Along this line of thought, it is assumed in [9, 10] that, at each receiver, the channel taps for all the transmitters experience the same Doppler frequency, but different receivers experience different Doppler shifts. The number of unknowns in the frequency dimension, as a consequence, is significantly reduced. Under this assumption, CIRs and the underlying Doppler frequency could be estimated in a separate manner [9] or in a joint manner by employing the generalization of the sparse learning via iterative minimization (GoSLIM) algorithm [10]. It is demonstrated in [10] that GoSLIM outperforms the separate estimation algorithm proposed in [9] in terms of estimation accuracy and robustness against suboptimal training sequences.

In [11], the aforementioned channel model is further simplified by assuming that the channel taps for all the transmitter and receiver pairs experience the same Doppler frequency. As a consequence, the impact of the Doppler frequency shift on the received measurements across all the receivers is taken into account through one unknown common frequency. Accordingly, a variation of GoSLIM, referred to as GoSLIM-V (V stands for variation), is proposed for channel estimation in [11]. Like GoSLIM, GoSLIM-V addresses sparsity through a hierarchical Bayesian model, and because GoSLIM-V is user parameter free, it is easy to use in practical applications. It is demonstrated in [11] that the employment of GoSLIM-V not only reduces the overall complexity in the channel estimation stage but also slightly improves the detection performance compared to its GoSLIM counterpart. Due to this reason, GoSLIM-V is used as the channel estimation algorithm in the present paper.

Following the channel estimation is the design of the detection scheme for extracting the transmitted signals. The channel-induced phase shift should be first compensated out using the Doppler frequency estimate [9]. Such phase compensation task, along with the aforementioned temporal resampling process, effectively converts a double-selective channel subject to both Doppler scaling effects and ISI to an ISI channel, which allows for the employment of various equalization techniques that can effectively combat ISI. We use a linear minimum mean-squared error (LMMSE) based filter for symbol detection. In a MIMO setup, on top of ISI, multiple simultaneously transmitted signals act as interferences to one another. Therefore, interference cancellation scheme also plays a critical role in the overall detection performance. A hard decision based interference cancellation scheme, including vertical BLAST (V-BLAST) [12] and RELAX-BLAST [5], subtracts out the hard decisions of detected signals from the received measurements to aid the detection of the remaining signals. By combining V-BLAST with the cyclic principle of the RELAX algorithm [13], RELAX-BLAST provides superior detection performance over V-BLAST at the cost of slightly increased complexities [5, 6].

The detection performance can be further enhanced by employing a soft interference cancellation scheme, including turbo equalization [14–16]. For a receiver employing turbo equalization, both the equalizer and decoder involved are configured as soft-input soft-output. The detection performance improves as the soft information cycles between the equalizer and decoder. The main drawback of the turbo equalization scheme is the increased computational complexity compared to its hard-decision-based counterparts. To address this problem, we consider a low complexity approximation of soft-input soft-output equalizer [14]. We will show via numerical and experimental examples that the employment of the proposed approximate equalizer enjoys a computational complexity comparable to RELAX-BLAST and provides only slightly degraded detection performance compared to an exactly implemented turbo equalizer.

The rest of the paper is organized as follows. Section 2 presents a system outline. Section 3 describes a model for the acoustic channel subject to both ISI and Doppler scaling effects and reviews the temporal resampling procedure. Section 4 formulates the channel estimation problem in both training-directed and decision-directed modes and then introduces GoSLIM-V as the channel estimation algorithm. Section 5 first formulates the symbol detection problem and then details the LMMSE based soft-input soft-output equalizer and its low complexity approximation. Section 6 presents the simulation results of the turbo equalization scheme, followed by the experimental results obtained from analyzing the MACE10 in-water measured data. The paper is concluded in Section 7.

Notation. Vectors and matrices are denoted by boldface lowercase and uppercase letters, respectively, $\|\cdot\|$ denotes the Euclidean norm of a vector, $|\cdot|$ is the modulus and $(\cdot)^*$ is the complex conjugate of a scalar. $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and conjugate transpose, respectively, of a matrix or vector, $\mathbf{I}$ denotes an identity matrix of appropriate dimension, and $\hat{\mathbf{x}}$ denotes the estimate of $\mathbf{x}$. $\mathrm{diag}(\mathbf{v})$ represents a diagonal matrix in which the elements of $\mathbf{v}$ are on the diagonal. $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ represent the real and the imaginary components of a complex-valued scalar, respectively. $\mathbf{A} \otimes \mathbf{B}$ denotes the Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$. Other mathematical symbols are defined after their first appearance.

2. System Outline

Consider an $N \times M$ mobile MIMO UAC system equipped with N transmit transducers and M receive hydrophones. The transmitted payload sequences are divided into multiple blocks, each of which is encoded separately. Figure 1(a) demonstrates the construction of a single payload symbol block (the construction of other blocks follows the same procedure). Denote $a(k) \in \{0, 1\}$ as the kth source bit for $k = 1, \ldots, NK$. $\{a(k)\}_{k=1}^{NK}$ are first fed into a 1/2 rate convolutional encoder with generator polynomials (1 0 0 1 1) and (1 1 0 0 1 1). The encoded bits $\{b(k)\}_{k=1}^{2NK}$ are then passed to a random interleaver, followed by a quadrature

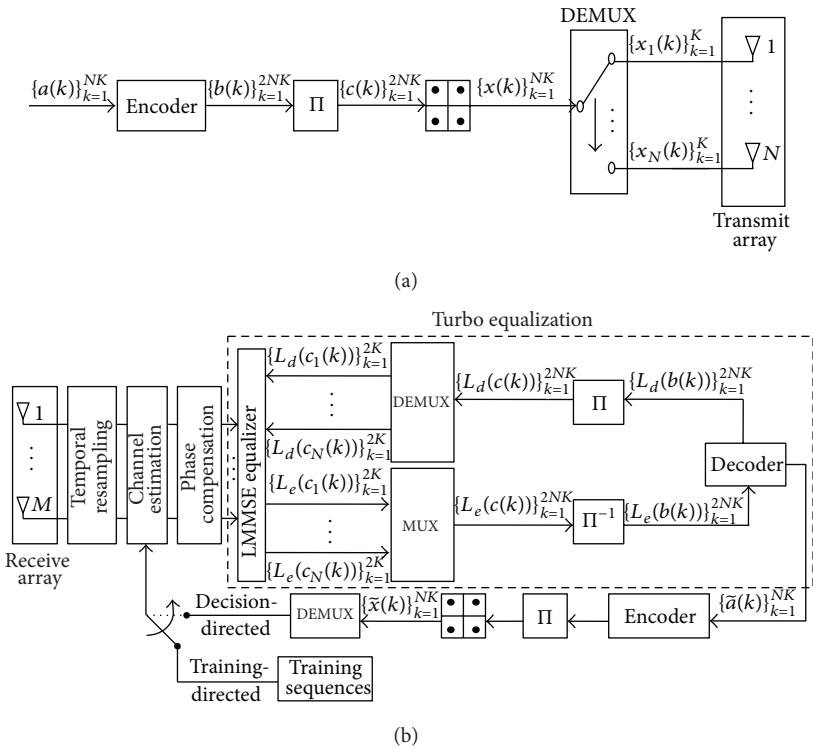


Figure 1: An $N \times M$ MIMO UAC system. (a) Transmitter structure. (b) Receiver structure by employing turbo equalization.

phase-shift keying (QPSK) modulation using Gray code mapping. In Figure 1, interleaver and deinterleaver modules are represented by Π and Π^{-1}, respectively. Next, the so-obtained QPSK payload symbols $\{x(k)\}_{k=1}^{NK}$ are demultiplexed into the N payload blocks, each consisting of K symbols, across the N transmitters in a round-robin fashion (this is where the "DEMUX" module in Figure 1(a) comes into play). More specifically, in our design, $x(n+(q-1)N)$ corresponds to the qth symbol sent by the nth transmitter, denoted as $x_n(q)$, for $q = 1, \ldots, K$ and $n = 1, \ldots, N$. Accordingly, we denote $c_n(2q-1)$ and $c_n(2q)$ as the two consecutive interleaved bits in $\{c(k)\}_{k=1}^{2NK}$ that map to $x_n(q)$ according to the formula given below:

$$x_n(q) = \frac{1}{\sqrt{2}}\left[j(-1)^{c_n(2q-1)} + (-1)^{c_n(2q)}\right], \quad q = 1, \ldots, K,\ n = 1, \ldots, N. \tag{1}$$

Since $c(k) \in \{0, 1\}$, the support of $\{x_n(k)\}$ is a 4-element alphabet set $\mathcal{S} = \{(\pm j \pm 1)/\sqrt{2}\}$.

The structure of a receiver employing a turbo equalization scheme is shown in Figure 1(b). The measurements acquired by the M receive hydrophones are first resampled, followed by channel estimation and phase compensation. After phase compensation, the double-selective channel is converted to an ISI channel, and the turbo equalization scheme is employed herein to retrieve the transmitted information. The superior detection performance promised by turbo equalization is mainly due to its mechanism of cycling soft information between the equalizer and the decoder [14]. Accordingly, turbo equalization consists of two key modules, namely, a soft-input soft-output equalizer and a soft-input soft-output decoder [17]. The soft information of a generic bit $a \in \{0, 1\}$, commonly known as the log-likelihood ratio (LLR), is defined as

$$L(a) = \ln \frac{P(a = 0)}{P(a = 1)}, \tag{2}$$

where $P(a = 0) \in [0, 1]$ represents the probability of a being 0. As shown in Figure 1(b), the multiplexed and deinterleaved version of $\{L_e(c_n(k))\}_{n=1,k=1}^{N2K}$, the *a posteriori extrinsic* LLR generated by the equalizer, forms the *a priori* inputs $\{L_e(b(k))\}_{k=1}^{2NK}$ to the decoder. Conversely, the interleaved and demultiplexed version of $\{L_d(b(k))\}_{k=1}^{2NK}$, the *a posteriori extrinsic* LLR generated by the decoder, serves as *a priori* information to the equalizer. The subscript e or d reminds us that the LLRs are generated by the equalizer or the decoder, respectively. The soft information is cycled between the equalizer and the decoder multiple times before making hard decisions on the source bits. Note that the interleaver and deinterleaver involved at the transmitter and receiver have the same structure, whereas the "DEMUX" module inside the dashed rectangle in Figure 1(b) is different from that in Figure 1(a) in the sense that the former and the latter demultiplex, respectively, the soft information $\{L_d(c(k))\}_{k=1}^{2NK}$ and QPSK symbols $\{x(k)\}_{k=1}^{NK}$. Once $\{\widetilde{a}(k)\}_{k=1}^{NK}$, the hard decisions on the source bits, are available, we follow the steps in the symbol generation process shown in Figure 1(a): $\{\widetilde{a}(k)\}_{k=1}^{NK}$ are fed into the convolutional encoder, followed by

random interleaving, QPSK mapping, and demultiplexing. This way, an error free decoding ensures a perfect recovery of the transmitted QPSK symbols $\{x_n(k)\}_{n=1,k=1}^{NK}$. The recovered payload symbols will be used in the decision-directed channel estimation stage; see Figure 1(b).

3. Double-Selective Channel with Doppler Scaling Effects

In this section, we start with the modeling of the double-selective channel suffering from both ISI and Doppler scaling effects. Then we describe the temporal resampling procedure to mitigate the Doppler scaling effects. After that, we provide a practical approach to estimate the Doppler scaling factor.

3.1. Channel Model. By adopting a single-carrier communication scheme, at the nth transmitter, the continuous baseband signal $x_n(t)$ (generated by passing the discrete payload symbols $\{x_n(k)\}$ to a pulse shaping filter) and its corresponding frequency modulated signal $\breve{x}_n(t)$ are related through

$$\breve{x}_n(t) = \mathrm{Re}\left\{x_n(t)\, e^{j2\pi f_c t}\right\}, \quad n = 1, \ldots, N, \tag{3}$$

where f_c represents the carrier frequency. For simplicity, the pulse shaping filter, frequency modulation, and real component extraction operation are not shown in Figure 1(a).

Due to multipath effects, the actual transmitted signals $\{\breve{x}_n(t)\}_{n=1}^{N}$ can reach the receive hydrophones via different propagation paths with different delays. Herein, the underlying acoustic channel between each transmitter and receiver pair is characterized by R resolved paths. The rth path between the nth transmitter and mth receiver pair ($r = 1, \ldots, R$, $n = 1, \ldots, N$, and $M = 1, \ldots, M$) will affect the transmitted signal $\breve{x}_n(t)$ in three aspects, namely, amplitude attenuation, Doppler scaling, and delay, which are denoted, respectively, by three real-valued scalars $\kappa_{n,m}(r)$, $\alpha_{n,m}(r)$, and $\tau_{n,m}(r)$. The signal transmitted via the rth path and acquired by the mth receiver can be written as $\kappa_{n,m}(r) \cdot \breve{x}_n(\alpha_{n,m}(r)t - \tau_{n,m}(r))$. By taking into account all of the N transducers and R resolved paths, the received signal at the mth hydrophone can be expressed as (for simplicity, the noise term is omitted for the time being)

$$\breve{z}_m(t) = \sum_{n=1}^{N}\sum_{r=1}^{R} \kappa_{n,m}(r) \cdot \breve{x}_n\left(\alpha_{n,m}(r)\, t - \tau_{n,m}(r)\right), \quad m = 1, \ldots, M. \tag{4}$$

We assume that the propagation paths for all the transmitter and receiver pairs experience a common Doppler scaling factor and the resolved paths are synchronized among all the transmitter and receiver pairs, that is, $\alpha_{n,m}(r) = \alpha$ and $\tau_{n,m}(r) = \tau(r)$ (interested readers are referred to [18] for a detailed treatment of synchronization procedure). By using these assumptions, (4) reduces to

$$\breve{z}_m(t) = \sum_{n=1}^{N}\sum_{r=1}^{R} \kappa_{n,m}(r) \cdot \breve{x}_n(\alpha t - \tau(r)), \quad m = 1, \ldots, M. \tag{5}$$

Substituting (3) into (5) yields

$$\breve{z}_m(t) = \mathrm{Re}\left\{\sum_{n=1}^{N}\sum_{r=1}^{R} \kappa_{n,m}(r) \cdot x_n(\alpha t - \tau(r))\, e^{j2\pi f_c(\alpha t - \tau(r))}\right\}. \tag{6}$$

3.2. Temporal Resampling. By resampling the received measurements $\{\breve{z}_m(t)\}$ using a factor β, the resampled signal $\breve{y}_m(t)$ is given by [4, 19, 20]

$$\breve{y}_m(t) = \breve{z}_m\left(\frac{t}{\beta}\right). \tag{7}$$

Then, the baseband received signal $y_m(t)$, which is related to $\breve{y}_m(t)$ via $\breve{y}_m(t) = \mathrm{Re}\{y_m(t)e^{j2\pi f_c t}\}$, can be expressed as

$$y_m(t) = e^{-2j\pi((\beta-\alpha)/\beta)f_c t}\sum_{n=1}^{N}\sum_{r=1}^{R} h_{n,m,r} \cdot x_n\left(\frac{\alpha}{\beta}t - \tau(r)\right), \tag{8}$$

where $h_{n,m,r} \triangleq \kappa_{n,m}(r)e^{-j\pi f_c \tau(r)}$ represents the rth channel tap between the nth transmitter and the mth receiver pair, for $r = 1, \ldots, R$, $n = 1, \ldots, N$, and $m = 1, \ldots, M$. It can be readily verified that as long as $\alpha/\beta \approx 1$, we have $x_n((\alpha/\beta)t - \tau(r)) \approx x_n(t - \tau(r))$. Accordingly, (8) can be approximated as

$$y_m(t) \approx e^{-2j\pi((\beta-\alpha)/\beta)f_c t}\sum_{n=1}^{N}\sum_{r=1}^{R} h_{n,m,r} \cdot x_n(t - \tau(r)). \tag{9}$$

One observes from (9) that effective temporal resampling (meaning $\alpha \approx \beta$) converts the Doppler scaling effects to Doppler frequency shifts with the frequency given below:

$$f = \left(\frac{\beta - \alpha}{\beta}\right) f_c. \tag{10}$$

Therefore, the determination of the resampling factor β plays a crucial role in the effective mitigation of the Doppler scaling effects.

3.3. Resampling Factor Estimation. We take advantage of the preamble and the postamble of a data packet to estimate β [19] (the structure of a data packet will be discussed in Section 6). By cross-correlating the received signal with the known preamble and postamble, the receiver estimates the time duration of a packet $\widehat{T}_{rx}$ [4]. By comparing $\widehat{T}_{rx}$ with T_{tx}, the duration of the same packet at the transmitter side, the Doppler scaling factor can be estimated as

$$\widehat{\beta} = \frac{\widehat{T}_{rx}}{T_{tx}}. \tag{11}$$

Although this method is conceptually simple and easy to implement, its accuracy is sensitive to the signal-to-noise-ratio (SNR).

More accurate Doppler scaling factor estimate can be achieved via channel estimation instead of cross correlation. Based on the CIRs estimated from the two measurement segments in response to the preamble and postamble, the change in the time duration $\widehat{T}_d$ imposed on the packet can be inferred from the tap shift of the principal arrivals of these two CIRs. Then the Doppler scaling factor estimate can be computed as

$$\widehat{\beta} = \frac{T_{tx} + \widehat{T}_d}{T_{tx}}. \tag{12}$$

The $\widehat{\beta}$ obtained using (12) is more robust against the noise contamination than the one from (11). We will show later on in Section 6.2.1 via the MACE10 in-water experimental data that the method in (12) works well in practice.

4. Channel Estimation

Since the $\widehat{\beta}$ obtained using (12) can never be perfectly accurate, after temporal resampling, Doppler frequency shifts (see (10)) still exist, although Doppler scaling effects become negligible. We start below with the problem formulation of channel estimation in both training-directed and decision-directed modes [5, 6]. Then, we propose the GoSLIM-V algorithm for jointly estimating the underlying CIRs and Doppler frequency.

In what follows, $\{y_m(t)\}_{m=1}^{M}$ and $\{x_n(t)\}_{n=1}^{N}$ in (9) are represented in discrete-time form. Unless otherwise stated, it is assumed that the channel taps for all the $N \times M$ transmitter-receiver pairs experience the same Doppler frequency f.

4.1. Training-Directed Mode. The initial task of the receiver is to acquire knowledge of the underlying channel between all transmitter and receiver pairs using the training sequences. By adopting the cyclic prefix scheme in [7], the training sequence at the nth transmitter ($n = 1, \ldots, N$) is given by

$$\mathbf{x}_n = \Bigg[\underbrace{x_n(P - L_{\mathrm{CP}} + 1), \ldots, x_n(P)}_{L_{\mathrm{CP}} \text{ prefix symbols}}, \underbrace{x_n(1), x_n(2), \ldots, x_n(P)}_{P \text{ core training symbols}} \Bigg], \tag{13}$$

where $[x_n(1), \ldots, x_n(P)]$ is the core training sequence and the leading L_{CP} symbols form the cyclic prefix. In general, we have $P > L_{\mathrm{CP}} \geq R - 1$. From an amplifier efficiency point of view, it is practically desirable to use unit modulus (unimodular) training sequences, that is, $|x_n(p)| = 1$ for $n = 1, \ldots, N$ and $p = 1, \ldots, P$.

For MIMO UAC over acoustic channels subject to both ISI and Doppler frequency shifts, the measurement vectors can be written as [8, 21]

$$\mathbf{y}_m = \overline{\mathbf{\Lambda}} \sum_{n=1}^{N} \mathbf{X}_n \mathbf{h}_{n,m} + \mathbf{e}_m, \quad m = 1, \ldots, M, \tag{14}$$

where

$$\mathbf{y}_m = [y_m(1), \ldots, y_m(P)]^T, \quad m = 1, \ldots, M, \tag{15}$$

contains the P synchronized measured symbols (for instance, $\{y_m(1)\}$ maps to $\{x_n(1)\}$) at the mth receiver. $\mathbf{X}_n \in \mathbb{C}^{P \times R}$ is given by

$$\mathbf{X}_n = \begin{bmatrix} x_n(1) & x_n(P) & \cdots & x_n(P-R+2) \\ x_n(2) & x_n(1) & \cdots & x_n(P-R+3) \\ \vdots & \vdots & \ddots & \vdots \\ x_n(P) & x_n(P-1) & \cdots & x_n(P-R+1) \end{bmatrix}, \tag{16}$$

where $n = 1, \ldots, N$ and $\mathbf{e}_m$ represents additive noise at the mth receiver. In addition,

$$\mathbf{h}_{n,m} = [h_{n,m,1}, \ldots, h_{n,m,R}]^T, \tag{17}$$

characterizes the channel of length R between the nth transmitter and the mth receiver, where $n = 1, \ldots, N$ and $m = 1, \ldots, M$ ($\{h_{n,m,r}\}$ is defined after (8)). Finally, the so-called Doppler shift matrix $\overline{\mathbf{\Lambda}} \in \mathbb{C}^{P \times P}$ in (14) is constructed as

$$\overline{\mathbf{\Lambda}} = \mathrm{diag}\left(\left[1, e^{-2j\pi f T_s}, \ldots, e^{-2j\pi f T_s (P-1)} \right] \right), \tag{18}$$

for $m = 1, \ldots, M$, where f and T_s represent the Doppler frequency and symbol period, respectively.

The ISI and Doppler shift effects can be viewed separately in (14). More specifically, the term $\sum_{n=1}^{N} \mathbf{X}_n \mathbf{h}_{n,m}$ indicates the net contribution of N ISI channels, while the impact of the Doppler effects on the measurements comes through $\overline{\mathbf{\Lambda}}$ only, which corresponds to the assumption that all the NMR CIR taps involved (recall that we have N transmit transducers, M receive hydrophones, and each transmitter-receiver pair corresponds to an R-tap channel) experience the same Doppler frequency f. The purpose of setting the first diagonal element of $\overline{\mathbf{\Lambda}}$ to 1 (see (18)) is to eliminate ambiguities. In our example, relative to $y_m(1)$, a generic measurement, say $y_m(p)$, experiences a phase shift of $-fT_s(p-1)$.

We express (14) in a more compact form as

$$\mathbf{y}_m = \overline{\mathbf{\Lambda}\mathbf{X}} \mathbf{h}_m + \mathbf{e}_m, \quad m = 1, \ldots, M, \tag{19}$$

where $\overline{\mathbf{X}} = [\mathbf{X}_1, \ldots, \mathbf{X}_N]$ and $\mathbf{h}_m = [\mathbf{h}_{1,m}^T, \ldots, \mathbf{h}_{N,m}^T]^T$. By stacking up the measurements, (19) can be rewritten as

$$\mathbf{y} = \mathbf{\Lambda} \mathbf{X} \mathbf{h} + \mathbf{e}, \tag{20}$$

where $\mathbf{y} = [\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_M^T]^T$, $\mathbf{h} = [\mathbf{h}_1^T, \mathbf{h}_2^T, \ldots, \mathbf{h}_M^T]^T$, $\mathbf{e} = [\mathbf{e}_1^T, \mathbf{e}_2^T, \ldots, \mathbf{e}_M^T]^T$, $\mathbf{\Lambda} = \mathbf{I}_{M \times M} \otimes \overline{\mathbf{\Lambda}}$, and $\mathbf{X} = \mathbf{I}_{M \times M} \otimes \overline{\mathbf{X}}$. Then the training-directed channel estimation reduces to estimating

$\mathbf{h}$ and f from the measurement vector $\mathbf{y}$ and known $\mathbf{X}$. The subject of synthesizing unimodular training sequences, coupled with the employment of the cyclic prefix scheme, to facilitate ISI channel estimation is thoroughly treated in [6]. The shifted PeCAN waveforms [22] are used as the training sequences in the MACE10 in-water experimentations.

4.2. Decision-Directed Mode. The decision-directed channel estimation problem is only a slight twist of its training-directed counterpart. For the former, we use the hard decisions of the previously estimated payload symbols, instead of the training symbols, to estimate the channels; see Figure 1(b). Accordingly, (14) can still be used, where

$$\mathbf{y}_m = \left[y_m(t_i), \ldots, y_m(t_f)\right]^T, \quad m = 1, \ldots, M, \tag{21}$$

contains the measurements at the mth receiver belonging to the time index interval $[t_i, t_f]$, and

$$\mathbf{X}_n = \begin{bmatrix} \widetilde{x}_n(t_i) & \widetilde{x}_n(t_i - 1) & \cdots & \widetilde{x}_n(t_i - R + 1) \\ \widetilde{x}_n(t_i + 1) & \widetilde{x}_n(t_i) & \cdots & \widetilde{x}_n(t_i - R + 2) \\ \vdots & \vdots & \ddots & \vdots \\ \widetilde{x}_n(t_f) & \widetilde{x}_n(t_f - 1) & \cdots & \widetilde{x}_n(t_f - R + 1) \end{bmatrix}, \tag{22}$$

for $n = 1, \ldots, N$, where $\widetilde{x}_n(t_i - R + 1)$ and $\widetilde{x}_n(t_f)$ represent the hard decisions of the first and last previously estimated symbols (some of them could be the known training symbols), respectively, used for updating the channel. (For notational simplicity, $\mathbf{X}_n$ is used in both (16) and (22) to represent two similar but different quantities. The use of $\mathbf{X}_n$, however, should be clear from the context.) The tracking length is represented as $L_{TR} = t_f - t_i + 1$, that is, the number of rows of $\mathbf{X}_n$. To conform with the matrix dimensions, the Doppler shift matrix $\overline{\mathbf{\Lambda}}$ now is L_{TR} by L_{TR}, constructed as $\overline{\mathbf{\Lambda}} = \mathrm{diag}([1, e^{-2j\pi f T_s}, \ldots, e^{-2j\pi f T_s (L_{TR}-1)}])$. Similar to the training-directed mode, the channel estimation problem in the decision-directed mode aims to estimate $\mathbf{h}$ and f from the measurement vector $\mathbf{y}$ and known $\mathbf{X}$ formed from the decision-directed $\{\mathbf{X}_n\}_{n=1}^N$ in (22).

4.3. Channel Estimation Algorithm: GoSLIM-V. The channel estimation algorithm, in either training- or decision-directed mode, has the generic form given by (20). We remark that the number of elements in $\mathbf{y}_m$, namely, d_y, might vary in different modes. $\mathbf{e}$ in (20) is assumed to contain circularly symmetric independent and identically distributed (i.i.d.) complex-valued Gaussian random variables with zero-mean and variance η, denoted as $\mathbf{e} \sim \mathcal{CN}(\mathbf{0}, \eta\mathbf{I})$. The problem is then to estimate f (contained in $\mathbf{\Lambda}$) and $\mathbf{h}$ given $\mathbf{y}$ and $\mathbf{X}$. In UAC systems, the channel $\mathbf{h}$ is usually sparse; that is, although it contains NMR unknowns, many of them can be approximated as zero [23]. We present the GoSLIM-V algorithm to solve this channel estimation problem. Note that since $\mathbf{h}$ contains the CIRs of all $N \times M$ transmitter-receiver pairs, the GoSLIM-V algorithm will estimate them simultaneously.

Consider the following hierarchical Bayesian model:

$$\mathbf{y} \mid \mathbf{h}, \mathbf{\Lambda}, \eta \sim \mathcal{CN}(\mathbf{\Lambda X h}, \eta\mathbf{I}), \tag{23}$$

$$\mathbf{h} \mid \mathbf{p} \sim \mathcal{CN}(\mathbf{0}, \mathbf{P}), \tag{24}$$

where (23) follows directly from the assumption $\mathbf{e} \sim \mathcal{CN}(\mathbf{0}, \eta\mathbf{I})$. Let $p_{n,m,r}$ be the variance of $h_{n,m,r}$ for $n = 1, \ldots, N$, $m = 1, \ldots, M$, and $r = 1, \ldots, R$, and define $\mathbf{p}_{n,m} = [p_{n,m,1}, p_{n,m,2}, \ldots, p_{n,m,R}]^T$, $\mathbf{p}_m = [\mathbf{p}_{1,m}^T, \mathbf{p}_{2,m}^T, \ldots, \mathbf{p}_{N,m}^T]^T$, and $\mathbf{p} = [\mathbf{p}_1^T, \mathbf{p}_2^T, \ldots, \mathbf{p}_M^T]^T$. Then the covariance matrix $\mathbf{P}$ in (24) is constructed as $\mathbf{P} = \mathrm{diag}(\mathbf{p})$.

Furthermore, by considering a flat prior on f, η, and $\{p_{n,m,r}\}_{n=1,m=1,r=1}^{NMR}$, the channel vector $\mathbf{h}$, Doppler frequency f, the covariance matrix $\mathbf{P}$ (or more precisely, its diagonal elements $\mathbf{p}$), and the noise power η can be estimated based on the MAP criterion

$$\max_{\mathbf{h},\mathbf{p},\eta,f} p(\mathbf{h}, \mathbf{p}, \eta, f \mid \mathbf{y}) = \max_{\mathbf{h},\mathbf{p},\eta,f} p(\mathbf{y} \mid \mathbf{h}, \eta, f)\, p(\mathbf{h} \mid \mathbf{p}). \tag{25}$$

By combining (23), (24), and (25), and by taking the negative logarithm of the cost function, the optimization problem formulated in (25) becomes

$$\min_{\mathbf{h},\mathbf{p},\eta,f} \left(d_y M \log\eta + \frac{\left\|\mathbf{y} - \mathbf{\Lambda X h}\right\|^2}{\eta} + \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{r=1}^{R} \log p_{n,m,r} + \sum_{n=1}^{N}\sum_{m=1}^{M}\sum_{r=1}^{R} \frac{\left|h_{n,m,r}\right|^2}{p_{n,m,r}} \right), \tag{26}$$

which can be solved using an alternating approach; at each iteration, one of the parameters $\mathbf{h}$, $\mathbf{p}$, η, and f is updated while keeping the other three fixed. In this way, the single difficult joint optimization problem is divided into 4 simpler separate subproblems. GoSLIM-V keeps iterating until a predefined iteration number is reached. Under mild conditions, the cyclic optimization scheme guarantees that the GoSLIM-V algorithm converges, at least to a local minimum of (26) [24].

The 5 steps of the GoSLIM-V algorithm at the tth iteration are outlined below.

(1) Given $\mathbf{h}^{(t-1)}$, the optimal $\mathbf{P}^{(t)}$ that minimizes the cost function in (26) is given by

$$p_{n,m,r}^{(t)} = \left|h_{n,m,r}^{(t-1)}\right|^2, \tag{27}$$

for $n = 1, \ldots, N$, $m = 1, \ldots, M$, and $r = 1, \ldots, R$. For better numerical stability, we set $p_{n,m,r}^{(t)}$ (or equivalently $h_{n,m,r}^{(t)}$) to zero if $p_{n,m,r}^{(t)} < 10^{-15}$.

(2) Once $\mathbf{P}^{(t)}$ is available, the CIR is updated as

$$\mathbf{h}^{(t)} = \left[\mathbf{X}^H\mathbf{X} + \eta^{(t-1)}\left(\mathbf{P}^{(t)}\right)^{-1}\right]^{-1}\left(\mathbf{\Lambda}^{(t-1)}\mathbf{X}\right)^H\mathbf{y}. \tag{28}$$

While inverting $\mathbf{P}^{(t)}$, its zero diagonal entries are removed, and the associated columns in $\mathbf{X}$ are discarded.

(3) Next, using the most recently obtained $\{\mathbf{h}_m^{(t)}\}$ in (28), we estimate the Doppler frequency f. For ease of exposition, we denote $z_m^{(t)}(i) = y_m^*(i)\breve{x}_m^{(t)}(i)$, where $y_m(i)$ and $\breve{x}_m^{(t)}(i)$ represent, respectively, the ith element of the measurement vector $\mathbf{y}_m$ and $\breve{\mathbf{x}}_m^{(t)}$ with $\breve{\mathbf{x}}_m^{(t)} = \overline{\mathbf{X}}\mathbf{h}_m^{(t)}$, $i = 1, \ldots, d_y$. It is easy to verify that

$$\left\|\mathbf{y} - \mathbf{\Lambda}\mathbf{X}\mathbf{h}^{(t)}\right\|^2 = \text{const} - 2\,\text{Re}\left[\sum_{i=1}^{d_y}\left(\sum_{m=1}^{M} z_m^{(t)}(i)\right)e^{-2j\pi f T_s(i-1)}\right]. \tag{29}$$

Since the constant term in (29) is not a function of f, minimizing the cost function in (26) is equivalent to solving

$$f^{(t)} = \arg\max_f \text{Re}\left[\sum_{i=1}^{d_y}\left(\sum_{m=1}^{M} z_m^{(t)}(i)\right)e^{-2j\pi f T_s(i-1)}\right]. \tag{30}$$

Since the summation term within the parenthesis above is nothing but the discrete-time Fourier transform (DTFT) of the sequence $\{\sum_{m=1}^{M} z_m^{(t)}(i)\}_{i=1}^{d_y}$ evaluated at frequency f, $f^{(t)}$ is obtained as the location of the dominant peak of the real part of the DTFT.

(4) Using the $\mathbf{h}^{(t)}$ and $\mathbf{\Lambda}^{(t)}$ most recently obtained via (28) and (30), respectively, we finally estimate the noise power as

$$\eta^{(t)} = \frac{1}{d_y M}\left\|\mathbf{y} - \mathbf{\Lambda}^{(t)}\mathbf{X}\mathbf{h}^{(t)}\right\|^2. \tag{31}$$

(5) Set $t = t + 1$. Go back to Step 1 if t is less than a predefined iteration number, or terminate otherwise.

In the training-directed mode, the channel characteristics in general are not available a priori. In our examples, $\mathbf{h}^{(0)}$ is initialized using the standard matched filter, $f^{(0)}$ is initialized as 0, and the noise power $\eta^{(0)}$ is initialized with a small positive number, for instance, 10^{-10}. Our empirical experience suggests that the GoSLIM-V algorithm does not provide significant performance improvements after 15 iterations or less.

5. Symbol Detection

In this section, we proceed to study the detection of the payload symbols given the estimates of CIRs and Doppler frequency f obtained by GoSLIM-V. The detection task is achieved via two steps: phase compensation followed by turbo equalization. As shown in Figure 1(b), turbo equalization consists of an equalizer and a decoder, both configurated as soft-input soft-output. The decoder is conventionally implemented by the Max-Log-MAP algorithm [17], and our focus herein is on the soft-input soft-output equalizer. We first formulate the symbol detection problem and then describe the phase compensation procedure. After that, we elaborate the LMMSE-based turbo equalization design and discuss its low complexity approximation.

5.1. Problem Formulation. Treating the transmitted symbols as the unknowns and the CIRs and Doppler frequency as known, the measurement vector in (14) can be expressed as [8, 21]

$$\mathbf{y}_m(k) = \widehat{\mathbf{\Lambda}}(k)\sum_{n=1}^{N}\widehat{\mathbf{H}}_{n,m}\mathbf{x}_n(k) + \mathbf{e}_m, \quad m = 1, \ldots, M, \tag{32}$$

where the estimated CIR matrix $\widehat{\mathbf{H}}_{n,m} \in \mathbb{C}^{R\times(2R-1)}$ is given by

$$\widehat{\mathbf{H}}_{n,m} = \begin{bmatrix} \widehat{h}_{n,m,R} & \cdots & \widehat{h}_{n,m,1} & & \mathbf{0} \\ \vdots & \ddots & & \ddots & \vdots \\ \mathbf{0} & & \widehat{h}_{n,m,R} & \cdots & \widehat{h}_{n,m,1} \end{bmatrix}, \tag{33}$$

for $n = 1, \ldots, N$ and $m = 1, \ldots, M$. The entry $\widehat{h}_{n,m,r}$ here represents the estimate of $h_{n,m,r}$ in (17) given by GoSLIM-V at the conclusion of the iteration. Also,

$$\mathbf{x}_n(k) = [x_n(k-R+1), \ldots, x_n(k), \ldots, x_n(k+R-1)]^T, \quad n = 1, \ldots, N, \tag{34}$$

$$\mathbf{y}_m(k) = [y_m(k), \ldots, y_m(k+R-1)]^T, \quad m = 1, \ldots, M. \tag{35}$$

The variable k represents the time index corresponding to the payload symbols of current interest. Although $\mathbf{y}_m$ represents different portions of the received signal in (15), (21), and (35), its use should be clear from the context. Per the discussions following (18), once $\widehat{f}$ is available, the estimated Doppler shift matrix $\widehat{\mathbf{\Lambda}}(k)$ in (32) can be constructed as

$$\widehat{\mathbf{\Lambda}}(k) = \text{diag}\left(\left[e^{-2j\pi\widehat{f}T_s(k-1)}, \ldots, e^{-2j\pi\widehat{f}T_s(k+R-2)}\right]\right). \tag{36}$$

When detecting symbols, we use the estimates $\{\widehat{\mathbf{h}}_{n,m}\}$ and $\widehat{f}$ obtained from the previous channel update and we treat $\{\widehat{\mathbf{H}}_{n,m}\}$ and $\widehat{\mathbf{\Lambda}}(k)$ in (32) as known.

5.2. Phase Compensation. Stacking up all the measurements, (32) can be written as

$$\begin{bmatrix} \mathbf{y}_1(k) \\ \vdots \\ \mathbf{y}_M(k) \end{bmatrix} = \mathring{\mathbf{\Lambda}}(k)\sum_{n=1}^{N}\begin{bmatrix} \widehat{\mathbf{H}}_{n,1} \\ \vdots \\ \widehat{\mathbf{H}}_{n,M} \end{bmatrix}\mathbf{x}_n(k) + \begin{bmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_M \end{bmatrix}, \tag{37}$$

or equivalently as

$$\mathbf{y}(k) = \mathring{\mathbf{\Lambda}}(k)\sum_{n=1}^{N}\widehat{\mathbf{H}}_n\mathbf{x}_n(k) + \mathbf{e} = \mathring{\mathbf{\Lambda}}(k)\,\widehat{\mathbf{H}}\mathbf{x}(k) + \mathbf{e}, \tag{38}$$

where $\mathbf{y}(k)$ and $\mathbf{e} \in \mathbb{C}^{MR\times 1}$, $\mathring{\mathbf{\Lambda}}(k) = \mathbf{I}_{M\times M} \otimes \widehat{\mathbf{\Lambda}}(k)$, $\{\widehat{\mathbf{H}}_n\}_{n=1}^{N} \in \mathbb{C}^{MR\times(2R-1)}$, $\widehat{\mathbf{H}} = [\widehat{\mathbf{H}}_1, \ldots, \widehat{\mathbf{H}}_N]$, and $\mathbf{x}(k) = [\mathbf{x}_1^T(k), \ldots, \mathbf{x}_N^T(k)]^T$. The phase compensation task is simply achieved by multiplying $[\mathring{\mathbf{\Lambda}}(k)]^H$ to both sides of (38), yielding

$$\mathring{\mathbf{y}}(k) = \widehat{\mathbf{H}}\mathbf{x}(k) + \mathring{\mathbf{e}}, \tag{39}$$

where $\mathring{\mathbf{y}}(k) = [\mathring{\mathbf{\Lambda}}(k)]^H\mathbf{y}(k)$ and $\mathring{\mathbf{e}} = [\mathring{\mathbf{\Lambda}}(k)]^H\mathbf{e}$. Given $\mathbf{e} \sim \mathcal{CN}(\mathbf{0}, \eta\mathbf{I})$, $\mathring{\mathbf{e}}$ still has the distribution of $\mathcal{CN}(\mathbf{0}, \eta\mathbf{I})$ since $[\mathring{\mathbf{\Lambda}}(k)]^H$ is unitary.

Phase compensation, along with the aforementioned temporal resampling process, effectively converts the original double-selective channel to an ISI channel. Given the phase-compensated measurement vector $\mathring{\mathbf{y}}(k)$, the estimated CIR matrix $\widehat{\mathbf{H}}$, and $\{L_d(c_n(k))\}_{n=1,k=1}^{N2K}$, we consider using an LMMSE-based soft-input soft-output equalizer to compute the *a posteriori extrinsic* information of the transmitted signal.

5.3. LMMSE-Based Soft-Input Soft-Output Equalizer. As shown in Figure 2, an LMMSE-based soft-input soft-output equalizer can be functionally divided into four modules. The *a priori* LLR preprocessor calculates the mean and the variance of each QPSK payload symbol $x_n(k)$, denoted as $\overline{x}_n(k)$ and $v_n(k)$, respectively, from the *a posteriori extrinsic* information $L_d(c_n(2k-1))$ and $L_d(c_n(2k))$ generated by the decoder for $n = 1, \ldots, N$ and $k = 1, \ldots, K$. Next, the transmitted symbol $x_n(k)$ is estimated via LMMSE filtering given $\mathring{\mathbf{y}}(k)$ and $\widehat{\mathbf{H}}$ in (39), along with $\{\overline{x}_n(k)\}$ and $\{v_n(k)\}$. Specifically, as demonstrated in Figure 2, the LMMSE filter is applied to the residual signal generated by subtracting out the so-called soft interferences from the phase-compensated measurements. The soft interferences characterize the contributions of all the payload symbols except $x_n(k)$, the one of the current interest, in terms of soft information. Based on the symbol estimates $\widehat{x}_n(k)$, the *a posteriori* LLR generator provides the *extrinsic* LLR outputs $L_e(c_n(2k-1))$ and $L_e(c_n(2k))$ ($n = 1, \ldots, N$, $k = 1, \ldots, K$), which will be fed into the soft-input soft-output decoder as *a priori* LLR; see Figure 1(b). In the following, these modules will be elaborated further.

5.3.1. A Priori LLR Preprocessor. In this task, we calculate $\overline{x}_n(k)$ and $v_n(k)$ from $L_d(c_n(2k-1))$ and $L_d(c_n(2k))$. According to the definitions, the mean and variance of $x_n(k)$ are given by [14]

$$\overline{x}_n(k) = \sum_{i=1}^{4} \alpha_i \cdot P\left(x_n(k) = \alpha_i\right), \tag{40}$$

$$\begin{aligned} v_n(k) &= \sum_{i=1}^{4} \left|\alpha_i\right|^2 \cdot P\left(x_n(k) = \alpha_i\right) - \left|\overline{x}_n(k)\right|^2 \\ &= 1 - \left|\overline{x}_n(k)\right|^2, \end{aligned} \tag{41}$$

where $\{\alpha_i\}_{i=1}^{4}$ denote the four QPSK constellation points of $\mathcal{S}$ and $|\alpha_i| = 1$ for $i = 1, \ldots, 4$; see the definition of $\mathcal{S}$ after (1). One can see from (41) that $v_n(k)$ depends on $\overline{x}_n(k)$, and the evaluation of $\overline{x}_n(k)$ in (40) requires $P(x_n(k) = \alpha_i)$ for $i = 1, \ldots, 4$. Since the interleaved bits $\{c(k)\}$ can be reasonably assumed to be independent of each other due to the employment of the random interleaver (see Figure 1(a)), $P(x_n(k) = \alpha_i)$ is determined as the product of the probabilities of the two interleaved bits that map to α_i. For instance, $c_n(2k-1) = 0$ and $c_n(2k) = 1$ map to $x_n(k) = (-1 + j)/\sqrt{2}$ according to (1), and, therefore, $P(x_n(k) = (-1 + j)/\sqrt{2}) = P(c_n(2k-1) = 0) \cdot P(c_n(2k) = 1)$, where for a generic interleaved bit $c_n(q)$ we have

$$\begin{aligned} P\left(c_n(q) = 0\right) &= \frac{e^{L_d(c_n(q))}}{1 + e^{L_d(c_n(q))}}, \\ P\left(c_n(q) = 1\right) &= \frac{1}{1 + e^{L_d(c_n(q))}}, \end{aligned} \tag{42}$$

for $n = 1, \ldots, N$ and $q = 1, \ldots, 2K$. Equation (42) follows from (2) and $P(c_n(q) = 0) + P(c_n(q) = 1) = 1$.

Plugging (42) into (40) gives

$$\begin{aligned} \overline{x}_n(k) = \frac{1}{\sqrt{2}} \Bigg\{ & j \tanh\left[\frac{L_d\left(c_n(2k-1)\right)}{2}\right] \\ & + \tanh\left[\frac{L_d\left(c_n(2k)\right)}{2}\right]\Bigg\}, \\ & n = 1, \ldots, N,\ k = 1, \ldots, K, \end{aligned} \tag{43}$$

which, combined with (41), yields $v_n(k)$.

5.3.2. LMMSE Filtering. Depending on whether the *a priori* LLR information is incorporated or not, two types of LMMSE filters are studied in the following.

In the Absence of A Priori Knowledge. The equalizer is performed in the absence of *a priori* knowledge at the very first iteration before using the decoder. This scenario amounts to setting $L_d(c_n(k)) = 0$ for $n = 1, \ldots, N$ and $k = 1, \ldots, 2K$, which implies that $\overline{x}_n(k) = 0$ and $v_n(k) = 1$ according to (43) and (41) for $n = 1, \ldots, N$ and $k = 1, \ldots, K$. In this case, the LMMSE filter coefficient vector, denoted as $\mathbf{f}_n$, is given by [5, 25]

$$\mathbf{f}_n = \left(\widehat{\mathbf{H}}\widehat{\mathbf{H}}^H + \widehat{\eta}\mathbf{I}\right)^{-1}\mathbf{s}_n. \tag{44}$$

Here, $\widehat{\eta}$ represents the noise power estimate given by GoSLIM-V at the conclusion of the iteration, $\mathbf{s}_n = [\widehat{\mathbf{h}}_{n,1}^T, \ldots, \widehat{\mathbf{h}}_{n,M}^T]^T$ denotes the steering vector corresponding to $x_n(k)$ in (38) for $n = 1, \ldots, N$, and $\widehat{\mathbf{h}}_{n,m}$ is the estimate of $\mathbf{h}_{n,m}$ defined in (17). An estimate of $x_n(k)$ is obtained by applying $\mathbf{f}_n$ to the phase-compensated measurement vector obtained in (39) as

$$\widehat{x}_n(k) = \mathbf{f}_n^H\mathring{\mathbf{y}}(k), \quad n = 1, \ldots, N. \tag{45}$$

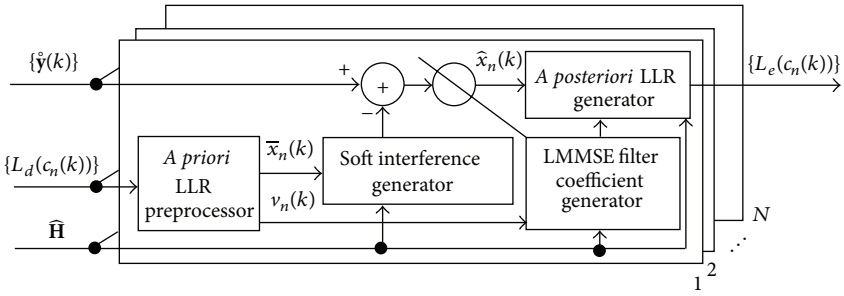


Figure 2: The structure of the LMMSE-based soft-input soft-output equalizer.

In the Presence of A Priori Knowledge. In this case, the LMMSE estimate of $x_n(k)$ is given by [14]

$$\widehat{x}_n(k) = \overline{x}_n(k) + v_n(k)\,\mathbf{f}_n'(k)^H \left[\mathring{\mathbf{y}}(k) - \widehat{\mathbf{H}}E(\mathbf{x}(k))\right], \quad (46)$$

where

$$\mathbf{f}_n'(k) = \left[\widehat{\mathbf{H}}\mathbf{V}(k)\,\widehat{\mathbf{H}}^H + \widehat{\eta}\mathbf{I}\right]^{-1}\mathbf{s}_n \quad (47)$$

represents the LMMSE filter coefficient vector. In (46), each component of $E(\mathbf{x}(k))$ is the expected value of the corresponding component of $\mathbf{x}(k)$ calculated in (40). In (47), the covariance matrix $\mathbf{V}(k) = \mathrm{diag}([\mathbf{v}_1(k)^T, \ldots, \mathbf{v}_N(k)^T])$, and

$$\mathbf{v}_n(k) = [v_n(k-R+1), \ldots, v_n(k), \ldots, v_n(k+R-1)]^T, \quad n = 1, \ldots, N. \quad (48)$$

Each component of $\mathbf{v}_n(k)$ is obtained according to (41).

Equation (46) suggests that the estimation of $x_n(k)$ depends on its own *extrinsic* LLR information $L_d(c_n(2k-1))$ and $L_d(c_n(2k))$, whose impact on $\widehat{x}_n(k)$ comes through $\overline{x}_n(k)$ and $v_n(k)$. From the belief propagation theory point of view, the generation of *extrinsic* information of a payload symbol needs to avoid such dependency [26]. To achieve this goal, we modify (46) as

$$\widehat{x}_n(k) = \mathbf{s}_n^H\left[\widehat{\mathbf{H}}\mathbf{V}(k)\,\widehat{\mathbf{H}}^H + \widehat{\eta}\mathbf{I} + (1 - v_n(k))\,\mathbf{s}_n\mathbf{s}_n^H\right]^{-1} \times \left[\mathring{\mathbf{y}}(k) - \widehat{\mathbf{H}}E(\mathbf{x}(k)) + \overline{x}_n(k)\,\mathbf{s}_n\right]. \quad (49)$$

Compared to (46), the presence of the two additional terms in (49), namely, $(1-v_n(k))\mathbf{s}_n\mathbf{s}_n^H$ and $\overline{x}_n(k)\mathbf{s}_n$, resembles a scenario of $\overline{x}_n(k) = 0$ and $v_n(k) = 1$ (or equivalently, $L_d(c_n(2k-1)) = L_d(c_n(2k)) = 0$) in (46), as if $x_n(k)$ is estimated without incorporating its own LLR information.

Define

$$\mathbf{f}_n''(k) = \left[\widehat{\mathbf{H}}\mathbf{V}(k)\,\widehat{\mathbf{H}}^H + \widehat{\eta}\mathbf{I} + (1 - v_n(k))\,\mathbf{s}_n\mathbf{s}_n^H\right]^{-1}\mathbf{s}_n, \quad (50)$$

$$\dot{\mathbf{y}}_n(k) = \mathring{\mathbf{y}}(k) - \left[\widehat{\mathbf{H}}E(\mathbf{x}(k)) - \overline{x}_n(k)\,\mathbf{s}_n\right]. \quad (51)$$

Then, (49) can be rewritten as

$$\widehat{x}_n(k) = \mathbf{f}_n''(k)^H\dot{\mathbf{y}}_n(k). \quad (52)$$

In (51), the terms within the square brackets correspond to the output of the soft interference generator in Figure 2. To get $\widehat{x}_n(k)$, the LMMSE filter coefficient vector in (50) is applied to the residual measurement vector $\dot{\mathbf{y}}_n(k)$. Note that (52) includes (45) as a special case when no *a priori* knowledge is available, that is, $L_d(c_n(k)) = 0$ for $n = 1, \ldots, N$ and $k = 1, \ldots, 2K$.

5.3.3. A Posteriori LLR Generator. This task calculates the *extrinsic* LLR $L_e(c_n(2k-1))$ and $L_e(c_n(2k))$ from the symbol estimates $\widehat{x}_n(k)$ obtained in (45) or (52) for $n = 1, \ldots, N$ and $k = 1, \ldots, K$.

We assume that, given $x_n(k) = \alpha_i$, $\widehat{x}_n(k)$ is a circularly symmetric i.i.d. complex-valued Gaussian random process, that is, $P(\widehat{x}_n(k) \mid x_n(k) = \alpha_i) \sim \mathcal{CN}(\mu_i, \sigma^2)$, where the mean μ_i and variance σ^2 are calculated, respectively, as $\mu_i = \alpha_i\mathbf{f}_n''(k)^H\mathbf{s}_n$ and $\sigma^2 = \mathbf{f}_n''(k)^H\mathbf{s}_n - \mathbf{f}_n''(k)^H\mathbf{s}_n\mathbf{s}_n\mathbf{f}_n''(k)$ [14]. Under this assumption, the output LLR of the two consecutive bits mapping to $x_n(k)$ is calculated as [14]

$$L_e(c_n(2k-1)) = \frac{\sqrt{8}\,\mathrm{Im}\left(\mathbf{f}_n''(k)^H\dot{\mathbf{y}}_n(k)\right)}{1 - \mathbf{s}_n^H\mathbf{f}_n''(k)}, \qquad L_e(c_n(2k)) = \frac{\sqrt{8}\,\mathrm{Re}\left(\mathbf{f}_n''(k)^H\dot{\mathbf{y}}_n(k)\right)}{1 - \mathbf{s}_n^H\mathbf{f}_n''(k)}, \quad (53)$$

for $n = 1, \ldots, N$ and $k = 1, \ldots, K$.

Let $\mathbf{R}'(k) = \widehat{\mathbf{H}}\mathbf{V}(k)\widehat{\mathbf{H}}^H + \widehat{\eta}\mathbf{I}$ and $\mathbf{R}_n''(k) = \widehat{\mathbf{H}}\mathbf{V}(k)\widehat{\mathbf{H}}^H + \widehat{\eta}\mathbf{I} + (1 - v_n(k))\mathbf{s}_n\mathbf{s}_n^H$. Then $\mathbf{f}_n'(k)$ in (47) and $\mathbf{f}_n''(k)$ in (50) can be rewritten as $\mathbf{f}_n'(k) = \mathbf{R}'(k)^{-1}\mathbf{s}_n$ and $\mathbf{f}_n''(k) = \mathbf{R}_n''(k)^{-1}\mathbf{s}_n$, respectively. One observes that the derivation of $\{\mathbf{f}_n''(k)\}$ requires the inversion of $\mathbf{R}_n''(k)$ for each transmitter at each time index, whereas the computation of $\{\mathbf{f}_n'(k)\}$ needs to invert $\mathbf{R}'(k)$ at each time index. Consequently, by following (47) and (50) directly, the computational complexity of calculating $\{\mathbf{f}_n''(k)\}$ is approximately N times more expensive than obtaining $\{\mathbf{f}_n'(k)\}$.

Since $\mathbf{R}_n''(k) = \mathbf{R}'(k) + (1 - v_n(k))\mathbf{s}_n\mathbf{s}_n^H$, the use of the matrix inversion lemma gives

$$\mathbf{R}_n''(k)^{-1} = \mathbf{R}'(k)^{-1} - \frac{(1 - v_n(k))\,\mathbf{R}'(k)^{-1}\mathbf{s}_n\mathbf{s}_n^H\mathbf{R}'(k)^{-1}}{1 + (1 - v_n(k))\,\mathbf{s}_n^H\mathbf{R}'(k)^{-1}\mathbf{s}_n}. \quad (54)$$

Right multiplying $\mathbf{s}_n$ on both sides of (54) yields

$$\mathbf{f}_n''(k) = \frac{\mathbf{f}_n'(k)}{1 + (1 - \nu_n(k))\,\mathbf{s}_n^H \mathbf{f}_n'(k)}, \tag{55}$$

which, combined with (53), follows

$$\begin{aligned} L_e(c_n(2k-1)) &= \frac{\sqrt{8}\,\mathrm{Im}\left(\mathbf{f}_n'(k)\,\dot{\mathbf{y}}_n(k)\right)}{1 - \nu_n(k)\,\mathbf{s}_n^H \mathbf{f}_n'(k)}, \\ L_e(c_n(2k)) &= \frac{\sqrt{8}\,\mathrm{Re}\left(\mathbf{f}_n'(k)\,\dot{\mathbf{y}}_n(k)\right)}{1 - \nu_n(k)\,\mathbf{s}_n^H \mathbf{f}_n'(k)}. \end{aligned} \tag{56}$$

Complexitywise, the LLR calculation formula in (56) is preferable over (53) since, as we just remarked, it is more efficient to calculate $\{\mathbf{f}_n'(k)\}$ than $\{\mathbf{f}_n''(k)\}$. Due to this reason, LLR is calculated according to (56) in our numerical and experimental examples provided later on.

5.4. Low-Complexity Approximate LMMSE Filtering. Although the calculation of *a posteriori* LLR according to (56) is more efficient than (53), it still constitutes the major computational bottleneck in turbo equalization mainly because $\{\mathbf{f}_n'(k)\}$ needs to be calculated at each time index. To further reduce the computational complexity, we consider a low-complexity approximate LMMSE filter whose coefficient vector is given by [14]

$$\mathbf{f}_n' = \left(\widehat{\mathbf{H}}\,\overline{\mathbf{V}}\,\widehat{\mathbf{H}}^H + \widehat{\eta}\mathbf{I}\right)^{-1} \mathbf{s}_n, \tag{57}$$

where $\overline{\mathbf{V}} = (1/K)\sum_{k=1}^{K} \mathbf{V}(k)$. Since $\{\mathbf{f}_n'\}$ is constant for each transmitter over one payload block (hence the time index k is dropped in (57)), the overall complexity of calculating $\{\mathbf{f}_n'\}$ is approximately K times faster than deriving $\{\mathbf{f}_n'(k)\}$ according to (47). Substituting $\mathbf{f}_n'(k)$ in (56) with $\mathbf{f}_n'$ yields

$$\begin{aligned} L_e(c_n(2k-1)) &= \frac{\sqrt{8}\,\mathrm{Im}\left(\mathbf{f}_n'\dot{\mathbf{y}}_n(k)\right)}{1 - \nu_n(k)\,\mathbf{s}_n^H \mathbf{f}_n'}, \\ L_e(c_n(2k)) &= \frac{\sqrt{8}\,\mathrm{Re}\left(\mathbf{f}_n'\dot{\mathbf{y}}_n(k)\right)}{1 - \nu_n(k)\,\mathbf{s}_n^H \mathbf{f}_n'}. \end{aligned} \tag{58}$$

We hereafter refer to the turbo equalization scheme that calculates the *a posteriori extrinsic* information according to (56) and (58) as exact-LMMSE turbo and approximate-LMMSE turbo, respectively.

Note that matrix inversion is an indispensable stage in calculating the LMMSE filter coefficients in (44), (47), and (57). To expedite the calculation, we can make use of the conjugate gradient (CG) method and fast Fourier transform (FFT) operations, as elaborated in [27]. Although [27] focuses on the efficient calculation of the LMMSE filter coefficients in the form of (44), the extension to a more general scenario in (47) or (57) is straightforward. In the present paper, both exact-LMMSE turbo and approximate-LMMSE turbo are implemented using the FFT-based CG method.

6. Numerical and Experimental Results

6.1. Numerical Results. Consider transmitting four payload blocks simultaneously over time-invariant ISI channels using a MIMO UAC system equipped with $N = 4$ transmitters and $M = 12$ receivers. Block length is fixed at $K = 250$. The four payload blocks across the $N = 4$ transmitters are constructed from a randomly generated binary source sequence of length $NK = 1000$ according to the procedure detailed in Section 2. We simulate $N \times M = 48$ frequency-selective channels involved in the MIMO UAC system. To resemble practical UAC scenarios, these simulated CIRs are estimated from MACE10 in-water experimental data and each CIR has $R = 50$ taps. CIRs have been normalized to 1, that is, $\|\mathbf{h}_{n,m}\|^2 = 1$ for $n = 1, \ldots, N$ and $m = 1, \ldots, M$. The received data samples are then constructed according to (14). Since Doppler effects are not considered in this example, $\overline{\mathbf{\Lambda}} = \mathbf{I}$. The noise vector $\{\mathbf{e}_m\}$ is assumed to contain circularly symmetric i.i.d. complex-valued Gaussian random variables with zero-mean and variance σ^2. The simulation of ISI channels, combined with the assumption that each receiver has perfect knowledge on the channel characteristics $\{\mathbf{h}_{n,m}\}$, suggests that we can bypass the temporal resampling, channel estimation, and phase compensation modules in Figure 1(b) and apply exact-LMMSE turbo, approximate-LMMSE turbo, and RELAX-BLAST directly to the received measurements. Figures 3(a) and 3(b) show the average coded bit error rate (BER) given by exact-LMMSE turbo and approximate-LMMSE turbo, respectively, along with the RELAX-BLAST performance at different SNRs, where SNR is defined as $1/\sigma^2$. Each point is averaged over 500 Monte-Carlo trials. The binary source sequence and the noise pattern vary from one trial to another. The curve labeled as "No Iteration" is obtained by employing the equalizer and the decoder only once, that is, the feedback loop is yet to be formed. In addition, the average coded BER given by RELAX-BLAST is obtained after three iterations. We can see from Figure 3 that both types of turbo equalization schemes effectively reduce the coded BER as the iteration proceeds and significantly outperform RELAX-BLAST, and exact-LMMSE turbo provides only slightly better detection performance than approximate-LMMSE turbo. Complexity wise, the average time required to finish one trial is 18.64 s, 0.49 s, and 0.19 s on an ordinary workstation (Intel Xeon E5506 processor 2.13 GHz, 12 GB RAM, Windows 7 64-bit, and MATLAB R2010b) for exact-LMMSE turbo, approximate-LMMSE turbo, and RELAX-BLAST, respectively. Consequently, approximate-LMMSE turbo is preferred over its exact-LMMSE turbo counterpart since the former provides almost the same detection performance as the latter but with a computational complexity on the same order as RELAX-BLAST.

6.2. MACE10 In-Water Experimental Results

6.2.1. Experiment Specifics. The MACE10 in-water experiment was conducted by the Woods Hole Oceanographic Institution (WHOI) off the coast of Martha's Vineyard, MA, USA, in June 2010. A source array consisting of 4 transducers

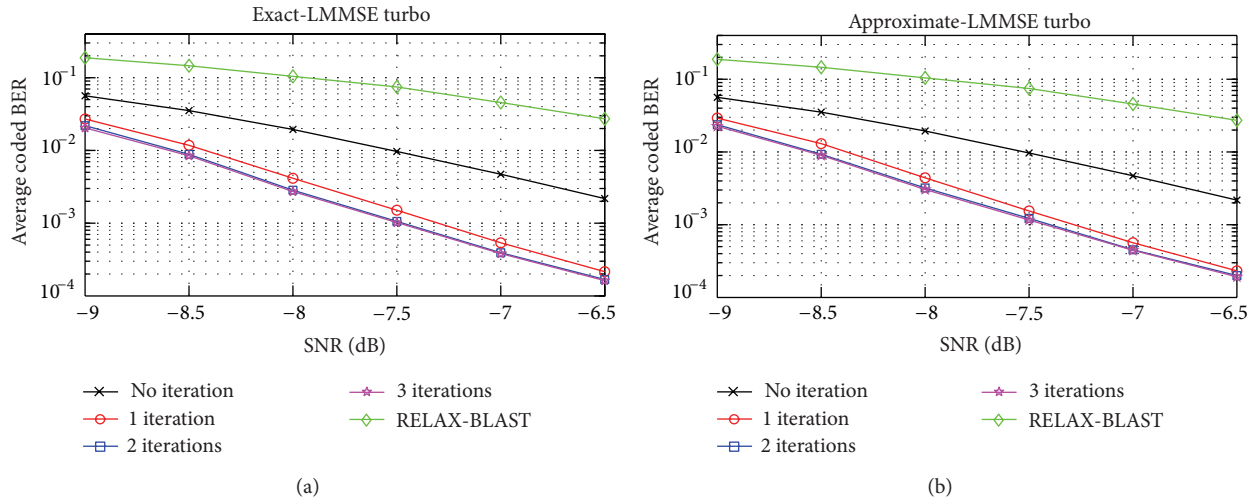


FIGURE 3: (a) Coded BER performance by using exact-LMMSE turbo along with RELAX-BLAST performance. (b) Coded BER performance by using approximate-LMMSE turbo along with RELAX-BLAST performance. Each point is averaged over 500 Monte Carlo trials. In this simulation, $N = 4$, $M = 12$, $R = 50$, and $K = 250$.

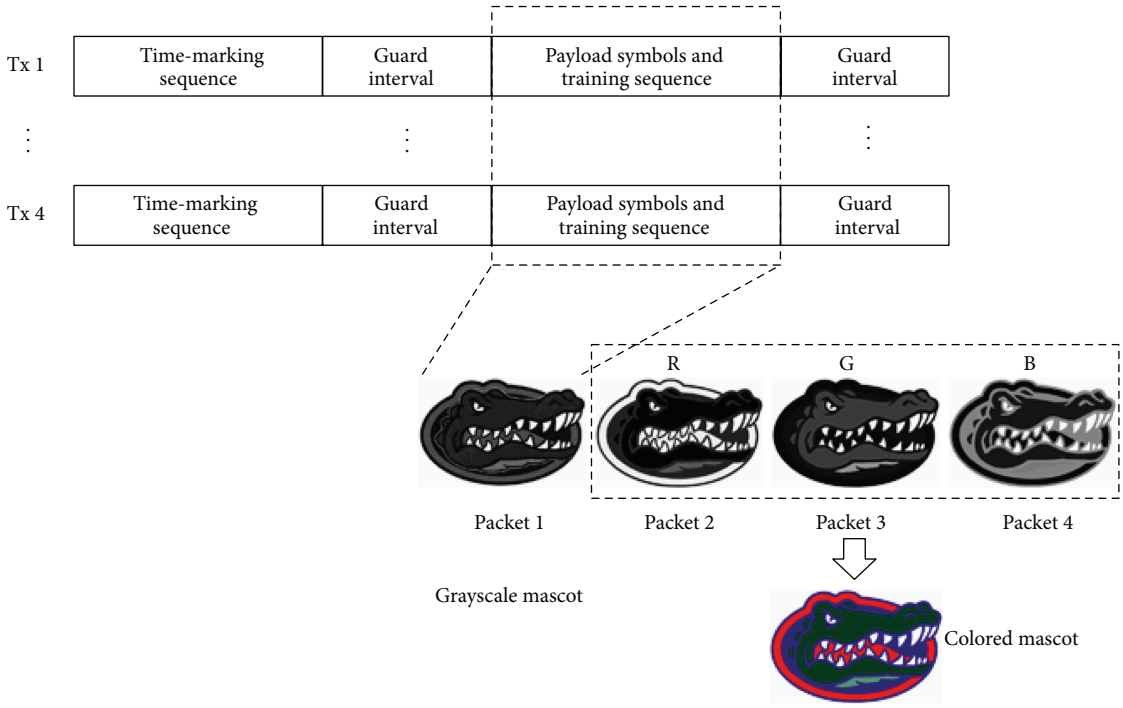


FIGURE 4: The structure of the package used in the MACE10 experiment.

was vertically deployed at a depth of 80 m and towed by a vessel. At the receiver side, a 12-element hydrophone array was mounted on a buoy. The vessel moved from the minimum range of 500 m away from the receiving array outbound to the maximum range of 4000 m and then inbound back to the minimum range. The carrier frequency, sampling frequency, and symbol rate employed in the MACE10 experiment were 13 kHz, 39.0625 kHz, and 3.90625 kHz, respectively. By transmitting $N = 4$ sequences simultaneously and incorporating the measurements acquired from all of the $M = 12$ receiver elements for analysis, we established a 4×12 MIMO UAC system.

The structure of a transmitted data package is shown in Figure 4. Each package consists of 4 packets. The first packet conveys a grayscale Gator mascot and the subsequent 3 packets are combined from a colored mascot. The RGB components of the colored image were transmitted in the 2nd, 3rd, and 4th packets, respectively. Each pixel of the Gator grayscale image is represented by 5 bits, corresponding to 32 different intensities (e.g., pure white and pure dark

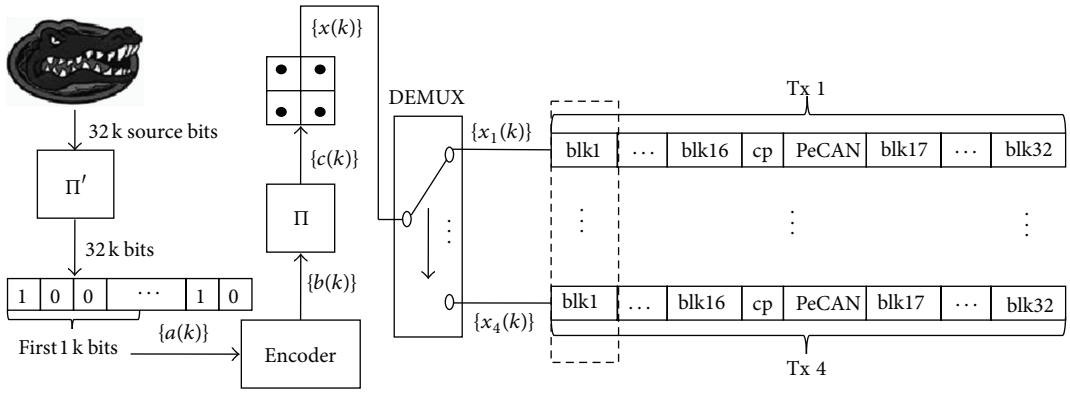


FIGURE 5: The structure of the transmitted symbols for the 4 × 12 MIMO BLAST scheme used in MACE10.

pixels are represented by 11111 and 00000, resp.). The 64-pixel by 100-pixel grayscale mascot image, as a consequence, is represented by a total of 32 k source bits. Accordingly, a colored mascot image is represented by 96 k bits. The contrast of the grayscale image, as well as the hue of the colored image, has been carefully adjusted so that the image carries approximately equal numbers of 1s and 0s.

As shown in Figure 4, each packet is constructed as follows: time-marking sequences are placed at the beginning of each packet to facilitate the temporal resampling procedure; two guard intervals, each containing 500 silent symbols, are placed, respectively, before and after the segments containing the payload symbols and training sequences. The payload symbols contain the information of the Gator mascot image. We herein elaborate how to generate the 1st packet from the grayscale Gator mascot image (the packet generation for each of the RGB components of the colored image follows the same procedure). Specifically, the 32 k source bits are first interleaved so that the bits fed into the convolutional encoder module have an equal chance of being 0 or 1; see Figure 5. The so-obtained 32 k interleaved source bits are then divided into 32 groups, each containing 1 k bits. The bits in the ith group ($i = 1, \ldots, 32$) will be used to construct the ith payload symbol block across the 4 transmitted sequences, and the construction procedure follows Figure 1(a). Note that in Figure 5, the depth of the interleavers Π' and Π is 32 k and 2 k, respectively. Figure 5 illustrates a scenario with $i = 1$. The shifted PeCAN training sequences with length $P = 512$, in conjunction with $L_{\mathrm{CP}} = 99$ cyclic prefix symbols, form the training section, which is located between the 16th and 17th payload blocks. This MIMO UAC design leads to a net coded data rate of 11.7 kbps. The data package was transmitted periodically and recorded by the receiver array. A total of 120 epochs were available and they are referred to as "E001"–"E120," respectively.

To estimate the Doppler scaling factor, we treat the time-marking sequences at the beginning of a packet as its preamble and those at the beginning of the subsequent packet as the postamble. Take the 2nd packet of epoch "E002," for example. For the channel between the 1st transmitter and the 1st receiver, the superimposed modulus of the CIRs obtained by GoSLIM-V from the preamble and postamble is shown in Figure 6. The indexes of the principal arrivals for the preamble and postamble are 12 and 21, respectively. Hence, the time duration change imposed on the packet is $\widehat{T}_d = (21 - 12)T_s$, where T_s is the symbol period defined after (18). Then the Doppler scaling factor $\widehat{\beta}$ can be estimated according to (12).

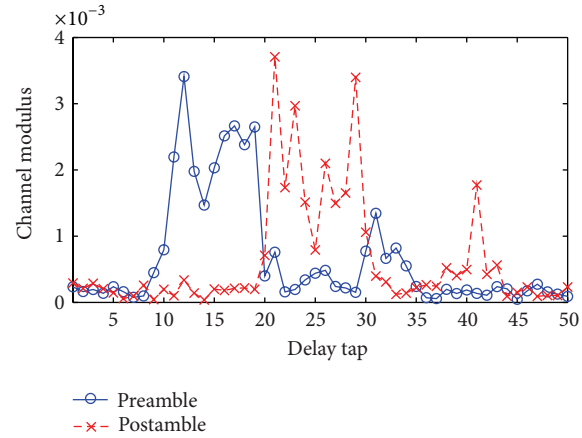


FIGURE 6: The superimposed modulus of the CIRs obtained from the preamble and postamble, respectively.

To assess the performance of the resampling process, the CIR and Doppler frequency evolutions obtained by GoSLIM-V before we resample the 2nd packet of epoch "E002" are shown in Figures 7(a) and 7(c), respectively. In comparison, Figures 7(b) and 7(d) demonstrate the corresponding CIR and Doppler frequency evolutions obtained after resampling the packet, respectively. We can see from Figure 7 that the temporal resampling procedure successfully reduces the Doppler scaling effects to Doppler frequency shifts. The relative speed between the transmitter and the receiver arrays can be estimated as $\widehat{v} = (\widehat{\beta} - 1) \cdot c$, using a common underwater sound speed of $c = 1500$ m/s. It is interesting to look at Figure 8 where the vessel speed estimated during the resampling stage is plotted on top of the GPS reference information provided by WHOI (the GPS device was equipped on the moving vessel). The good agreement between these two

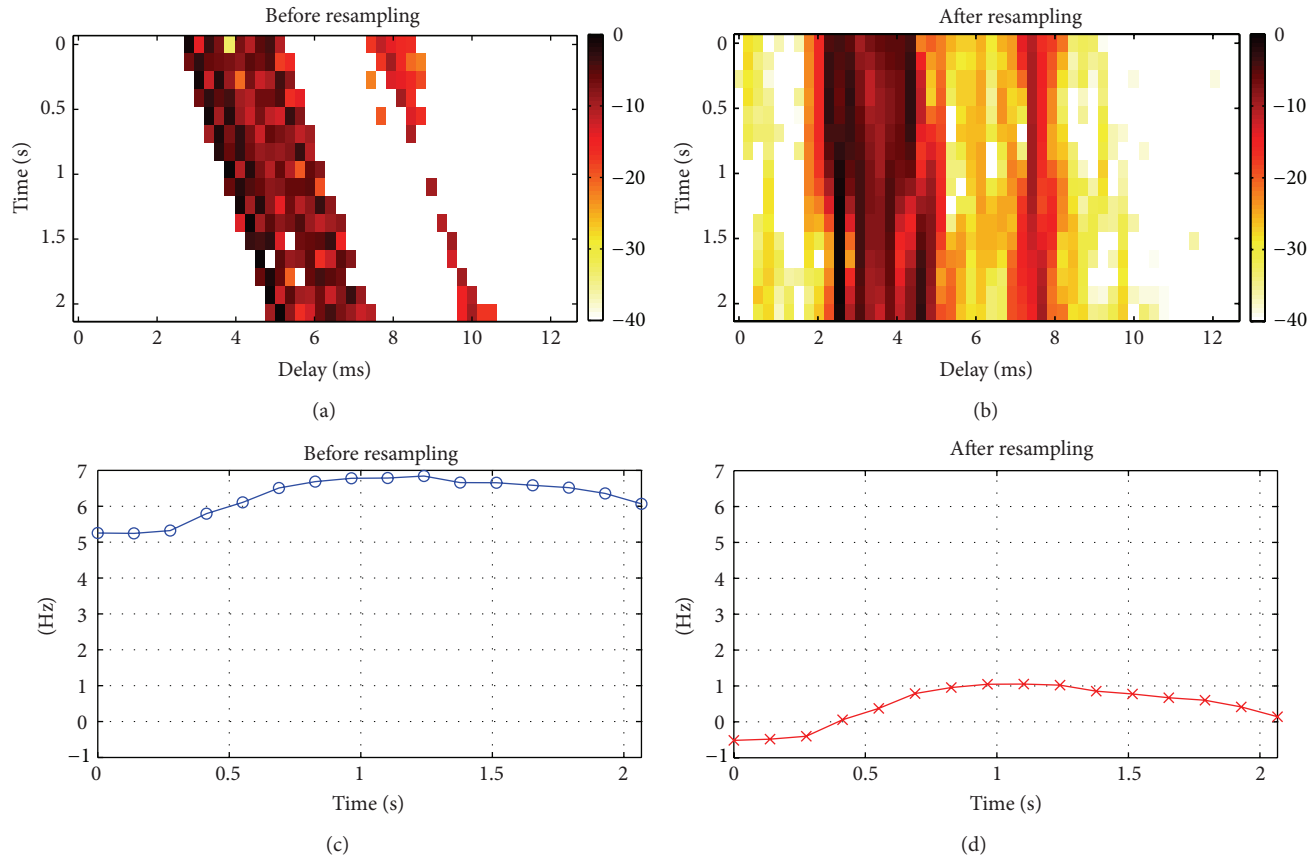


FIGURE 7: (a) CIR evolution of epoch "E002" before resampling. (b) CIR evolution of epoch "E002" after resampling. (c) Doppler frequency evolution of epoch "E002" before resampling. (d) Doppler frequency evolution of epoch "E002" after resampling.

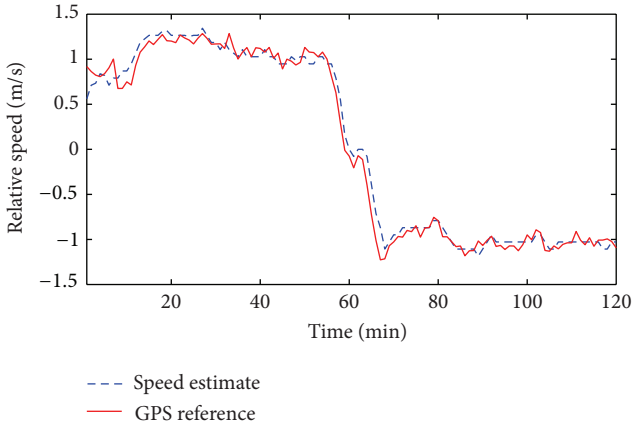
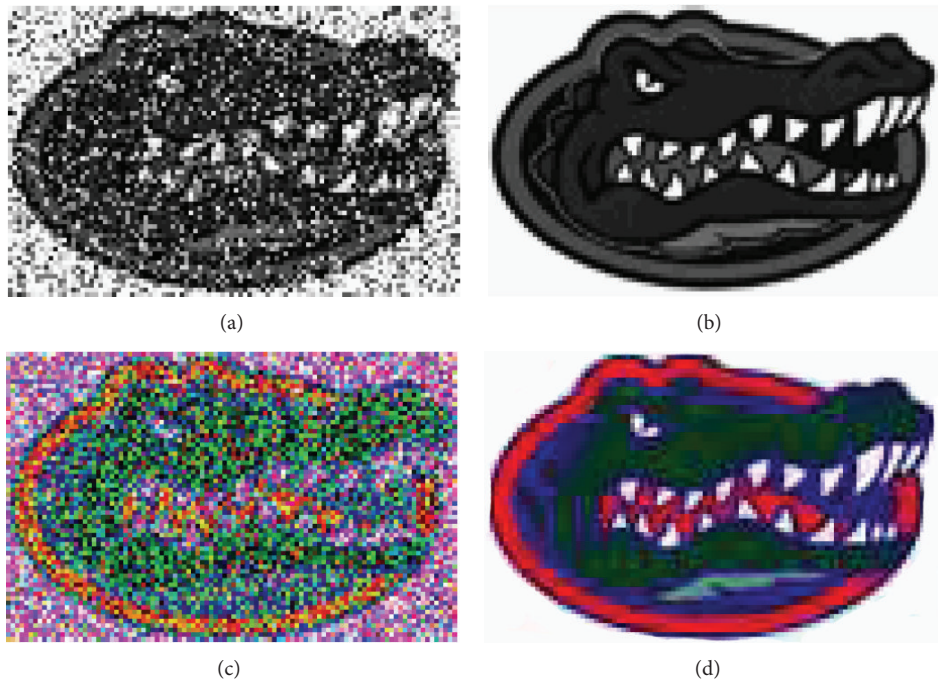


FIGURE 8: The relative speed between the transmitter and receiver array given by GPS and estimated during the temporal resampling stage (courtesy of Milica Stojanovic's group).

curves verifies the effectiveness of the resampling procedure we employ. The analysis presented hereafter is based on the resampled measurements.

6.2.2. Performance Evaluation. By fixing the tracking length at $L_{TR} = 450$, the channel tracking starts with training-directed channel estimation using GoSLIM-V. Then we perform phase compensation on the received measurements, followed by the detection of the first $K = 250$ payload symbols contained in the 17th payload block for each transmitted sequence using RELAX-BLAST, exact-LMMSE turbo, and approximate-LMMSE turbo; see Figure 5. Next, the channels are updated in the decision-directed mode using $L_{TR} = 450$ symbols (containing the previously detected payload symbols, as well as a portion of the training sequence as well). With the updated CIRs and Doppler frequency, after phase compensation, the subsequent $K = 250$ payload symbols contained in the 18th block are estimated using the same symbol detection scheme. This process continues until all of the 16 payload blocks to the right-hand side of the training sequences are detected. This same tracking scheme can be applied in a reverse manner to the detection of the 16 payload blocks ahead of the training sequences.

We deem a packet to be successfully detected if the resulting coded BER is less than 0.1. After analyzing a total of 480 packets available, Table 1 summarizes the successfully detected packet percentage, the zero BER packet percentage, the coded BER averaged over the successful packets, and the time ratio of the time consumed to process a packet on the workstation specified in Section 6.1 to $T_{tx} = 2.741$ s (T_{tx} is defined in (12)) obtained using exact-LMMSE



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

Figure 9: (a) Grayscale mascot recovered from epoch "E054" using RELAX-BLAST. (b) Grayscale mascot recovered from epoch "E054" using turbo equalization. (c) Colored mascot recovered from epoch "E054" using RELAX-BLAST. (d) Colored mascot recovered from epoch "E054" using turbo equalization.

Table 1: A summary of the performance of the three detection schemes (3 iterations applied).

	Successful packet percentage (%)	Zero BER packet percentage (%)	Average coded BER	Time ratio
Exact-LMMSE turbo	100	76.7	9.2×10^{-5}	488.1
Approximate-LMMSE turbo	100	74.4	2.1×10^{-4}	17.2
RELAX-BLAST	82.5	4.8	1.6×10^{-2}	16.9

turbo, approximate-LMMSE turbo, and the RELAX-BLAST scheme, respectively. The results are obtained by applying 3 iterations for all of the three types of detection schemes considered. One observes from Table 1 that (1) BER-wise, both exact-LMMSE turbo and approximate-LMMSE turbo outperform RELAX-BLAST significantly, (2) compared to exact-LMMSE turbo, approximate-LMMSE turbo greatly reduces the computational time at the cost of slight BER performance degradation, and (3) compared to RELAX-BLAST, approximate-LMMSE turbo improves the BER performance by two orders of magnitude without significantly increasing the computational complexities. These observations are in line with those made from the numerical examples in Section 6.1. Moreover, we analyze epoch "E054" that leads to perfect recovery of both the grayscale and colored mascots (see Figures 9(b) and 9(d)) using either exact-LMMSE Turbo or approximate-LMMSE turbo. In comparison, the grayscale and colored mascots recovered from epoch "E054" using RELAX-BLAST are shown in Figures 9(a) and 9(c), respectively, with the corresponding coded BERs being 1.8×10^{-1} and 1.5×10^{-1}. We note that the turbo equalization schemes are highly effective.

To further illustrate the detection performance of turbo equalization, Table 2 shows the coded BER averaged over all of the 480 packets at different iteration numbers obtained by exact-LMMSE turbo and approximate-LMMSE turbo. We can see from Table 2 that the coded BER improves with iteration. Empirical experience indicates that the detection performance for both types of turbo equalization converges after three iterations. Next, we choose one payload block and denote $\{L_d(a(k))\}_{k=1}^{1000}$ as the LLR soft information of the corresponding 1 k source bits $\{a(k)\}_{k=1}^{1000}$ generated by the Max-Log-MAP decoder. Figures 10(a)–10(d) and 10(e)–10(h) show $\{L_d(a(k))\}_{k=1}^{1000}$ obtained by exact-LMMSE turbo and approximate-LMMSE turbo, respectively, at different iteration numbers. $\{\widetilde{a}(k)\}$ in Figure 1(b) are the hard decisions determined from $\{L_d(a(k))\}$. Specifically, if $L_d(a(k)) > 0$ then $\widetilde{a}(k) = 0$; otherwise, $\widetilde{a}(k) = 1$ (see (2)). In Figure 10, the circles indicate bit errors. We can see from Figure 10 that the LLRs of source bits are moving away from zero as the iteration proceeds (the first iteration has the most significant impact), which suggests that with the help of cycling soft information, the decoder is more and more confident about the corresponding source bits being 0 or 1.

Table 2: The average coded BER obtained by exact-LMMSE turbo and approximate-LMMSE turbo, respectively.

	No iteration	1 iteration	2 iterations	3 iterations
Exact-LMMSE turbo	2.7×10^{-1}	8.1×10^{-4}	1.3×10^{-4}	9.2×10^{-5}
Approximate-LMMSE turbo	2.7×10^{-1}	2.2×10^{-3}	3.2×10^{-4}	2.1×10^{-4}

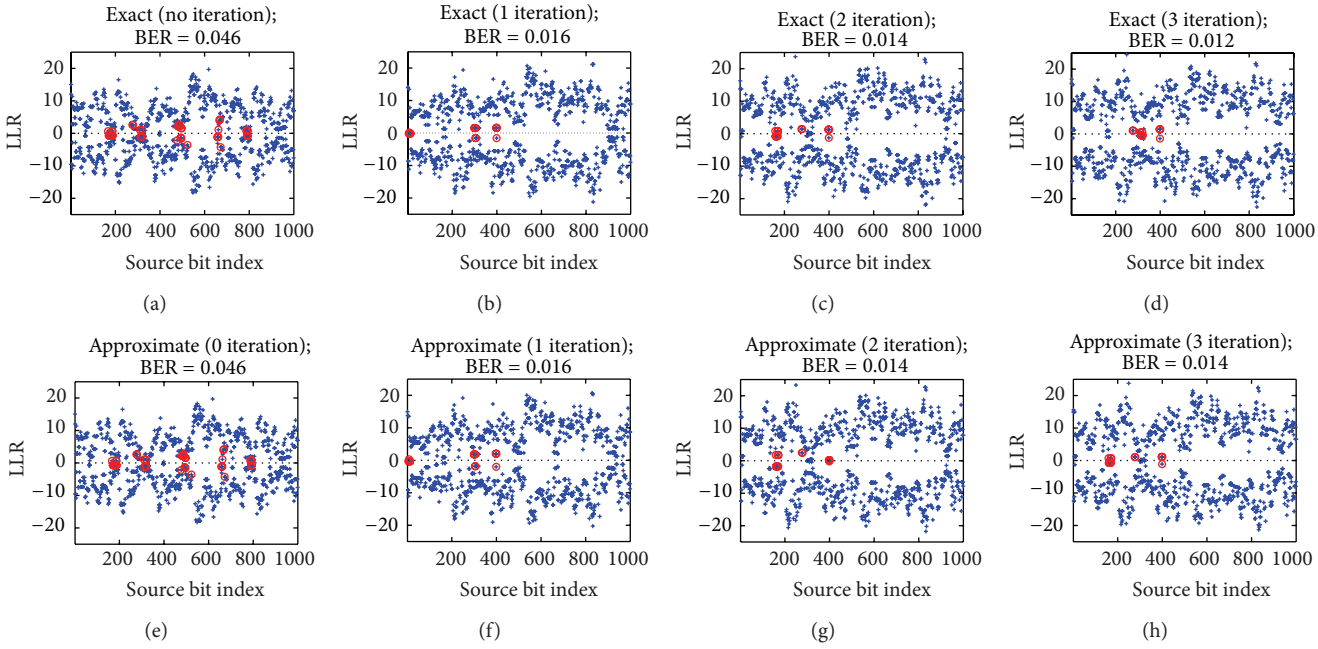


Figure 10: The LLR soft information about the source bits at the output of the decoder. ((a)–(d)) are obtained by exact-LMMSE turbo from no iteration to 3 iterations, respectively. ((e)–(h)) are obtained by approximate-LMMSE turbo from no iteration to 3 iterations, respectively.

7. Conclusions

For double-selective channels encountered in mobile MIMO UAC, we have demonstrated via the MACE10 in-water experimental data analysis that it is reasonable to assume a common Doppler scaling factor imposed on the propagation paths among all the transmitter and receiver pairs when the Doppler effects are mainly induced by the relative motions between the transmitter and receiver arrays. Temporal resampling has been used to effectively convert the Doppler scaling effects to Doppler frequency shifts. A data-adaptive sparse channel estimation algorithm, referred to as the GoSLIM-V algorithm, is used to estimate the underlying CIRs and Doppler frequency in a joint manner. For symbol detection, we have investigated the turbo equalization schemes implemented by the LMMSE-based soft-input soft-output equalizer as well as its low complexity approximation. The latter provides only slightly degraded detection performance but at a significantly lower computational complexity compared to the former and is thus preferred. The effectiveness of the considered approaches has been verified using both numerical and the MACE10 in-water experimental results.

Acknowledgments

This work was supported in part by the Office of Naval Research (ONR) under Grant no. N00014-10-1-0054. The authors gratefully acknowledge WHOI for the fruitful collaborations with them to conduct the in-water experimentations and for sharing data with them.

References

[1] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: recent advances and future challenges," *Marine Technology Society Journal*, vol. 42, no. 1, pp. 103–116, 2008.

[2] D. B. Kilfoyle and A. B. Baggeroer, "State of the art in underwater acoustic telemetry," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 4–27, 2000.

[3] M. Stojanovic, J. A. Catipovic, and J. G. Proakis, "Phase-coherent digital communications for underwater acoustic channels," *IEEE Journal of Oceanic Engineering*, vol. 19, no. 1, pp. 100–111, 1994.

[4] B. Li, S. Zhou, M. Stojanovic, L. L. Freitag, and P. Willett, "Multicarrier communication over underwater acoustic channels with nonuniform Doppler shifts," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 198–209, 2008.

[5] J. Ling, T. Yardibi, X. Su, H. He, and J. Li, "Enhanced channel estimation and symbol detection for high speed multi-input multi-output underwater acoustic communications," *Journal of the Acoustical Society of America*, vol. 125, no. 5, pp. 3067–3078, 2009.

[6] J. Ling, X. Tan, T. Yardibi, J. Li, H. He, and M. L. Nordenvaad, "Enhanced channel estimation and efficient symbol detection in

MIMO underwater acoustic communications," in *Proceedings of the 43rd Asilomar Conference on Signals, Systems and Computers*, pp. 600–604, Pacific Grove, Calif, USA, November 2009.

[7] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, New York, NY, USA, 2005.

[8] W. Li and J. C. Preisig, "Estimation of rapidly time-varying sparse channels," *IEEE Journal of Oceanic Engineering*, vol. 32, no. 4, pp. 927–939, 2007.

[9] A. Song, M. Badiey, and V. K. McDonald, "Multichannel combining and equalization for underwater acoustic MIMO channels," in *Proceedings of the MTS/IEEE Oceans Conference (OCEANS '08)*, pp. 15–21, Quebec, Canada, September 2008.

[10] J. Ling, K. Zhao, J. Li, and M. Lundberg Nordenvaad, "Multi-input multi-output underwater communications over sparse and frequency modulated acoustic channels," *Journal of the Acoustical Society of America*, vol. 130, no. 1, pp. 249–262, 2011.

[11] K. Zhao, J. Ling, and J. Li, "On estimating sparse and frequency modulated channels for MIMO underwater acoustic communications," in *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton '11)*, pp. 453–460, Monticello, Ill, USA, September 2011.

[12] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel," in *Proceedings of the URSI International Symposium on Signals, Systems, and Electronics (ISSSE '98)*, pp. 295–300, October 1998.

[13] J. Li and P. Stoica, "Efficient mixed-spectrum estimation with applications to target feature extraction," *IEEE Transactions on Signal Processing*, vol. 44, no. 2, pp. 281–295, 1996.

[14] M. Tüchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using a priori information," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 673–683, 2002.

[15] M. Tüchler, R. Koetter, and A. C. Singer, "Turbo equalization: principles and new results," *IEEE Transactions on Communications*, vol. 50, no. 5, pp. 754–767, 2002.

[16] R. Koetter, A. C. Singer, and M. Tüchler, "Turbo equalization," *IEEE Signal Processing Magazine*, vol. 21, no. 1, pp. 67–80, 2004.

[17] P. Robertson, E. Villebrun, and P. Hoeher, "Comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proceedings of the IEEE International Conference on Communications*, pp. 1009–1013, June 1995.

[18] J. Ling, H. He, J. Li, W. Roberts, and P. Stoica, "Covert underwater acoustic communications," *Journal of the Acoustical Society of America*, vol. 128, no. 5, pp. 2898–2909, 2010.

[19] B. S. Sharif, J. Neasham, O. R. Hinton, and A. E. Adams, "Computationally efficient Doppler compensation system for underwater acoustic communications," *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 52–61, 2000.

[20] P.-P. J. Beaujean and L. R. LeBlanc, "Adaptive array processing for high-speed acoustic communication in shallow water," *IEEE Journal of Oceanic Engineering*, vol. 29, no. 3, pp. 807–823, 2004.

[21] J. C. Preisig, "Performance analysis of adaptive equalization for coherent acoustic communications in the time-varying ocean environment," *Journal of the Acoustical Society of America*, vol. 118, no. 1, pp. 263–278, 2005.

[22] H. He, D. Vu, P. Stoica, and J. Li, "Construction of unimodular sequence sets for periodic correlations," in *Proceedings of the 43rd Asilomar Conference on Signals, Systems and Computers*, pp. 136–140, Pacific Grove, Calif, USA, November 2009.

[23] T. Yardibi, J. Li, P. Stoica, M. Xue, and A. B. Baggeroer, "Source localization and sensing: a nonparametric iterative adaptive approach based on weighted least squares," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 425–443, 2010.

[24] W. I. Zangwill and B. Mond, *Nonlinear Programming: A Unified Approach*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1969.

[25] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, Wiley, New York, NY, USA, 1949.

[26] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 140–152, 1998.

[27] J. Ling, X. Tan, J. Li, and M. L. Nordenvaad, "Efficient channel equalization for MIMO underwater acoustic communications," in *Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM '10)*, pp. 73–76, Ma'ale Hahamisha, Israel, October 2010.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 592890, 12 pages
http://dx.doi.org/10.1155/2013/592890

Research Article

Continuous Top-k Contour Regions Querying in Sensor Networks

Shangfeng Mo,[1,2,3] Hong Chen,[1,2] Cuiping Li,[1,2] Deying Li,[1,2] and Yinglong Li[1,2,3]

[1] *Key Laboratory of Data Engineering and Knowledge Engineering of MOE, Renmin University of China, Beijing 100872, China*
[2] *School of Information, Renmin University of China, Beijing 100872, China*
[3] *Hunan University of Science and Technology, Xiangtan 411201, China*

Correspondence should be addressed to Hong Chen; chong@ruc.edu.cn

Received 8 January 2013; Revised 24 February 2013; Accepted 9 March 2013

Academic Editor: Zhangbing Zhou

Wireless sensor networks (WSNs) are important parts of Internet of Things or Cyber-Physical Systems (CPS). WSNs can be seen as a new type of distributed database systems. The data query processing is very important for WSNs. In this paper, we proposed a Continuous Top-k Contour Regions Querying algorithm (CTCRQ) which can continuously obtain the top-k contour regions and does not lose the rate of precision (accuracy). We take full advantage of the kth value of top-k result in current round as the threshold to suppress the nodes whose readings do not belong to the top-k result in next round. Extensive experiments are conducted to evaluate the performance of the proposed CTCRQ approach by using a synthetic data set. The results provide a number of insightful observations and show that CTCRQ substantially outperforms Centralized algorithm, Centralized Optimized algorithm, and CCM algorithm in terms of data transmitted.

1. Introduction

With the development of microelectronics, embedded computing, and wireless communication technology, sensor hardware technology is also improved. Low-power, tiny sensor nodes can be integrated with information collection, data processing, wireless communication, and other functions [1]. Wireless sensor networks (WSNs) composed by a large number of sensor nodes are used to collect and process information of perceived objects.

The sensor nodes are usually battery-powered and deployed in harsh physical environments. It is usually impossible to replace the batteries or the nodes. So the goal of querying in wireless sensor networks (WSNs) is to reduce the energy consumption and prolong the network lifetime. Compared with the calculation, the communication between sensor nodes consumed much more energy. For example, executing one instruction needs energy consumption about 0.84 nJ, but the energy consumption of transmitting a sensory data packet is about 0.685 mJ between MICA2 sensor nodes [2]. So the key problem of saving the energy consumption is to reduce the amount of data transmission.

There are many energy-efficient queries in WSNs, for example, top-k querying [3–8], contour regions querying [9–15], aggregate querying [16], event querying [17, 18], and so forth. The traditional continuous top-k querying in WSNs can return the list of k sensor nodes with the highest (or lowest) readings at every sampling period.

To visualize the sensor network regions, we can use the contour mapping. A contour map of an attribute (e.g., temperature) displays the distribution of the attribute value over the topographic regions. Figure 1 shows the contour regions of temperatures in a real volcanic area, Kawah Ijen crater lake [3]. The sink can detect and analyze the environmental events using the contour regions.

There are many continuous contour regions querying schemes for WSNs, including eScan [9], isoline aggregation [10], Iso-Map [11, 12], CCM [13], the literature in [14], and improved Isoline aggregation [15]. These schemes will obtain the overall contour mapping regions. Most of these protocols use approximate algorithms to reduce the data transmitted but may lose the rate of precision (accuracy). In other words, these algorithms obtain approximate contour mapping regions.

It is difficult to achieve the overall contour mapping regions and not lose the rate of precision (accuracy), because sensor nodes have constrained resources and insufficient knowledge. If top-k highest (or lowest) contour regions can satisfy the user's requirements, it will significantly reduce the data transmitted and then save a great number of energy as well as prolong the network lifetime. In practice, as shown in Figure 2, scientists wish to concentrate on studying the most important environmental events, and they can continuously obtain the top-k contour regions with the highest (or lowest) temperatures at every sampling period. In addition, in some specific applications, people are often interested in the most important regions in the network. For example, Traffic Radio Station wants to find the most congested regions to obtain the overall traffic situation and then broadcast the traffic conditions to drivers in a city; scientists need to find the most dense regions of animals gathering in a forest to learn about the animals' living habit status; farmers hope to find the most arid regions in a farm to give priority to irrigate; museum curator wants to know the most dense showcases which visitors gathered to determine the tour route.

In this paper, we focus on the top-k highest (or lowest) contour regions, not the overall contour regions. But the top-k highest (or lowest) contour regions are accurate, not approximate. Our contributions are summarized as follows.

(i) Compared with obtaining the approximate overall contours region, the amount of data transmitted of our obtaining top-k highest (or lowest) contour regions algorithm is less.

(ii) We take full advantage of the kth value of top-k result in current round as the threshold to suppress the nodes whose readings do not belong to the top-k result in next round.

(iii) Extensive experiments are conducted to evaluate the performance of the proposed CTCRQ approach by using a synthetic data set. The results provide a number of insightful observations and show that CTCRQ substantially outperforms Centralized algorithm, Centralized Optimized algorithm, and CCM algorithm in terms of amount of Kbytes (kilo bytes) data transmitted.

The remainder of this paper is organized as follows. Section 2 summarizes related work. Section 3 introduces some assumptions. Section 4 describes the proposed CTCRQ scheme in detail. Section 5 presents experimental results. Section 6 concludes the paper.

2. Related Work

We have studied many querying algorithms in WSNs, but we have not found a Continuous Top-k Contour Regions Querying algorithm as we proposed. There are many top-k querying algorithms [3–8] and continuous contour regions querying schemes for WSNs [9–13]. We have selected part of them to analyze their core mechanisms, characteristics, advantages, and disadvantages.

First, let us introduce some top-k querying algorithms.

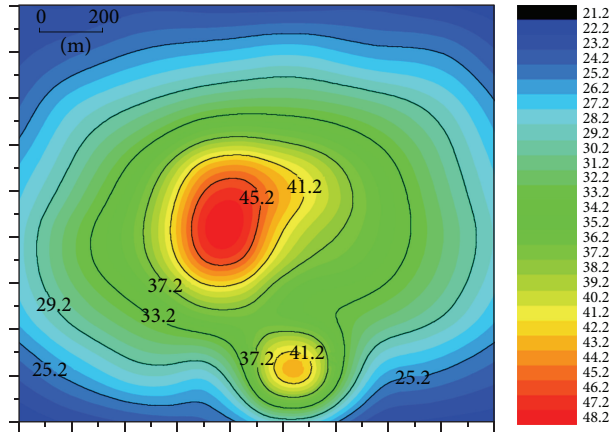


Figure 1: Contour regions of Kawah Ijen crater lake.

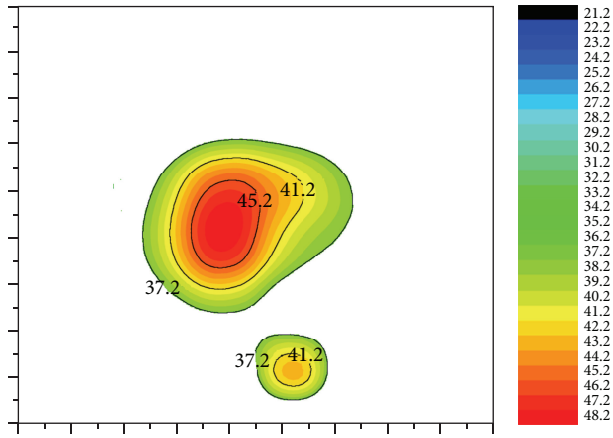


Figure 2: Top-k contour regions of Kawah Ijen crater lake.

In POT [3] protocol, they classify sensor nodes into a number of Partial Ordered Trees (POT), and the sink maintains the global ranking list (GR). POT protocol is useful for the occasion that top-k results are spatially correlated. When the top-k results are not spatially correlated, FILA [4] protocol outperforms POT.

In FILA [4] protocol, at every sampling point, if the new sensed data of a node does not change beyond the filtering window, the data will not be sent to the sink. If the sensed data comes into the filtering window of other nodes, the sink will broadcast to all nodes in the WSNs and acquire the needed data. In FILA, the transmission of data is discrete.

In PRIM [5] and PRIM-c [6] protocols, there are N partial TDMA frames in the TDMA schedule for collecting sensor readings. The sensed data are sent at different frames based on different values, and the higher sensor data can be sent to the sink in the more previous frames. The protocols are typical methods to save energy at the expense of time.

In XP [7] protocol, the authors construct a new routing structure in a bottom-up, spatially clustered fashion (called cluster tree) for the cross-pruning (XP) framework. Because the aggregation node will broadcast a filtering threshold to

its remaining children, it will consume more energy additionally.

Second, we will show some continuous contour regions querying protocols.

eScan [9] focuses on monitoring the remaining energy information of nodes in wireless sensor networks. The sensor nodes will report their remaining energy via a data collection tree. When the data is being sent back to the sink, intermediate nodes will aggregate the information as it flows. If the nodes are geographically adjacent and their readings are in the same value range, the aggregation may be done. Data is aggregated into polygons of similar value.

In isoline aggregation [10] protocol, each node needs to broadcast its reading to its neighbors. When a node receives the readings of all neighbors, the node will compare its reading with the readings of all neighbors. If the readings lie in different sides of an isoline, then a report needs to be generated. Reported isoline consists of the reading of one node and the readings of its neighbors whose readings come across the isoline.

In Iso-Map [11, 12] protocol, they proposed a parameter gradient direction based on the isoline aggregation [10] protocol. When all isoline nodes send their 3-tuple to the sink, the sink can construct the contour map based on the received 3-tuple. 3-tuple includes the isolevel of the node, position of the node, gradient direction of the node. The literature in [12] is an expanded version of the Iso-Map [11].

In CCM [13] protocol, each node maintains a CN-array structure, where 1 or 0 bit information of s_i's one-hop neighbors is saved, sequenced in counterclockwise cyclic order around the node s_i. If the reading of node s_i and the reading of its neighbor fall into the same level, s_i uses 0 to represent the reading of its neighbor in CN-array, otherwise s_i uses 1 to represent it. Each node updates its CN-array after receiving neighboring node broadcasts. Only a few contour nodes need to report their readings and CN-arrays to the sink and suppress their neighbors.

In the literature [14], a group of mobile data collecting nodes are deployed. The sensors are mounted on mobile objects so that they can be located in sample positions within target areas. Then the nodes emit signal vertically towards an upper reference plane. By detecting the returned wave, the receiver will work out the correct distance. Finally, an algorithm is applied on all the collected samples to plot the contour map. In this literature, many mobile nodes are deployed. While in our application scenario, the sensor nodes are stationary.

3. Preliminaries

In this paper, there are N sensor nodes constituting a network by self-organized manner. The sensor nodes sample the data periodically. Each sampling period is called a round. The sink node continuously requests the list of top-k contour regions with the highest (or lowest) contour level in every sampling period. The ith sensor node is denoted by s_i and the corresponding sensor nodes set $S = \{s_1, s_2, \ldots, s_n\}$, $|S| = N$.

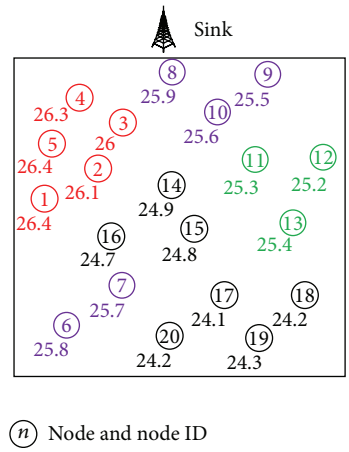


Figure 3: Temperature of each node in the network.

We make the following assumptions.

(1) All ordinary sensor nodes are homogeneous and have the same capabilities. The communication radiuses of all ordinary nodes are the same. When all nodes are deployed, they will be stationary, and each one has a unique identification (ID).

(2) There is only one sink (base station), and the sink node can be recharged.

(3) Links are symmetric. If node s_i can communicate with node s_j, node s_j can also communicate with node s_i.

(4) The energy resource of ordinary sensor nodes is highly limited and unreplenished.

4. The CTCRQ Scheme

The temporal data correlation [19, 20] means that the sensed readings are quite similar during a short period of time. We can use the temporal data correlation to reduce the number of data transmitted. If the data in current round is the same as the data in the last round, the data do not need to be sent to the sink repeatedly, which can save the energy consumption, as well as extend the network lifetime. If the data of a node is unlikely to belong to the top-k regions, the data will also be filtered.

4.1. Definitions. The idea of a normalized mechanism (which is the same as quantization of SENS-Join [21]) is to approximate a continuous range of values by a relatively small set of discrete values.

Definition 1. The whole range of an attribute value can be bounded using [**min_value**, **max_value**). Figure 3 displays the detailed temperature of each node in the network, and the whole range of the temperature is [0, 100).

Definition 2. A reading of a node belongs to a value subrange. A value subrange is denoted by [*LowerB, UpperB*), which means the lower bound and upper bound, respectively. Parameter **step** indicates the difference between *LowerB* and

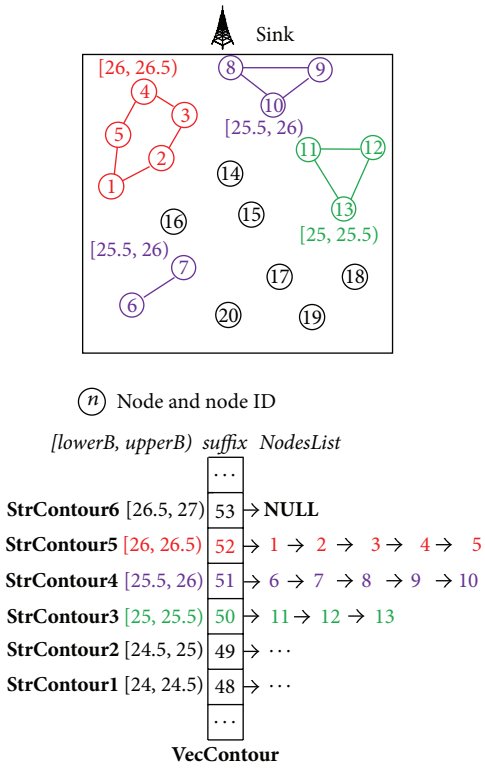


FIGURE 4: top-3 contour (polygon) regions.

UpperB. **step** = *UpperB* − *LowerB*. As shown in Figure 4, there is a temperature value sub-range [26.5, 27.0) and the **step** is 0.5.

Definition 3. We use a parameter *suffix* to express the normalized result value. *suffix* = *floor((value* −**min_value**)/**step**). The function *floor(A)* returns the nearest integer which is less than or equal to *A*. For example, $floor(5/4) = 1$. If the temperature is 26.2, the normalized result value *suffix* is $floor((26.2 - 0)/0.5) = 52$, which can be expressed by 1 byte. We have defined the whole range of an attribute value as **[min_value, max_value)**. The whole normalized range of an attribute value will be **[min_suffix, max_suffix)**. Thus the continuous real readings can be expressed by discrete integers.

Definition 4. The contour regions can be expressed by a structure **StrContour**, which is described as follows:

```
struct StrContour
{double LowerB; // lower bound.
 double UpperB; // upper bound.
 int suffix;
 list <int> NodesList; // node id list.
};
```

The parameter *NodesList* means the list of node id. If the number of nodes in the *NodesList* is 0, the *NodesList* will be denoted by **NULL**. If the nodes have the same *suffix*, they can be aggregated into the same *NodesList*. As shown in Figure 4, the *NodesList* of **StrContour6** has no nodes, which is denoted by **NULL**. The attribute value sub-range [*LowerB*, *UpperB*) of **StrContour5** is [26.0, 26.5), and the corresponding *suffix* is 52. The *suffix*es of nodes 1, 2, 3, 4, and 5 are the same, and these nodes are linked to the *NodesList* of **StrContour5**.

Definition 5. The total contour regions of the network can be expressed by a vector **VecContour**. Vector is implemented using dynamic array. The element of vector **VecContour** is the structure **StrContour**. As shown in Figure 4, the **VecContour** includes 6 valid **StrContour**, which are **StrContour1**, **StrContour2**,. . ., and **StrContour6**, respectively.

If the *NodesList* ofa **StrContour** is not **NULL**, the **StrContour** denotes a contour (polygon) region. The adjacent nodes in the same *NodesList* are interconnected to form one or more than one contour subregions. As shown in the top subfigure of Figure 4, we use three kinds of solid lines of different colors to represent the top-3 contour (polygon) regions. The corresponding **StrContours** are **StrContour5, StrContour4**, and **StrContour3**, whose value subranges are [26.0, 26.5), [25.5, 26.0), and [25.0, 25.5), respectively. The *NodesList* of **StrContour5** forms 1 contour subregion, which is denoted by red lines. The *NodesList* of **StrContour4** forms 2 contour subregions, which are denoted by purple lines. The *NodesList* of **StrContour3** forms 1 contour subregion, which is denoted by green lines.

4.2. The Detailed CTCRQ Scheme. Based on the above discussions and analyses, we proposed a Continuous Top-k Contour Regions Querying (CTCRQ) algorithm, which means continuously obtaining top-k **StrContours** with the highest (or lowest) *suffix* of the attribute value. In CTCRQ, if the reading of a node is not out of the value sub-range, the node will not report its reading to the sink in next round.

As shown in Figure 5, it is the flow chart of the sink. At the beginning of each round, the sink waits and receives the reported values from sensor nodes. Then the sink calculates the top-k results. If the number of the top-k results is greater than or equal to *k*, the sink broadcasts **M_NEXT_ROUND_BEGIN**(*ThresholdSuffix*) message and exits. Otherwise, the sink broadcasts a **M_PROBE**(*ThresholdSuffix*, *OldThresholdSuffix*) message to all nodes in the network. Then, the sink waits and receives the reported values from sensor nodes. After that, the sink calculates the top-k results and broadcasts **M_NEXT_ROUND_BEGIN**(*ThresholdSuffix*) message and exits.

The detailed algorithms are shown in Algorithm 1 (*Sink algorithm*) and Algorithm 2 (*Sensor node algorithm*).

As shown in Algorithm 1, in the initialize phase, the threshold suffix *ThresholdSuffix* is equal to **min_suffix**. When the sink receives a data message which includes one or more than one sensor node's data information, the sink finds the old location of each node in the *NodesList* of **StrContour** based on *suffix* and deletes it then links to the new location in *NodesList* of other **StrContour**. Then the sink sorts **StrContours** based on *suffix* in the **VecContour**.

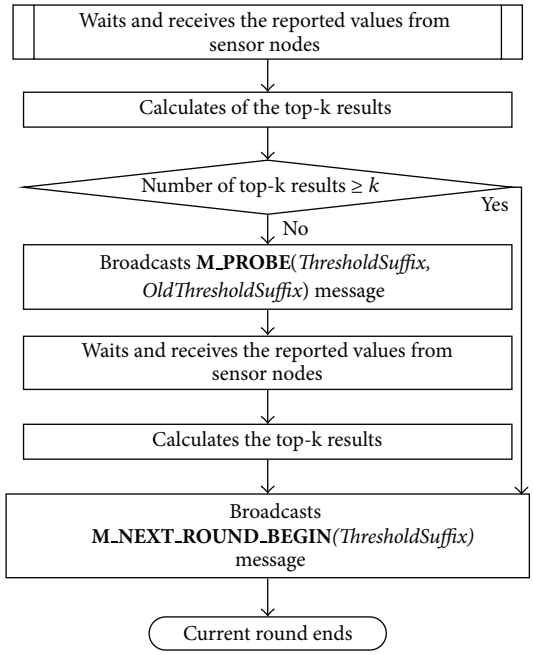


Figure 5: The flow chart of the sink.

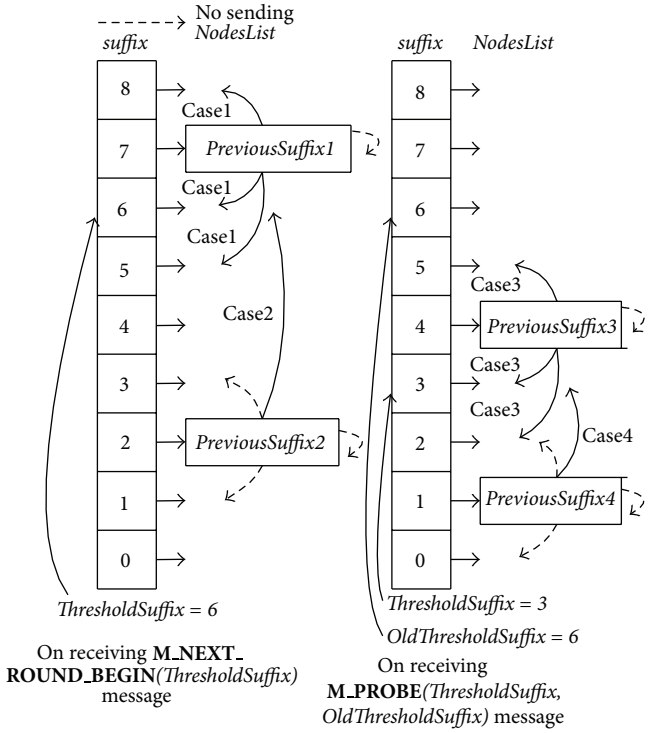


Figure 6: The cases of sending data to the sink.

From **max_suffix** to *ThresholdSuffix*, the sink calculates the top-k **StrContour**s (contour regions) whose *NodesList* is not **NULL** based on *suffix*. If the number of top-k results is greater than or equal to k, the sink sets the *ThresholdSuffix* to be the *suffix* of the kth **StrContour**s. The *suffix* of the kth **StrContour**s is denoted by *TopkSuffix*. After that the sink broadcasts an **M_NEXT_ROUND_BEGIN**(*ThresholdSuffix*) message to all nodes in the network to prepare for next round sampling. The threshold suffix *ThresholdSuffix* is contained in this message. The current round terminates. If the number of **StrContour**s is less than k, it will enter the probing phase. The sink sets *OldThresholdSuffix* to be *ThresholdSuffix*. From *ThresholdSuffix* to **min_suffix**, if the **StrContour** has one or more than one data updated, the sink adds this **StrContour** to the top-k result set (contour regions). If the number of result sets is greater than or equal to k, the sink sets the *ThresholdSuffix* to be *TopkSuffix*, otherwise sets the *ThresholdSuffix* to be **min_suffix**. The sink broadcasts an **M_PROBE**(*ThresholdSuffix*, *OldThresholdSuffix*) message to all nodes in the network and waits for a time period which guarantees the required data can be collected. When the time period expired, the sink calculates the top-k results and broadcasts **M_NEXT_ROUND_BEGIN**(*ThresholdSuffix*) message and exits.

Next, we will describe the sensor node algorithm which is shown in Algorithm 2. The parameter *PreviousSuffix* preserves the previous *suffix* of last round in which the sensor node sends data to the sink. In the initialize phase, *PreviousSuffix* is set to be −1.

When the sensor node receives an **M_NEXT_ROUND_BEGIN**(*ThresholdSuffix*) message, which means the beginning of the next round, the sensor node begins to sample the sensory attribute value and calculates *CurrentSuffix* based on the attribute value. The data (readings) of a sensor node will be sent to the sink in the following 2 cases.

Case 1. If *CurrentSuffix* is not equal to *PreviousSuffix*, and *PreviousSuffix* is greater than or equal to *ThresholdSuffix*, the sensor node sets *PreviousSuffix* to be *CurrentSuffix* and sends the data to the sink. As shown in Figure 6 on the left part of the figure, *ThresholdSuffix* is 6. The dotted line denotes no data sending. When *PreviousSuffix1* is 7 and *CurrentSuffix* is equal to 8, 6, or 5, the sensor node will send the data to the sink.

Case 2. If *CurrentSuffix* is not equal to *PreviousSuffix*, *PreviousSuffix* is less than *ThresholdSuffix* and *CurrentSuffix* is greater than or equal to *ThresholdSuffix*, the sensor node sets *PreviousSuffix* to be *CurrentSuffix* and sends the data to the sink. As shown in Figure 6 on the left part of the figure, *ThresholdSuffix* is 6. When *PreviousSuffix2* is 2 and *CurrentSuffix* is 6, the sensor node will send the data to the sink.

When the sensor node receives an **M_PROBE**(*ThresholdSuffix*, *OldThresholdSuffix*) message, the data (readings) of a sensor node will be sent to the sink in the following 2 cases.

Case 3. If *CurrentSuffix* is not equal to *PreviousSuffix*, *PreviousSuffix* is greater than or equal to *ThresholdSuffix* and less than *OldThresholdSuffix* and *CurrentSuffix* is less than *OldThresholdSuffix*, the sensor node sets *PreviousSuffix* to be *CurrentSuffix* and sends the data to the sink. As shown in Figure 6 on the right part of the figure, *ThresholdSuffix* is 3 and *OldThresholdSuffix* is 6. When *PreviousSuffix3* is 4 and

```
(1) initialize
(2)    ThresholdSuffix = min_suffix;
(3) end-initialize
Main program:
(4) Waits and receives the reported values from sensor nodes;
(5) for (each node which sends data to the sink)
(6)    Finds old location in NodesList of StrContour based on suffix and deletes it;
(7)    Links to the new location in NodesList of other StrContour;
(8) end-for
(9) Sorts StrContour based on suffixin the VecContour;
(10) for (max_suffix to ThresholdSuffix)
(11)    Calculates the top-k results;
(12) end-for
(13) if (number of top-k results) ⩾ k
(14)    ThresholdSuffix = TopkSuffix; // The suffix of the kth StrContour.
(15)    Broadcasts M_NEXT_ROUND_BEGIN(ThresholdSuffix) message and exits;
(16) else // need to probe
(17)    OldThresholdSuffix = ThresholdSuffix;
(18)    for (ThresholdSuffix to min_suffix)
(19)       if data updated then updates top-k results; end-if
(20)    end-for
(21)    if (number of top-k results) ⩾ k
(22)       ThresholdSuffix = TopkSuffix; // TopkSuffix has updated.
(23)    else ThresholdSuffix = min_suffix;
(24)    end-if
(25)    Broadcasts M_PROBE(ThresholdSuffix, OldThresholdSuffix) message;
(26)    Waits and receives the reported values from sensor nodes;
(27)    Repeats the steps of lines 5–15;
(28) end-if
```

Algorithm 1: Sink algorithm.

CurrentSuffix is 5, 3, or 2, the sensor node will send the data to the sink.

Case 4. If *CurrentSuffix* is not equal to *PreviousSuffix*, *PreviousSuffix* is less than *ThresholdSuffix* and *CurrentSuffix* is greater than or equal to *ThresholdSuffix* and less than *OldThresholdSuffix*, the sensor node sets *PreviousSuffix* to be *CurrentSuffix* and sends the data to the sink. As shown in Figure 6 on the right part of the figure, *ThresholdSuffix* is 3 and *OldThresholdSuffix* is 6. When *PreviousSuffix4* is 1 and *CurrentSuffix* is 3, the sensor node will send the data to the sink.

4.3. Theorem

Theorem 6. *The described CTCRQ algorithm correctly reports the top-k* **StrContour**s *(contour regions) result set ordered on their suffix.*

Proof. In each round, the final top-k result set will be obtained with probing phase or without probing phase.

If the final top-k result set is obtained without probing phase, each sensor node only receives the **M_NEXT_ROUND_BEGIN** message, as shown in Figure 6 on the left part of the figure. *FS* denotes the final top-k result set:

$$FS = \{s_i \mid s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\}_{\text{top-k}}. \quad (1)$$

When the sensor node receives an **M_NEXT_ROUND_BEGIN**(*ThresholdSuffix*) message, the data will be sent to the sink in Cases 1 and 2, as shown in Algorithm 2.

SS denotes the sending data set by sensor nodes.

Case 1

$$\begin{aligned} &SS\text{-}case\ 1 \\ &\quad = \{s_i \mid s_i \cdot PreviousSuffix \geqslant ThresholdSuffix\} \\ &\quad \supset \{s_i \mid s_i \cdot PreviousSuffix \geqslant ThresholdSuffix \\ &\qquad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\}. \end{aligned} \quad (2)$$

Case 2

$$\begin{aligned} &SS\text{-}case\ 2 \\ &\quad = \{s_i \mid s_i \cdot PreviousSuffix < ThresholdSuffix \\ &\qquad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\} \\ &(SS\text{-}case\ 1 \cup SS\text{-}case\ 2) \\ &\quad = (\{s_i \mid s_i \cdot PreviousSuffix \geqslant ThresholdSuffix\} \\ &\qquad \cup \{s_i \mid s_i \cdot PreviousSuffix < ThresholdSuffix \\ &\qquad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\}) \end{aligned} \quad (3)$$

```
(1) initialize
(2)    int PreviousSuffix = −1; // preserve the previous suffix.
(3) end-initialize
(4) On receiving M_NEXT_ROUND_BEGIN(ThresholdSuffix) message;
(5)   Begin sampling;
(6)   CurrentSuffix = floor((value − min_value)/step); // floor(A) returns the nearest integer which ⩽A.
(7)   if (CurrentSuffix ! = PreviousSuffix)
(8)     if PreviousSuffix ⩾ ThresholdSuffix // belongs to the top-k contour regions in last round. // Case 1
(9)       PreviousSuffix = CurrentSuffix;
(10)      Sends the data to the sink;
(11)    else // does not belongs to the top-k contour regions in last round.
(12)      if (CurrentSuffix >= ThresholdSuffix) // Case 2
(13)        PreviousSuffix = CurrentSuffix;
(14)        Sends the data to the sink;
(15)      end-if
(16)    end-if
(17)  end-if
(18) end processing M_NEXT_ROUND_BEGIN message;
(19) On receiving M_PROBE(ThresholdSuffix, OldThresholdSuffix) message;
(20)  if (CurrentSuffix ! = PreviousSuffix)
(21)    if ((PreviousSuffix >= ThresholdSuffix) && (PreviousSuffix < OldThresholdSuffix)
            && (CurrentSuffix < OldThresholdSuffix)) // Case 3
(22)      PreviousSuffix = CurrentSuffix;
(23)      Sends the data to the sink;
(24)    else
(25)      if ((PreviousSuffix < ThresholdSuffix) && (CurrentSuffix >= ThresholdSuffix)
              && (CurrentSuffix < OldThresholdSuffix)) // Case 4
(26)        PreviousSuffix = CurrentSuffix;
(27)        Sends the data to the sink;
(28)      end-if
(29)    end-if
(30)  end-if
(31) end processing M_PROBE message;
```

Algorithm 2: Sensor node algorithm.

$$\begin{aligned}
&\supset \left(\{s_i \mid s_i \cdot PreviousSuffix \geqslant ThresholdSuffix \right.\\
&\quad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\}\\
&\quad \cup \{s_i \mid s_i \cdot PreviousSuffix < ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\}\left.\right)\\
&= \{s_i \mid \ s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\}\\
&\supset \{s_i \mid s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\}_{\text{top-k}} = FS.
\end{aligned} \tag{4}$$

Hence, the final top-k result set FS is a part of the sending data set ($SS\text{-}case\ 1 \cup SS\text{-}case\ 2$).

If the final top-k result set is obtained with probing phase, each sensor node receives the **M_NEXT_ROUND_BEGIN** and **M_PROBE** messages, as shown in Figure 6 on the right part of the figure. FS denotes the final top-k result set:

$$FS = \{s_i \mid s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\}_{\text{top-k}}. \tag{5}$$

If the sensor node receives an **M_PROBE**(*ThresholdSuffix*, *OldThresholdSuffix*) message, the data will be sent to the sink in Cases 3 and 4, as shown in Algorithm 2.

Case 3

$SS\text{-}case\ 3$

$$\begin{aligned}
&= \{s_i \mid s_i \cdot PreviousSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot PreviousSuffix < OldThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}\\
&\supset \{s_i \mid s_i \cdot PreviousSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot PreviousSuffix < OldThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}.
\end{aligned} \tag{6}$$

Case 4

$SS\text{-}case\ 4$

$$\begin{aligned}
&= \{s_i \mid s_i \cdot PreviousSuffix < ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}
\end{aligned} \tag{7}$$

$$
\begin{aligned}
&(SS\text{-}case\ 3 \cup SS\text{-}case\ 4)\\
&\supset \big(\{s_i \mid s_i \cdot PreviousSuffix \geqslant ThresholdSuffix\\
&\quad \wedge s_i \cdot PreviousSuffix < OldThresholdSuffix\\
&\quad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\quad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}\\
&\quad \cup \{s_i \mid s_i \cdot PreviousSuffix < ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}\big)\\
&= \{s_i \mid s_i \cdot PreviousSuffix < OldThresholdSuffix\\
&\quad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\quad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}.
\end{aligned} \tag{8}
$$

The following is based on the formula (2).

$$
\begin{aligned}
&SS\text{-}case\ 1\\
&= \{s_i \mid s_i \cdot PreviousSuffix \geqslant OldThresholdSuffix\}\\
&\supset \big(\{s_i \mid s_i \cdot PreviousSuffix \geqslant OldThresholdSuffix\\
&\quad \wedge s_i \cdot CurrentSuffix \geqslant OldThresholdSuffix\}\\
&\quad \cup \{s_i \mid s_i \cdot PreviousSuffix \geqslant OldThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}\big)
\end{aligned} \tag{9}
$$

$$
\begin{aligned}
&(SS\text{-}case\ 1 \cup SS\text{-}case\ 3 \cup SS\text{-}case\ 4)\\
&\supset \big(\{s_i \mid s_i \cdot PreviousSuffix \geqslant OldThresholdSuffix\\
&\quad \wedge s_i \cdot CurrentSuffix \geqslant OldThresholdSuffix\}\\
&\quad \cup \{s_i \mid s_i \cdot PreviousSuffix \geqslant OldThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}\big)\\
&\quad \cup \{s_i \mid s_i \cdot PreviousSuffix < OldThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}\\
&= \big(\{s_i \mid s_i \cdot PreviousSuffix \geqslant OldThresholdSuffix\\
&\quad \wedge s_i \cdot CurrentSuffix \geqslant OldThresholdSuffix\}\\
&\quad \cup \{s_i \mid s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}\big).
\end{aligned} \tag{10}
$$

The following is based on the formula (3).

$$
\begin{aligned}
&SS\text{-}case\ 2\\
&= \{s_i \mid s_i \cdot PreviousSuffix < OldThresholdSuffix\\
&\quad \wedge s_i \cdot CurrentSuffix \geqslant OldThresholdSuffix\}
\end{aligned} \tag{11}
$$

$$
\begin{aligned}
&(SS\text{-}case\ 1 \cup SS\text{-}case\ 2 \cup SS\text{-}case\ 3 \cup SS\text{-}case\ 4)\\
&= (SS\text{-}case\ 1 \cup SS\text{-}case\ 3 \cup SS\text{-}case\ 4)\\
&\cup SS\text{-}case\ 2\\
&\supset \big(\{s_i \mid s_i \cdot PreviousSuffix \geqslant OldThresholdSuffix\\
&\quad \wedge s_i \cdot CurrentSuffix \geqslant OldThresholdSuffix\}\\
&\quad \cup \{s_i \mid s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}\big)\\
&\cup \{s_i \mid s_i \cdot PreviousSuffix < OldThresholdSuffix\\
&\quad \wedge s_i \cdot CurrentSuffix \geqslant OldThresholdSuffix\}\\
&= \big(\{s_i \mid s_i \cdot CurrentSuffix \geqslant OldThresholdSuffix\}\\
&\quad \cup \{s_i \mid s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\\
&\qquad \wedge s_i \cdot CurrentSuffix < OldThresholdSuffix\}\big)\\
&= \{s_i \mid s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\}\\
&\supset \{s_i \mid s_i \cdot CurrentSuffix \geqslant ThresholdSuffix\}_{\text{top-k}} = FS.
\end{aligned} \tag{12}
$$

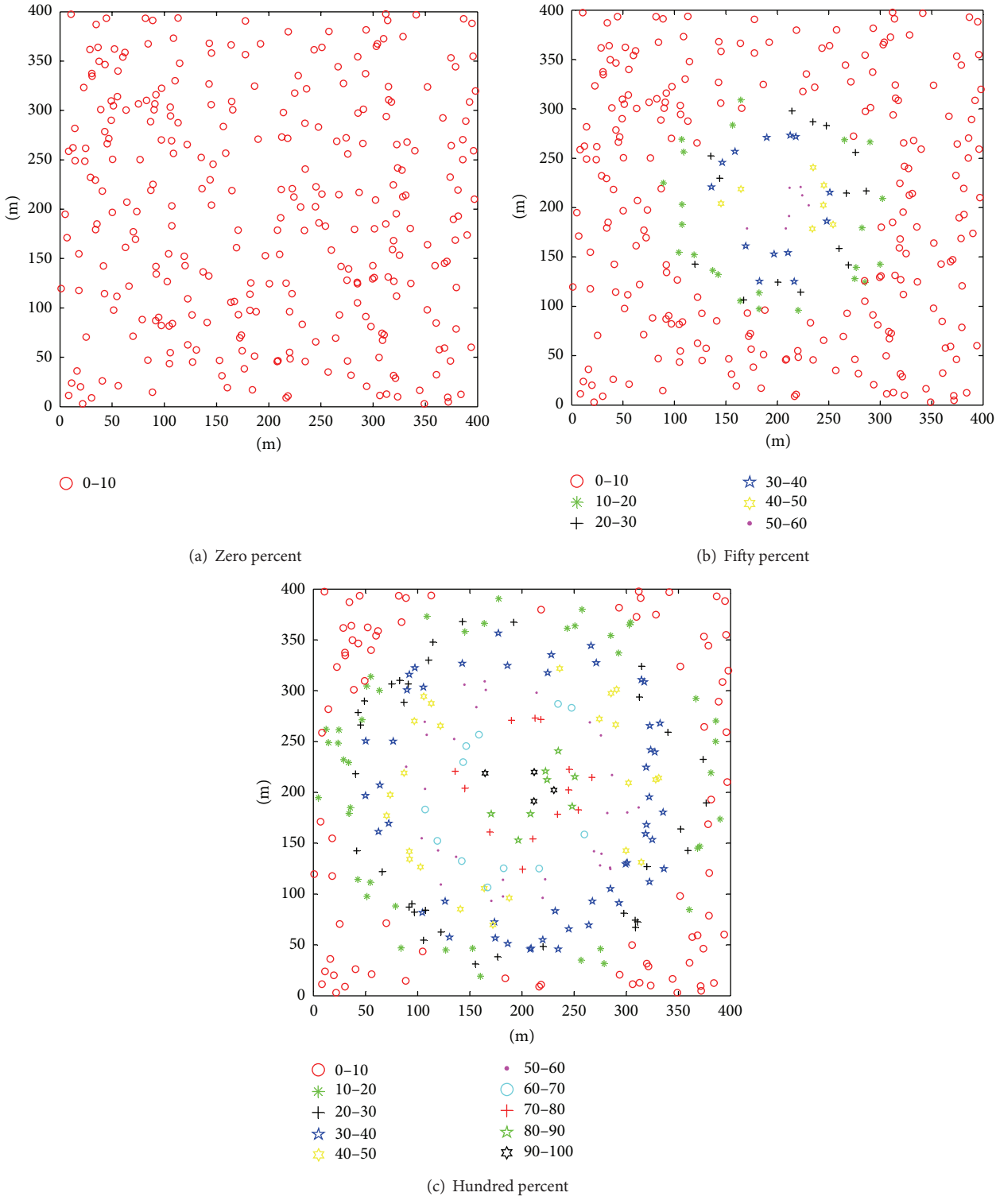
Hence, the final top-k result set FS is a part of the sending data set ($SS\text{-}case\ 1 \cup SS\text{-}case\ 2 \cup SS\text{-}case\ 3 \cup SS\text{-}case\ 4$).

Based on the above analysis, the sink will obtain the correct top-k result set. □

5. Simulation Results

To analyze the performance of our algorithm, we conduct experiments using omnetpp-4.1 [22].

We use a synthetic data set. We randomly deployed 300 homogeneous sensor nodes in the $400 * 400\,m^2$ rectangular region and the sink is located at the center. The data is generated using Gaussian distribution in which mean is proportional to the distance between a sensor node and the sink; the standard deviation is 0.5. As shown in Figure 7(a), the data values of all nodes are less than 10 in round 1. Starting from round 2, the data values of nodes closed to the sink increase gradually. As shown in the Figure 7(b), about round 25, the data values of the nodes which are closer to the sink are in the range [50–60), and the node is farther away from the sink; the data value is smaller. As shown in Figure 7(c), about round 50, the data values of the nodes which are closer to the sink are in the range [90–100). After that the data



(a) Zero percent

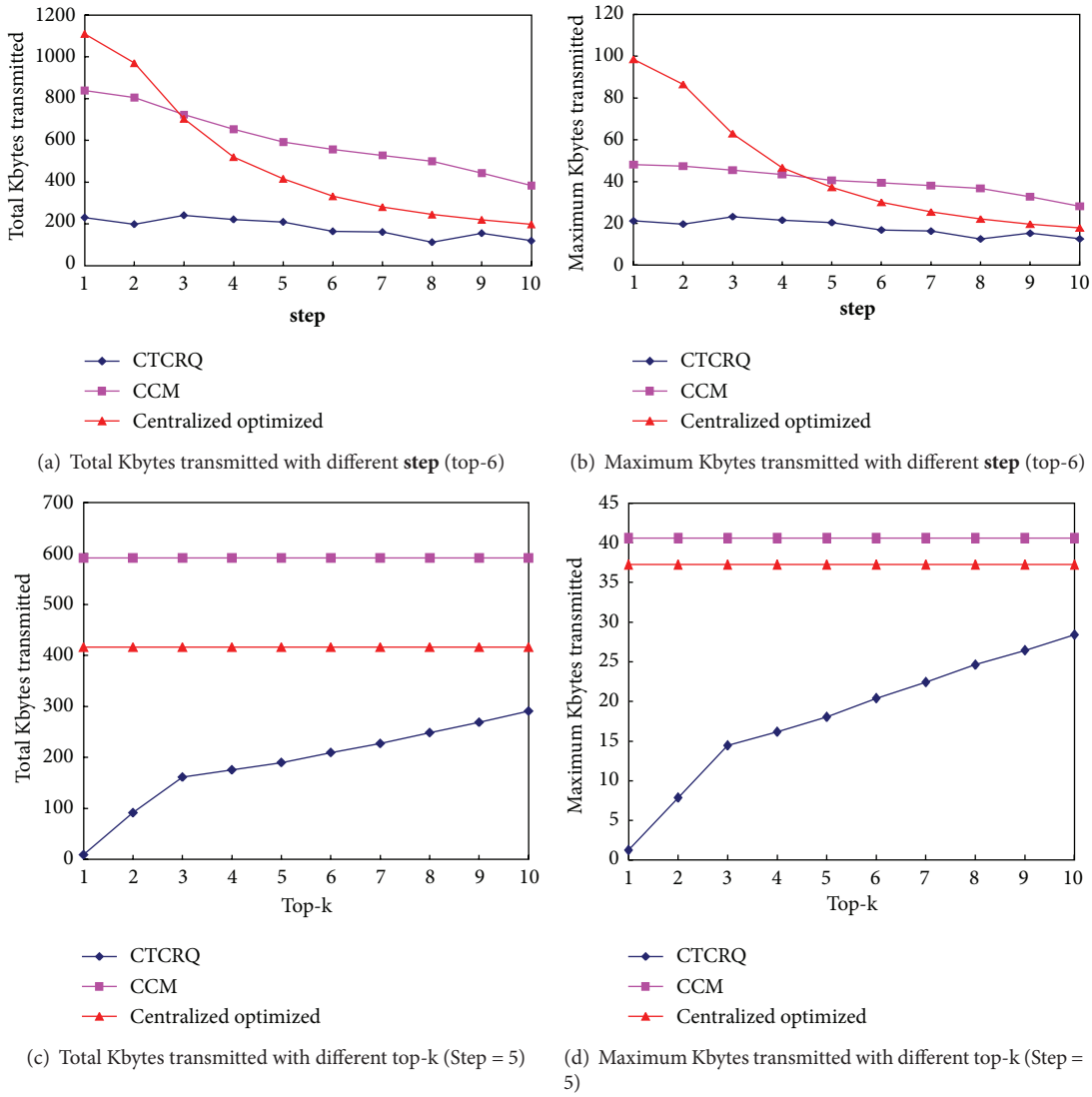
(b) Fifty percent

(c) Hundred percent

Figure 7: The data distribution of synthetic data set.

value of all nodes decreases gradually, it forms the subfigure (b) about round 75 and subfigure (a) about round 100. After that the data values of all nodes increase gradually, the cycle continues until it gets 1000 rounds data. We use this data set to simulate the process of expansion and shrinking gradually of the contour lines.

Our scheme is based on the TAG [23] routing algorithm and assumes communication links are error-free as well as



(a) Total Kbytes transmitted with different **step** (top-6)

(b) Maximum Kbytes transmitted with different **step** (top-6)

(c) Total Kbytes transmitted with different top-k (Step = 5)

(d) Maximum Kbytes transmitted with different top-k (Step = 5)

Figure 8: The performance comparison of synthetic data set.

MAC layer is ideal case. To compute the bytes transmitted, we define a sampling period as a round. Each of the following simulation result represents an average summary of 10 runs.

In our simulation, the size of node ID is 2 bytes. The attribute values are normalized and one normalized attribute value *suffix* occupies 1 byte. In CCM [13] algorithm, the parameter CN-array occupies 2 bytes.

The transmission range of the sink node is usually greater than the transmission range of ordinary sensor node, so we assume that the transmission range of the sink node can cover most regions of the monitoring networks.

5.1. Performance Analysis. As mentioned above, energy consumption is a critical issue for wireless sensor networks and radio transmission is the most dominate source of energy consumption. Thus, we measure the total and maximum amount of Kbytes (kilo bytes) data transmitted in wireless sensor networks as the performance metrics. Maximum amount of Kbytes data transmitted means the largest amount of data transmitted to the sink of a node among all nodes. We use this metric to evaluate the network lifetime, because one node which sends the largest amount of data to the sink means the node will consume the maximum energy and will die fastest. Therefore, compared with these two metrics, maximum amount of Kbytes data transmitted is more important than the total amount of Kbytes data transmitted.

As mentioned above, there are many continuous contour querying algorithms, such as eScan [9], isoline aggregation [10], Iso-Map [11, 12], CCM [13], the literature in [14], and improved Isoline aggregation [15]. In these algorithms, the literature in [12] is an expanded version of the Iso-Map [11]. In the literature in [14], many mobile nodes are deployed. While in our application scenario, the sensor nodes are stationary. So the algorithm proposed in the literature in [14] is not suitable as a comparison algorithm. The improved Isoline

aggregation [15] algorithm has no essential breakthrough compared with the isoline aggregation [10] algorithm, while the CCM [13] algorithm is a representative and energy efficient algorithm in these algorithms. Thus, we compared our CTCRQ algorithm with CCM algorithm.

For fair comparison, we modified the CCM algorithm a little. If CN-array and value range are not changed then node s_i does not broadcast a "report sent message" to its one-hop neighbors and does not send its ID back to the sink. If the sink does not receive a "report sent message" from node s_i, it denotes that the value of node s_i is not changed. In this way, the network can further reduce the amount of data transmitted than the original CCM algorithm.

To obtain the top-k contour region, the naive approach, that we called Centralized Algorithm, is that all nodes transmit their data to the sink at every round. The total data transmitted is 5031.5 Kbytes. The maximum data transmitted is 337.125 Kbytes. The total and maximum data transmitted of Centralized algorithm are obviously larger than the ones of other three algorithms (Centralized Optimized, CTCRQ, and CCM algorithms).

Centralized Optimized Algorithm uses temporal data correlation to reduce the amount of data transmitted. In this optimized version, nodes report their data (readings) directly to the sink only when their values have changed from one value sub-range to another.

We first investigate the impact of changing the value sub-range of **StrContour** on the network performance. Parameter **step** indicates the difference of sub-range, as shown in Definition 2. As shown in subfigures (a) and (b) of Figure 8, with the **step** increasing, the total and maximum Kbytes transmitted of all three algorithms decrease. The greater the **step** is, the greater the scope of the value sub-range is, as well as the smaller the possibility of the data need to be sent is. With the **step** increases, the total and maximum Kbytes transmitted of CTCRQ algorithm decrease too, and the trend is gentle. Because they only calculate the bytes transmitted of top-6 region nodes, and the number of top-6 region nodes is less than the nodes of total network.

When the **step** size is relatively smaller, the number of nodes which have changed their values from one sub-range to another sub-range is more, and the number of nodes that need to be reported is more too. In CCM algorithm, using the CN-array technology, a reporting node can suppress all contour nodes around it. Hence, when the **step** size is relatively smaller, the CCM algorithm is superior to the Centralized Optimized algorithm.

With the **step** size increasing, the number of nodes which have changed their values from one sub-range to another sub-range decreases and the number of nodes that need to be reported decreases too. In CCM algorithm, if a node changed its value from one sub-range to another sub-range, the node will broadcast its node ID and value information to its neighbor nodes. Its neighbor nodes may not change their values. Then the node will send "report sent message" which only includes the information of itself to the sink. While in the Centralized Optimized algorithm, nodes send their data (readings) to the sink only when their values have changed from one sub-range to another. Hence, when the **step** size is relatively larger, CCM algorithm is not as good as the Centralized Optimized algorithm.

The total and maximum Kbytes transmitted of CTCRQ algorithm are less than the ones of Centralized Optimized and CCM algorithms.

As shown in subfigures (c) and (d) of Figure 8, with the number of top-k increasing, the total and maximum Kbytes transmitted of CTCRQ algorithm increase too. Both in calculating the total number of bytes transmitted and the maximum amount of bytes transmitted, CTCRQ algorithm is better than Centralized Optimized and CCM algorithms.

Experimental result shows that CTCRQ algorithm outperforms the existing ones in term of data transmission.

6. Conclusions and Future Work

In this paper, we proposed a Continuous Top-k Contour Regions Querying (CTCRQ) algorithm in wireless sensor networks. Our experimental result shows that the proposed CTCRQ scheme can reduce the amount of bytes transmitted as well as extend the network lifetime.

In the future, we plan to extend the proposed scheme to other aggregate functions such as join, average, and sum.

Acknowledgments

This research was supported by the National Basic Research Program of China (973 program) (2012CB316205) and the National Natural Science Foundation of China (61070056, 61033010).

References

[1] Institute of Electrical Electronics Engineers, "Ten emerging technologies that will change your world," *IEEE Engineering Management Review*, vol. 32, pp. 20–30, 2004.

[2] H. Zhang, Z. Wu, D. Li, and H. Chen, "A sampling-based algorithm for approximating maximum average value region in wireless sensor network," in *Proceedings of the 39th International Conference on Parallel Processing Workshops (ICPPW '10)*, pp. 17–23, San Diego, CA, USA, September 2010.

[3] Y. Cho, J. Son, and Y. D. Chung, "POT: an efficient top-k monitoring method for spatially correlated sensor readings," in *Proceedings of the 5th International Workshop on Data Management for Sensor Networks (DMSN '08)*, pp. 8–13, August 2008.

[4] M. Wu, J. Xu, X. Tang, and W. C. Lee, "Top-k monitoring in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 7, pp. 962–976, 2007.

[5] M. H. Yeo, D. O. Seong, and J. S. Yoo, "PRIM: priority-based top-k monitoring in wireless sensor networks," in *Proceedings of the International Symposium on Computer Science and its Applications (CSA '08)*, pp. 326–331, Hobart, Australia, October 2008.

[6] M. Yeo, D. Seong, and J. Yoo, "Data-aware top-k monitoring in wireless sensor networks," in *Proceedings of the IEEE Radio and Wireless Symposium (RWS '09)*, pp. 103–106, San Diego, CA, USA, January 2009.

[7] X. Liu, J. Xu, and W. C. Lee, "A cross pruning framework for Top-k data collection in wireless sensor networks," in

Proceedings of the 11th IEEE International Conference on Mobile Data Management (MDM '10), pp. 157–166, Kansas City, MO, USA, May 2010.

[8] S. Mo, H. Chen, and Y. Li, "Clustering-based routing for top-k querying in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, article 73, 2011.

[9] Y. J. Zhao, R. Govindan, and D. Estrin, "Residual energy scans for monitoring wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '02)*, pp. 17–21, 2002.

[10] I. Solls and K. Obraczka, "Efficient continuous mapping in sensor networks using isolines," in *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems-Networking and Services (MobiQuitous '05)*, pp. 325–332, July 2005.

[11] Y. Liu and M. Li, "Iso-map: energy-efficient contour mapping in wireless sensor networks," in *Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS '07)*, Toronto, Canada, June 2007.

[12] M. Li and Y. Liu, "Iso-map: energy-efficient contour mapping in wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 5, pp. 699–710, 2010.

[13] C. Zhong and M. Worboys, "Continuous contour mapping in sensor networks," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, pp. 152–156, Las Vegas, NV, USA, January 2008.

[14] Q. Chen, "Automatic contour mapping system in sensor network," *Applied Mechanics and Materials*, vol. 182-183, pp. 1164–1168, 2012.

[15] R. Guocan and D. Guowei, "An improved isoline based data aggregation scheme in wireless sensor networks," *Procedia Engineering*, vol. 23, pp. 326–332, 2011.

[16] J.-J. Kim, I.-S. Shin, Y.-S. Zhang, D.-O. Kim, and K.-J. Han, "Aggregate queries in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 625798, 15 pages, 2012.

[17] R. Zhu, "Efficient fault-tolerant event query algorithm in distributed wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2010, Article ID 593849, 7 pages, 2010.

[18] R. Zhu, "Intelligent collaborative event query algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 728521, 11 pages, 2012.

[19] I. F. Akyildiz, M. C. Vuran, and Ä. B. Akan, "On exploiting spatial and temporal correlation in wireless sensor networks," in *Proceedings of the Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '04)*, pp. 71–80, 2004.

[20] N. D. Pham, T. D. Le, and H. Choo, "Enhance exploring temporal correlation for data collection in WSNs," in *Proceedings of the IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies*, pp. 204–208, Ho Chi Minh City, Vietnam, July 2008.

[21] M. Stern, E. Buchmann, and K. Böhm, "Towards efficient processing of general-purpose joins in sensor networks," in *Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE '09)*, pp. 126–137, Shanghai, China, April 2009.

[22] http://www.omnetpp.org/.

[23] S. Madden, M. J. Franklin, J. Hellerstein, and W. Hong, "TAG: a tiny aggregation service for ad-hoc sensor networks," in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, pp. 131–146, December 2002.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 472675, 10 pages
http://dx.doi.org/10.1155/2013/472675

Research Article

Concurrent Fault Diagnosis for Rotating Machinery Based on Vibration Sensors

Qing-Hua Zhang,[1] Qin Hu,[1,2] Guoxi Sun,[1] Xiaosheng Si,[3] and Aisong Qin[1]

[1] *Guangdong Petrochemical Equipment Fault Diagnosis Key Laboratory, Guangdong University of Petrochemical Technology, Maoming 525000, China*
[2] *School of Automation, Guangdong University of Technology, Guangzhou 510006, China*
[3] *Department of Automation, Tsinghua University, Beijing 100084, China*

Correspondence should be addressed to Guoxi Sun; 158011382@qq.com

Received 10 January 2013; Accepted 5 April 2013

Academic Editor: Zhangbing Zhou

Rotating machinery is widely used in modern industry. It is one of the most critical components in a variety of machinery and equipment. Along with the continuous development of science and technology, the structures of rotating machinery become of larger scale, of higher speed, and more complicated, which results in higher probability of concurrent failure in practice. It is important to enable reliable, safe, and efficient operation of large-scale and critical rotating machinery, which requires us to achieve accurate diagnosis of concurrent fault, for example, rolling bearing diagnosis, gearbox diagnosis, and compressor diagnosis. In this paper, to achieve concurrent fault diagnosis for rotating machinery, which cannot be accurately diagnosed by existing methods, we develop an integrated method using artificial immune algorithm and evidential theory.

1. Introduction

For a complex engineering system [1], many fault diagnosis problems involve quantitative data and qualitative information, as well as various types of uncertainties, for example, incompleteness and fuzziness [2]. Conventional analytical models based on pure data, for example, time series analysis models and filter based models, are not always applicable, since it is difficult to obtain a complete set of historical data for developing a perfect mathematical model to simulate a system [3, 4]. Furthermore, due to the fact that human beings hold ultimate responsibility in most situations, the subjective judgment plays an irreplaceable role in making final decision, which may not be always accurate. Thus, it is highly desirable to develop a fault diagnosis that can model and analyze diagnosis problems using uncertain information, which is likely to be incomplete and vague.

The development of methods for handling uncertain information has received considerable attention in the last three decades. Several numerical and symbolic methods have been proposed. Three of the most common frameworks for representing and reasoning with uncertain knowledge are (i) Bayesian probability theory; (ii) Demister-Shafer (D-S) theory of evidence; (iii) Fuzzy set theory. Due to the power of the D-S theory in handling uncertainties, so far, it has found many application areas, for example, expert system, uncertainty reasoning.

To avoid the drawbacks of using evidential theory alone to fault diagnosis, for example, its heavy calculation burden due to the exponential "explosion" of the focal elements involved by fault information and data combination [5], in this paper, we present an integrated approach for the concurrent fault diagnosis using artificial immune algorithm [6] and evidential theory [7], aiming to not only improve the diagnosis rate but also increase the reliability of diagnostic conclusion.

This research work has the following theoretical and practical contributions.

(i) Our method can represent the uncertainty existing in the result of fault diagnosis due to the evidence theory involved.

(ii) To demonstrate our method, we apply our method to the real test bed of concurrent fault diagnosis for rotating machinery.

(iii) The results show that our developed method can make accurate judgment for concurrent fault of the rotating machinery which has certain credibility.

(iv) Regular maintenance of equipment can be scheduled effectively.

The remaining part of this paper is organized as follows. In Section 2, the related work is surveyed and presented. In Section 3, we will describe the experimental conditions and the detailed parameters of the used vibration sensors. Section 4 describes the artificial immune algorithm and dimensionless parameters for fault diagnosis. Our proposed integrated concurrent fault diagnosis method base on the artificial immune algorithm and evidence theory will be given in Section 5. The systematical evaluation in a real test bed for this proposed method is given in Section 6, and Section 7 concludes this paper.

2. Related Work

Previous researchers have conducted considerable effort on rotating machinery diagnostics and developed a variety of diagnosis methods. In [11], a fault diagnosis system based on the wavelet transform and artificial immune system is presented, in which the wavelet transform is used to analyze nonstable signals and obtain their eigenvectors, and the artificial immune algorithm is also proposed to conduct self-nonself analysis based on these eigenvectors. The system is successfully applied to vehicle fault diagnosis with good results. In [12], a new hybrid approach based on conventional fuzzy soft clustering and artificial immune systems for sensor multiple faults is proposed, which can require no prior knowledge, or the system behavior, and no learning processes are required. This new approach uses the fuzzy clustering c-means algorithm firstly to generate a single fuser for the input sensor signals. Then a fault detector was generated based on the artificial immune systems. In [13], support vector machine (SVM) is a classification method, but some parameters in SVM are selected by man's experience. Aydin et al. used a multiobjective artificial immune algorithm (AIA) to optimize the parameters of SVM. The fault diagnosis of induction motors and anomaly detection problems shows that the SVM optimised by AIA can give higher recognition accuracy than the normal SVM. In [14], Wang et al. proposed the improved immune algorithm based on Discrete Particle Swarm Optimization (DPSO) technique to solve the problem that exists in fault diagnosis. This approach can improve the mutation mechanism and enhance the immune algorithms performance. Simulation results show that the new scheduling algorithm can deal with the uncertainty situation and be suitable for multifaults diagnosis. In [15], a composite fault diagnosis approach was proposed, combining the real-valued negative selection (RNS) algorithm with the support vector machine. In the new method, the difficult problem of lacking training samples was solved by using the new method in the conventional classification algorithm. In [16], Wanjun et al. proposed a fuzzy-immunity mixed fault diagnosis method. This method can resolve quantitatively diagnosis for gun-launched missile fault in the lack of prior knowledge. In [17], the authors propose two model-based fault detection and isolation schemes for robot manipulators using soft computing techniques, as an integrator of Neural Network and Fuzzy Logic. The first scheme isolates faults by passing residual signals through a neural network. The second scheme isolates faults by modelling faulty robot models for defined faults and combining these models as a generalized observers scheme structure. In [18], the authors present an integrated fault diagnostics model based on the Genetic Algorithm and Artificial Neural Network for identifying shifts in component performance and sensor faults. The diagnostics model uses response surfaces for computing objective functions to increase the exploration potential of the search space while easing the computational burden firstly. Then a nested neural network is used with genetic algorithm. The nested neural network functions as a filter to reduce the number of fault classes to be explored by the genetic algorithm. In [19], the authors present an integrated neural fuzzy approach for transformer fault diagnosis, which formulates the modeling problem of higher dimensions into lower dimensions. Then, the fuzzy rule base is designed by applying the subtractive clustering method which is very good at handling the noisy input data. The simulation result shows that the method possesses superior performance in identifying the transformer fault type. In [20], the authors make use of advantages of the neural network and the fault tree to construct the fault diagnosis system. The fault tree with an intuitive, logical, and strong features for a simple structure determines the source of the problem, but for accessing to complex and uncertain domain knowledge, and neural networks technology has to learn the characteristics of self-association and can just make up the shortcomings of the model fault tree diagnosis. The authors take control box faults of a certain digital control system as an example to simulate. The training results show that the method can be more accurate and quick to make fault diagnosis. In [21], the author proposed a new method based on support vector machine with genetic algorithm to fault diagnosis of a power transformer. In this method, the genetic algorithm is used to select appropriate free parameters of support vector machine. The simulation result indicates that the proposed method can achieve higher diagnostic accuracy. In [22], the authors proposed a fault diagnosis method based on kernel principal component analysis (KPCA) and support vector machines (SVM). Firstly initial feature vectors of motor vibration signal were mapped into higher-dimensional space with kernel function. Then the PCA method was used to analyze the data in the high-dimensional space to extract the nonlinear features which is used as training sample of SVM fault classifier. Lastly, the rotor fault is identified using the trained classifier. Research results showed that the method can identify motor rotor fault efficiently and fulfill fault classification accurately. In [23], the researcher demonstrated the ability of genetic programming to discover automatically the different bearing conditions. The genetic programming

(GP) can generate new features from the original dataset without prior knowledge of the probabilistic distribution. The created features are then used as the inputs to a neural classifier for the identification of six bearing conditions. In [24], the authors presented a fault diagnostic method based on a real-encoded hybrid genetic algorithm evolving a wavelet neural network (WNN). The main drawbacks of a back propagation algorithm of wavelet neural network (WNN) are that the optimal procedure is easily stacked into the local minima and cases that strictly demand initial value. A real-encoded hybrid genetic algorithm evolving a WNN can be used to optimize the structure and the parameters of WNN instead of humans in the same training process. A number of examples were further given to show that the method proposed has good classifying capability for single- and multiple-fault samples of power transformers as well as high fault diagnostic accuracy. In [25], a fault diagnosis method is proposed based on adaptive neurofuzzy inference system (ANFIS) in combination with classification and regression tree (CART), which is used as a feature selection procedure to select pertinent features from data set. The crisp rules obtained from the decision tree are then converted to fuzzy if-then rules. The hybrid of back-propagation and least squares algorithm are utilized to tune the parameters of the membership functions. Research results show that the CART-ANFIS model has potential for fault diagnosis of induction motors. In [26], the authors presented a study on the application of particle swarm optimization (PSO) combined with artificial neural networks (ANNs) and support vector machines (SVMs) for bearing fault detection in machines. The classifier parameters, for example, the number of nodes in the hidden layer for ANNs and the kernel parameters for SVMs, are selected along with input features using PSO algorithms. In [27], Dong et al. applied neural network and Dempster-Shafer theory (D-S) to rotor in turbine generator set to diagnose multiple faults. The D-S reasoning theory is used to do fusion decision making based on the diagnosis result.

From the above survey over the related works, we can observe that there is a continuing tread to develop a fusion method (which can fully utilize the advantages of multiple methods) for the concurrent fault diagnosis, due in part to the complexity of concurrent failures and uncertainty in its diagnosed results. This naturally leads to our primary objective of this paper, that is, to achieve concurrent fault diagnosis for rotating machinery, which cannot be accurately diagnosed by the current methods.

3. Experimental Conditions and Basis

3.1. The Studied Rotating Machinery. In this research work, a motor-gearbox-magnetic powder brake test device has been developed by our research lab (Guangdong province Petrochemical Equipment Fault Diagnosis Key Laboratory, Guangdong University of Petrochemical Technology, China), as shown in Figure 1, which consists of three major parts as (1) motor, (2) magnetic powder brake, and (3) transducer. Two sensors are used to gather sensory data from the rotating machinery. The detailed parameters for each unit are given in Table 1.

Table 1: The main components of the studied rotating machinery.

Equipment	Motor	Magnetic powder brake	Transducer
Model	Senlima YP	CZ-2	Anchuan-VS606V7
Parameters	Power 1.5 Kw; rated voltage 380 V; reference frequency 50 Hz	Rated torque 20 N·m; exciting current 0.6 A; allowable sliding power 1.6 Kw	Three phase; rated voltage 400 V; power 1.5 Kw

Table 2: The characteristics of the vibration sensor.

Measurement range	Collection functions
Acceleration: 0.1~199.9 m/s^2 Single peak value (0-P)	The range of frequencies: the minimum frequency:10 Hz; the maximum frequency: 100~10 KHz with adjusting continuously
Velocity: 0.01~19.99 cm/s	Dynamic range: 72 dB
Virtual value (rms) Displacement: 0.001~1.999 mm	The stop band of decay rate: 72 dB/oct
Peak-peak values (P-P)	the acquisition parameters: vibration acceleration, vibration velocity, and vibration displacement
The minimum frequency: 10 Hz	Antialiasing low-pass filter: 100~10 KHz with continuous variable
The maximum frequency: acceleration 10 Hz, velocity 1 Hz, and displacement 1 KHz	
The measure precision: plus-minus 5% plus-minus two words	

3.2. The Used Sensory Data Collection System. In Figure 2, we can see the used vibration sensor and machinery health collector in this research work. The sensor can gather three types of information: (1) vibration acceleration, (2) vibration velocity, and (3) vibration displacement. The detailed feature description about the used sensor is given in Table 2.

3.3. Used Variable in This Paper. The used variable in this paper is listed in Table 3.

4. Artificial Immune Algorithm and Dimensionless Parameters

Based on biological immune system through antigen recognition, immune response, and clone selection process to judge their "own" or "nonself" substances (Figure 3 shows the concept of self and nonself space), to maintain and protect a

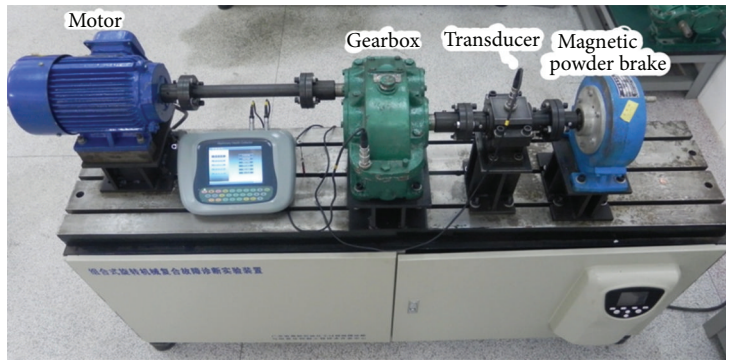


Figure 1: The developed real test bed.

Table 3: The used variable in this paper.

Variable	Definition
$p(x)$	The probability density function from the observed vibration signal's amplitude
x	The vibration amplitude
S_f	Waveform index
I_f	Impulse index
CL_f	Margin index
C_f	Peak index
K_v	Kurtosis index
X_{rms}	A dimension parameter named as root of mean square(RMS)
m_{ij}	The sensitivity of the ith kind of nondimensional index detector to the jth kind of failure
g_{ij}	The effective information preserved in the total feature information of a dimensionless index to generate a mature immune detector
d_{ij}	The diagnosis capability factor of the kind of ith dimensionless index immune detector to the kind of jth fault

stable internal environment, Forrest and Hofmeyr proposed a negative selection algorithm in [28] to detect patterns.

4.1. Negative Selection Algorithm. The basic idea of the negative selection algorithm is to produce a set of change detectors, which can detect changes in what is considered normal behavior of a system. The algorithm consists of two stages: Generating Detector Set stage and Training Detector Set stage. The Generating Detector Set stage caters for the generation of change detectors. Briefly, the procedure proposed the following: (1) define self data; (2) generate a candidate detector randomly; and (3) match each candidate detector with self data; if it matches; delete it, if not, add it to detector set. Subsequently, system is monitored for changes using the detectors generated in Generating Detector Set stage; if any detector ever matches the input data, then an abnormal is known to have occurred. Otherwise, the system is normal, as shown in Figure 4.

Figure 2: In this machinery health collector, two vibration sensors are associated. The gathered sensory data will be stored in the data collector first and then exported to computer.

In this paper, we first introduce this algorithm to detect the type of a fault in rotating machinery by constructing an immune detector based on the vibration signal analysis.

4.2. Dimensionless Parameters. Based on the probability density function $p(x)$ from the observed vibration signal, we obtained the dimension parameters such as the mean, the average amplitude, the root of mean square (RMS) value, slope, and kurtosis. These parameters have the sensitivity of equipment failure, but if we directly use them for fault detection, because of changing working conditions, such as speed, load, and equipment sensitivity, these dimension parameters also change over time. This means that these parameters are difficult to improve the diagnostic accuracy when used directly. To solve this problem, we introduce dimensionless parameters in the following [29].

Dimensionless parameters are the ratio of two-dimension parameters, defined as follows:

$$\zeta_x = \frac{\left[\int_{-\infty}^{+\infty} |x|^l p(x)\, dx\right]^{1/l}}{\left[\int_{-\infty}^{+\infty} |x|^m p(x)\, dx\right]^{1/m}}, \quad (1)$$

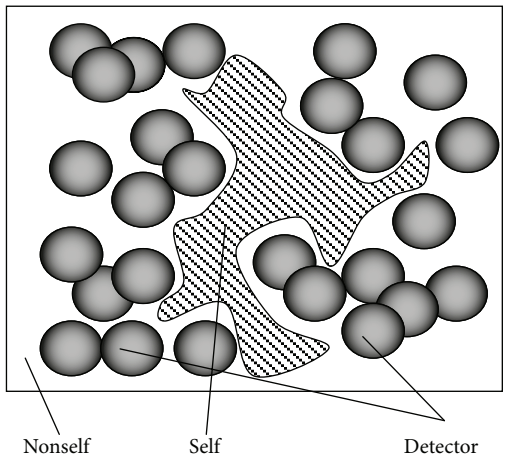


Figure 3: The figure illustrates the concept of self, nonself, and detector in a feature space.

where x denotes the vibration amplitude, and $p(x)$ denotes the probability density function of vibration amplitude.

In the practice, the following dimensionless parameters are frequently used and it is found that they behave well for rotating machinery fault diagnosis, specifically,

(i) if $l = 2, m = 1$, have waveform index S_f,

(ii) if $l \to \infty, m = 1$, have impulse index I_f,

(iii) if $l \to \infty, m = 1/2$, have margin index CL_f,

(iv) if $l \to \infty, m = 2$, have peak index C_f,

(v) kurtosis index $K_v = \beta / X_{\text{rms}}^4$. In the formula, β is a dimension parameter named kurtosis, which is defined as $\beta = \int_{-\infty}^{\infty} x^4 p(x) dx$. X_{rms} is a dimension parameter named as RMS, defined as $X_{\text{rms}} = \sqrt{\int_{-\infty}^{\infty} x^2 p(x) dx}$.

Based on $p(x)$ and the vibration signals, the five nondimensional parameters defined above can be directly used to detect the anomaly in the vibration signal in real time and implemented easily. It is also found that these parameters are sensitive enough to the fault and thus can reflect the fault condition and are not interfered with the absolute level of vibration signals. In addition, the dimensionless parameters are stable with respect to the load, conditions, and speed of equipment so that the relationship between the working conditions of the machine and these parameters is not significant. This unique feature is very useful for fault diagnosis since they can capture the anomaly only in the observed signals.

In order to facilitate us to introduce the integration diagnosis method, some definitions used in this paper are as follows.

Definition 1. we define m_{ij} as the fault sensitive index of the ith kind of nondimensional index detector to the jth kind of failure, with $m_{ij} \le 1, i = 1, 2, \ldots, l, j = 1, 2, \ldots, n$. The ratio m_{ij} measures the values for the ratio of the maximum nondimensional failure index value and the minimum size of the index under normal conditions. The smaller the ratio is, the less sensitive the index is to the fault.

Definition 2. we define g_{ij} as the effective information preserved in the total feature information of a dimensionless index to generate a mature immune detector, with $g_{ij} \le 1$, $i = 1, 2, \ldots, l$, $j = 1, 2, \ldots, n$. The quantity g_{ij} measures the ratio of the amount of the lost value and the useful fault information. The more the lost value is, the smaller the quantity g_{ij} is.

Definition 3. we defined d_{ij} as the diagnosis capability factor of the kind of ith dimensionless index immune detector to the kind of jth fault, with $d_{ij} \le 1, i = 1, 2, \ldots, l, j = 1, 2, \ldots, n$. The quantity d_{ij} can be calculated as follows,

$$d_{ij} = \frac{m_{ij} g_{ij}}{\sum_{k=1}^{l} m_{kj} g_{kj}}, \quad i = 1, 2, \ldots, l, \ j = 1, 2, \ldots, n. \tag{2}$$

5. The Integration of Artificial Immune and Evidential Theory

The method developed by Zhang [30] improved the negative selection algorithm and made use of the above five nondimensional parameters to generate the five immunity detectors. These detectors were further used to diagnose the fault of rotating machinery. We can obtain the range of each dimensionless parameter in different fault mode. It is found in the experimental results that the ranges of some nondimensional parameters for different values of fault modes will duplicate or even cross with each other. Making the target dimensionless parameter immune detector with each fault mode is unique and does not cross with those of other dimensionless characteristics. In practice, it is usual to reduce these value ranges to achieve an accurate fault diagnosis. The detector with such desired properties is named mature immune detector.

5.1. The Evidence Theory for Fault Diagnosis. The evidence theory developed by Dempster [31] was extended and refined by Shafer [32]. The evidence theory is related to Bayesian probability theory in the sense that they both can update subjective beliefs given new evidence. The major difference between the two theories is that the evidence theory is capable of combining evidence and dealing with ignorance in the evidence combination process. Specifically, it can distinguish the events such as "do not know," "uncertainty," and other important concepts on cognition. The basic concepts and definitions of the evidence theory relevant to this paper are briefly described as follows.

Let $\Theta = \{F_1, \ldots, F_n\}$ be a collectively exhaustive and mutually exclusive set of hypotheses, called the frame of discernment. A basic probability assignment (BPA) is a function $m : 2^{\Theta} \to [0, 1]$, called a mass function and satisfying

$$m(\emptyset) = 0, \qquad 0 \le m(A) \le 1, \qquad \sum_{A \subseteq \Theta} m(A) = 1, \tag{3}$$

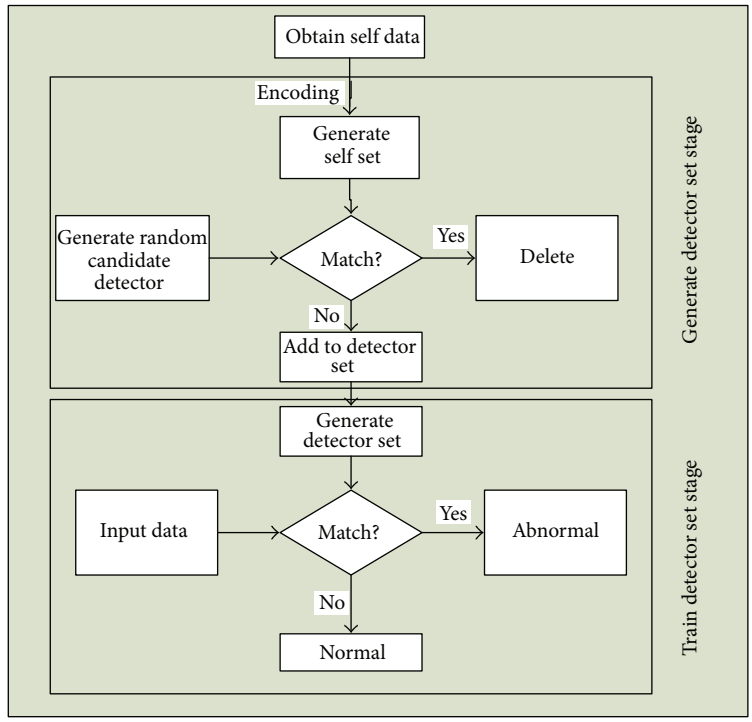


Figure 4: Negative selection algorithm.

where $\emptyset$ is an empty set, A is any subset of Θ, and 2^Θ is the power set of Θ, which consists of all the subsets of Θ, that is, $2^\Theta = \{\emptyset, \{F_1\}, \ldots, \{F_N\}, \{F_1, F_2\}, \ldots, \{F_1, F_N\}, \ldots, \Theta\}$. The assigned probability (also called probability mass) $m(A)$ measures the belief exactly assigned to A and represents how strongly the evidence supports A. All assigned probabilities sum to unity and there is no belief in the empty set Φ. The probability assigned to Θ, that is, $m(\Theta)$, is called the degree of ignorance. Each subset $A \subseteq \Theta$ such that $m(A) > 0$ is called a focal element of m. All the related focal elements are collectively called the body of evidence.

Associated with each BPA are a belief measure (Bel) and a plausibility measure (Pl) which are both functions: $2^\Theta \rightarrow [0, 1]$, defined by the following equations, respectively:

$$\begin{aligned} \mathrm{Bel}(A) &= \sum_{F \subseteq A} m(F), \\ \mathrm{Pl}(A) &= \sum_{A \cap F \neq \emptyset, F \subseteq \Theta} m(F) = 1 - \mathrm{Bel}\left(\neg A\right), \end{aligned} \tag{4}$$

where A and B are subsets of Θ. Bel(A) represents the exact support to$A$, that is, the belief of the hypothesis A being true; Pl(A) represents the possible support to A, that is, the total amount of belief that could be potentially placed in A. [Bel(A), Pl(A)] constitutes the interval of support to A and can be seen as the lower and the upper bounds of the probability to which A is supported. The two functions can be connected by the following equation:

$$\mathrm{Pl}(A) = 1 - \mathrm{Bel}\left(\neg A\right), \tag{5}$$

where $\neg A$ denotes the complement of A. The difference between the belief and the plausibility of a set A describes the ignorance of the assessment for the set A [32].

Since $m(A)$, Bel(A), and Pl(A) are in one-to-one correspondence, they can be seen as three facets of the same piece of information. There are several other functions such as commonality function and doubt function, which can also be used to represent evidence. They all represent the same information and provide flexibility in a variety of reasoning applications.

The kernel of the evidence theory is the Dempster's rule of combination by which the evidence from different sources is combined. The rule assumes that the information sources are independent and use the orthogonal sum to combine multiple belief structures $m = m_1 \oplus m_2 \oplus \cdots \oplus m_n$, where $\oplus$ represents the operator of combination. With two belief structures m_1 and m_2, the Dempster's rule of combination is defined as follows:

$$m_1 \oplus m_2 (C) = \begin{cases} 0, & C = \emptyset, \\ \dfrac{\sum_{A \cap B = C} m_1(A)\, m_2(B)}{1 - \sum_{A \cap B = \emptyset} m_1(A)\, m_2(B)}, & C \neq \emptyset, \end{cases} \tag{6}$$

where A and B are both focal elements and $[m_1 \oplus m_2](C)$ itself is a BPA. The denominator, $1 - \sum_{A \cap B = \emptyset} m_1(A) m_2(B)$, is called the normalization factor, and $\sum_{A \cap B = \emptyset} m_1(A) m_2(B)$ is called the degree of conflict, which measures the conflict between the pieces of evidence. Several researchers have investigated the combination rules of evidence theory and fuzzy sets. Note that the crude application of the D-S theory and the combination rule can lead to irrational conclusions in the aggregation of multiple pieces of evidence in conflict.

Table 4: The specific integrated diagnostic process.

Step 1	We first measure vibration signals of a variety of failure modes in rotating machinery through a large number of experiments and generate seven mature immune detectors corresponding to seven nondimensional parameters by the artificial immune system.
Step 2	Based on the results obtained from Step 1, we can obtain the index ranges of different faults and calculate the fault features (such as m_{ij}, g_{ij}, and d_{ij}).
Step 3	We can measure the non-dimensional index to determine the probability of different faults by the mature immune detector. For example, after obtaining a set of experimental data, we use waveform index to determine the fault, and kurtosis index to determine the fault j and then we can empirically calculate the basic probability assignment by the cumulative number of experiment results.
Step 4	When we get the diagnosis probability of various indices to different faults by the cumulative proportion of the total number of experiments, we define the diagnosis probability of various indices to different faults as basic probability assignment functions and use evidential theory to aggregate these data.
Step 5	We assume the greatest credibility as a final diagnostic result. That is, the fault is the one with the greatest credibility in the aggregated results.

Table 5: The range of each nondimensional parameter.

Fault type	S_f	C_f	I_f	CL_f	K_v	N_1	N_2
Normal	1.215~1.227	0.917~0.931	1.213~1.232	1.135~1.186	2.205~2.297	3.463~3.809	1.467~1.727
Grinding teeth	1.242~1.257	1.320~1.341	1.923~1.962	1.657~1.676	2.776~2.872	6.326~6.829	0.974~1.182
Inside and outside ring grinding	1.257~1.287	1.223~1.320	1.849~1.923	1.562~1.657	2.515~2.776	4.906~6.289	5.062~5.542
Concurrent fault	1.287~1.357	1.007~1.220	1.465~1.849	1.252~1.562	2.045~2.205	3.058~3.455	8.162~8.504

The evidence theory is applied to the fault diagnosis through the synthesis of information to get some basic probability assignment functions. After obtaining these basic probability assignment functions, the D-S combination rule can be used to aggregate them to get a final judgment with certain belief degrees in the focal elements. This final judgment can characterize the uncertainty in fault diagnosis and thus is useful to avoid the false alarm in final decision.

5.2. An Integrated Concurrent Fault Diagnosis Method Using Artificial Immune Algorithm and Evidence Theory. In the following section, we will present an integrated method using artificial immune algorithm and evidential theory for fault diagnosis.

Considering the existing dimensionless parameters is insensitive to the change of working condition, so it is applicable to fault diagnosis technology, yet it is not very satisfactory for concurrent fault diagnosis in rotating machinery, and the number of the conventional dimensionless parameters is few. In order to overcome the disadvantages of conventional dimensionless parameters in rotating machinery fault classification, building new dimensionless parameter, which especially possesses the features of integrated diagnosis, is of great significance to improve capability of diagnosing fault and analyzing fault characteristics. A new method based on genetic programming is proposed to construct the new dimensionless parameters [33]. According to this method, simple dimensionless parameters are combined and new dimensionless parameters are formed, then fitness function is adopted to measure the performance of new generated indexes. With this method, two new dimensionless parameters having better classification ability than that of existing ones are obtained for fault diagnosis of rotating machinery. These two new dimensionless parameters are as follows:

(i) $N_1 = K_v^2 + C_f^2 - CL_f I_f$;

(ii) $N_2 = (S_f K_v + I_f - C_f CL_f)(S_f I_f - C_f^2)$.

In our paper, we use these two parameters with five previous parameters to detect the fault. The specific integrated diagnostic process is summarized in Table 4.

The above integrated approach can be illustrated by Figure 5. We assign different weight coefficients to seven dimensionless indexes before aggregation using the evidence theory. This is based on the fact that each index has different diagnosis capability and sensitivity for different faults. Such technology via assigning weights for each index can improve the reliability of diagnosis results and avoid possible incorrect conclusion resulted by the evidence conflict. Many scholars have adopted this idea to improve the combination methods of evidence theory. The details can be found in these literatures and thus we do not touch them here to save the space.

6. A Case Study (See Figure 6)

Using our testing equipment, the experimental data are measured as shown in Table 5.

From the range of the indexes which are listed in Table 5, we can calculate m_{ij}, g_{ij}, and d_{ij} using our developed approach in Section 4.2. We do not list these indexes due

Table 6: The obtained basic probability assignment functions from seven indexes.

Fault type	Value of weighting						
	S_f	C_f	I_f	CL_f	K_v	N_1	N_2
Grinding teeth	0.41	0.32	0.39	0.4	0.14	0.19	0.11
Inside and outside ring grinding	0.23	0.35	0.22	0.28	0.45	0.58	0.34
Concurrent fault	0.36	0.33	0.39	0.32	0.41	0.23	0.55

Table 7: The final results of our integrated approach.

	Grinding teeth	Inside and outside the ring grinding	Concurrent fault	Diagnosis
M_2	0.4	0.24	0.36	Grinding teeth
M_3	0.44	0.16	0.4	Grinding teeth
M_4	0.51	0.12	0.37	Grinding teeth
M_5	0.13	0.11	0.76	Concurrent fault
M_6	0.1	0.23	0.67	Concurrent fault
M_7	0.02	0.17	0.81	Concurrent fault

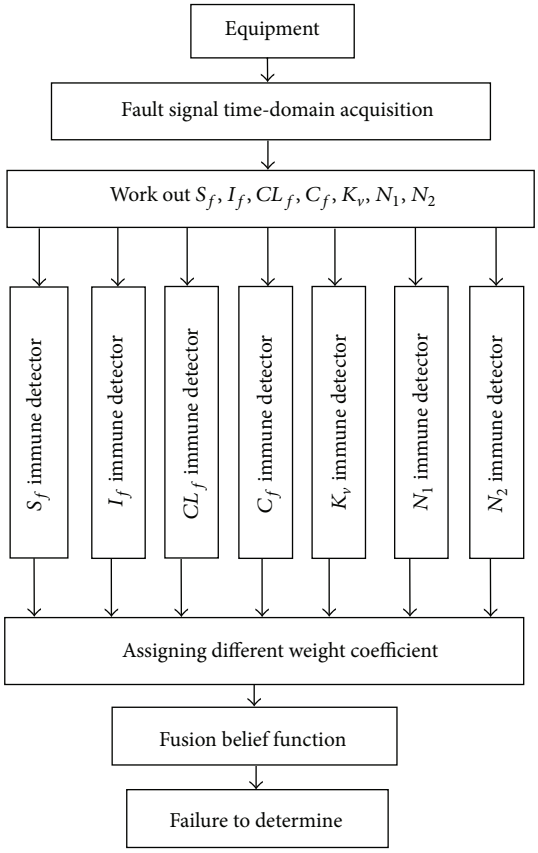


Figure 5: Schematic of the developed diagnosis approach.

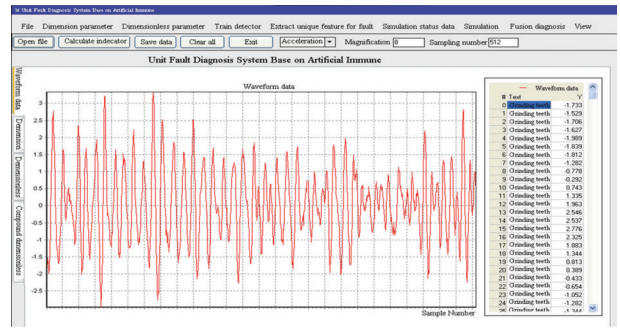


Figure 6: Waveform of the acceleration under grinding fault.

to the limited space. Based on these data, we choose d_{ij} as the relative weighting factor and then calculate the diagnosis probability of each index to different faults as basic probability assignment function. The normalized results which were calculated by our approach are shown in Table 6.

Based on the basic probability assignment functions summarized in Table 6, we then use evidential theory to fuse the data. The final aggregated results are shown in Table 7.

In Table 7, M_2 represents the fusion values of the first two normalized distribution of weight data in Table 6. The M_3 represents the fusion values of the first three normalized distributions of weight data, and so on. Finally, after aggregating all these seven basic probability assignment functions, we can find that the concurrent fault will occur with credibility 0.81, and thus we have enough evidence to claim that the concurrent fault occurs and actually this result matches the right fault result occurring in our experimental case. This demonstrates the implementation and effectiveness of our integrated approach for the concurrent fault.

7. Discussion

To achieve concurrent fault diagnosis for rotating machinery, which cannot be accurately diagnosed by the current methods, we develop an integrated method using artificial immune algorithm and evidential theory to solve this problem. Our developed method can take advantages of artificial

Table 8: The comparison of three methods in fault diagnosis.

Method	Strengths	Weaknesses
AIS [8, 9]	AIS can deal with the problem that the solution of fault diagnosis is lack of fault samples; the reasonable antibody set could be obtained from normal specimens using this algorithm.	This algorithm needs relatively large antibody set to ensure a higher detection rate, which leads to taking long time to generate detector and detect faults. In addition, the threshold value of the joint stress between antigen and antibody and the self radius is currently not well resolved. The designed threshold value has limitations for different faults
Evidence theory [10]	D-S evidence theory has a strong ability of dealing with uncertain information, which can effectively improve the credibility of diagnosis and reduce diagnostic uncertainty.	In the case of serious conflicting evidence, the result using the D-S evidence theory to fuse information directly does not agree with practice situation. The method of determining the reliability of evidence needs to be studied.
An integrated method using AIS and evidence theory presented in this paper	The integrated diagnostic technology uses more types of dimensionless immune detectors and decision-making systems of the evidence theory to detect fault; the excellent detector can be derived through multi-information fusion technology.	There is a subjective, human element when assigning a weight to dimensionless immune detector. We need experience to deal with the weight.

immune algorithm and evidential theory and thus has ability to diagnose the fault in a quick and effective manner. In particular, our method can represent the uncertainty existing in the result of fault diagnosis due to the evidence theory involved. In order to demonstrate our method, we apply our method to the case of concurrent fault diagnosis for rotating machinery. The results show that our developed method can make accurate judgments for concurrent fault of the rotating machinery under consideration, and the diagnosis result has certain credibility. Finally, we carried on some comparison with other algorithms in Table 8.

Acknowledgments

This work was partially supported by the NSFC under Grant 61174113 and the Natural Science Fund of Guangdong Province under Grant S2011020002735.

References

[1] Y. G. Lei, Z. J. He, and Y. Y. Zi, "A new approach to intelligent fault diagnosis of rotating machinery," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1593–1600, 2008.

[2] J. Lin and L. S. Qu, "Feature extraction based on morlet wavelet and its application for mechanical fault diagnosis," *Journal of Sound and Vibration*, vol. 234, no. 1, pp. 135–148, 2000.

[3] X. S. Si, C. H. Hu, J. B. Yang, and Q. Zhang, "On the dynamic evidential reasoning algorithm for fault prediction," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5061–5080, 2011.

[4] X. S. Si, C. H. Hu, J. B. Yang, and Z. J. Zhou, "A new prediction model based on belief rule base for system behavior prediction," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 4, pp. 456–471, 2011.

[5] G. Z. Dai, Q. Pan, S. Y. Zhang, and H. C. Zhang, "The developments and problems in evidence reasoning," *Control Theory and Applications*, vol. 16, no. 4, pp. 465–469, 1999.

[6] P. K. Harmer, P. D. Williams, G. H. Gunsch, and G. B. Lamont, "An artificial immune system architecture for computer security applications," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 252–280, 2002.

[7] X. F. Fan and M. J. Zuo, "Fault diagnosis of machines based on D-S evidence theory. Part 2: application of the improved D-S evidence theory in gearbox fault diagnosis," *Pattern Recognition Letters*, vol. 27, no. 5, pp. 377–385, 2006.

[8] Z. H. Han, F. Wang, X. D. Hao, and S. Liu, "Fault diagnosis of turbine vibration based on artificial immune algorithm," *Journal of North China Electric Power University*, vol. 37, no. 3, pp. 38–42, 2010.

[9] Y. Peng, C. L. Zhang, H. Zhao, and X. Yue, "Fault diagnosis of nuclear equipment based on artificial immune system," *Nuclear Power Engineering*, vol. 29, no. 2, pp. 124–128, 2008.

[10] P. Zhao and Z. S. Wang, "Aero-engine rotor fault diagnodis based on dempster-shafer evidential theory," *Machinery Design & Manufacture*, vol. 1, pp. 136–137, 2008.

[11] Q. H. Meng, X. J. Zhou, and Y. C. Wu, "Vehicle fault diagnosis based on wavelet-immune system," *Automotive Engineering*, vol. 26, no. 5, pp. 619–622, 2004.

[12] M. A. K. Jaradat and R. Langari, "A hybrid intelligent system for fault detection and sensor fusion," *Applied Soft Computing*, vol. 9, no. 1, pp. 415–422, 2009.

[13] I. Aydin, M. Karakose, and E. Akin, "A multi-objective artificial immune algorithm for parameter optimization in support vector machine," *Applied Soft Computing*, vol. 11, no. 1, pp. 120–129, 2011.

[14] C. J. Wang, S. X. Xia, and Q. Niu, "Artificial immune particle swarm optimization for fault diagnosis of mine hoist," *Acta Electronica Sinica*, vol. 38, no. 2, pp. 94–98, 2010.

[15] W. L. Jiang, H. F. Niu, and S. Y. Liu, "Composite fault diagnosis method and its verification experiments," *Journal of Vibration and Shock*, vol. 30, no. 6, pp. 176–180, 2011.

[16] Z. Wanjun, W. Xin, and L. Xinliang, "Mixed diagnosis tactic on fuzzy-immunity of gun-launched missile system," *Ordnance Industry Automation*, vol. 31, no. 16, pp. 1–64, 2012.

[17] T. Yüksel and A. Sezgin, "Two fault detection and isolation schemes for robot manipulators using soft computing techniques," *Applied Soft Computing*, vol. 10, no. 1, pp. 125–134, 2010.

[18] S. Sampath and R. Singh, "An integrated fault diagnostics model using genetic algorithm and neural networks," *Journal of Engineering for Gas Turbines and Power*, vol. 128, no. 1, pp. 49–56, 2006.

[19] R. Naresh, V. Sharma, and M. Vashisth, "An integrated neural fuzzy approach for fault diagnosis of transformers," *IEEE Transactions on Power Delivery*, vol. 23, no. 4, pp. 2017–2024, 2008.

[20] Y. Y. Wang and Q. J. Li, "Research on fault diagnosis expert system based on the neural network and the fault tree technology," *Procedia Engineering*, vol. 31, pp. 1206–1210, 2012.

[21] S. W. Fei and X. B. Zhang, "Fault diagnosis of power transformer based on support vector machine with genetic algorithm," *Expert Systems with Applications*, vol. 36, no. 8, pp. 11352–11357, 2009.

[22] L. Ping, "Fault diagnosis for motor rotor based on KPCA-SVM," *Applied Mechanics and Materials*, vol. 143-144, pp. 680–684, 2011.

[23] H. Guo, L. B. Jack, and A. K. Nandi, "Feature generation using genetic programming with application to fault classification," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 35, no. 1, pp. 89–99, 2005.

[24] C. Pan, W. Chen, and Y. Yun, "Fault diagnostic method of power transformers based on hybrid genetic algorithm evolving wavelet neural network," *IET Electric Power Applications*, vol. 2, no. 1, pp. 71–76, 2008.

[25] V. T. Tran, B.-S. Yang, M.-S. Oh, and A. C. C. Tan, "Fault diagnosis of induction motor based on decision trees and adaptive neuro-fuzzy inference," *Expert Systems with Applications*, vol. 36, no. 2, part 1, pp. 1840–1849, 2009.

[26] B. Samanta and C. Nataraj, "Use of particle swarm optimization for machinery fault detection," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 2, pp. 308–316, 2009.

[27] C. F. Dong, X. B. Wei, and T. Y. Wang, "A new method for diagnosis research of compound faults rotor in turbine generator set," *Turbine Technology*, vol. 45, no. 6, pp. 377–379, 2003.

[28] S. Forrest and S. A. Hofmeyr, "Immunology as information processing," in *Design Principles for the Immune System and Other Distributed Autonomous Systems*, L. A. Segel and I. R. Cohen, Eds., Oxford University Press, New York, NY, USA, 2000.

[29] Q. H. Zhang, A method of rotating machinery fault diagnosis based on non-dimension immune detectors. China. Utility Model Patent. CN101000276 2007-07-18.

[30] Q. H. Zhang, *The research on unit fault diagnosis technology based on artificial immune system [Ph.D. dissertation]*, South China University of Technology, Guangzhou, China, 2004.

[31] A. P. Dempster, "Upper and lower probabilities induced by a multi-valued mapping," *Annals Mathematical Statistics*, vol. 38, pp. 325–339, 1967.

[32] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, 1976.

[33] J. P. Xuan, T. L. Shi, G. L. Liao, and W. X. Lai, "Classification feature extraction of multiple gear faults using genetic programming," *Journal of Vibration Engineering*, vol. 19, no. 1, pp. 70–74, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 362649, 10 pages
http://dx.doi.org/10.1155/2013/362649

Research Article

An Probability-Based Energy Model on Cache Coherence Protocol with Mobile Sensor Network

Jihe Wang,[1] Bing Guo,[1] and Meikang Qiu[2]

[1] *Computer Science College, Sichuan University, Chengdu, Sichuan 610064, China*
[2] *Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY 40506, USA*

Correspondence should be addressed to Meikang Qiu; mqiu@engr.uky.edu

Received 12 January 2013; Accepted 3 April 2013

Academic Editor: Jianwei Niu

Mobile sensor networks (MSNs) are widely used in various domains to monitor, record, compute, and interact the information within an environment. To prolong the life time of each node in MSNs, energy model and conservation should be considered carefully when designing the data communication mechanism in the network. The limited battery volume and high workload on channels worsen the life times of the busy nodes. In this paper, we propose a new energy evaluating methodology of packet transmissions in MSNs, which is based on redividing network layers and describing the synchronous data flow with matrix form. We first introduce the cache coherence layer to the protocol stack of MSNs. Then, we use a set of energy probability matrices to describe and calculate the energy consumption of each state in the protocol. After that, based on our energy model, we will give out an energy evaluating method of the MSNs design, which is suitable for measuring and comparing the energy consumption from different implements of hardware/software. Our experimental results show that our approach achieves a precision with less than 2% error and provides a credible quantitative criterion for energy optimization of data communication in MSNs.

1. Introduction

A *Mobile sensor network* (MSN) is a wireless sensor network which is widely used in many industrial and consumer applications, such as industrial process monitoring and control and machine health monitoring. Various kinds of MSNs, such as *vehicular ad hoc networks* (VANETs), *underwater sensor networks* (UWSNs), and *wireless body area networks* (WBANs), have been widely researched and developed to provide ubiquitous solutions for real-time monitoring [1]. A sensor network usually comprises some node sensors, wireless communication devices, microcontrollers, and power source. As a mobile device, the limited battery volume cannot supply the durable power to the sensor node [2], known as *power wall*. Sensors in a wireless sensor network are prone to failure, due to energy depletion [3]. In order to save the energy consumption on communication, some base stations are added into the network as a center of a cluster of near nodes. These base stations are able to send queries and gather the data from the sensor nodes. Depending on the applications, the sensors are deployed randomly or using a systematic approach to gather the information from the environment [4]. Making effective use of multiple memory modules remains difficult, considering the combined effect of performance and power requirement [5]. Because fetching and synchronizing the data in remote node is a high energy consuming procedure, caching the data in the memory media of local node can effectively decrease the energy consumption on data communication. However, to measure and evaluate the energy consumption of cache coherence among a large number of nodes is not easy. (1) The application's data operations are randomly happening on each node, which make the energy consumed in communication unpredictable. (2) In a distributed cache system, synchronizing a data between a pair of nodes includes several steps, such as requiring, responding, and transmitting, and each step is independent from the other one, which increase the uncertainty of the procedure. So, precisely describing the complicated energy consumed in the procedure with mathematical tools is the key solution of this issue.

Generally, the energy of a node can be divided into three kinds: listening energy, mobility energy, and communication

energy. The listening energy is consumed when the sensor node is in inactive state, in which, the sensor's radio module is ready to transmit or receive data at any time. The mobility energy comes from mobile sensor physical movements, and the power dissipation through the movement of sensors is dependent on the number of sensors moved during the deployment phase. The communication energy is the energy consumed when the message is exchanging among sensors [6]. We focus on the communication energy in MSNs because (1) the listening energy is the statical part of the energy unrelated with applications; (2) in our architecture, the mobility energy can be ignored when a mobile node only moves around its master base station. As the dynamic part of the energy consumption, the communication energy is completely driven by the applications running on each node, which can be seen as the source of remote data fetching and caching.

It is foreseen that huge gains are achievable both in terms of overall MSN energy saving and data availability, if caching is properly implemented in MSN. Since queries can be serviced at nearby node or local cache, caching reduces data access time and thus obviates the need for query to travel to actual data source that would be quite far from the querying node [7]. There are two valid cache strategies of caching in MSNs: one is to cache the recently used data in the mobile node waiting for being reused by the local (or nearby) sensor; another one is to cache the data in the master base station of the mobile node so that all sensors in one cluster can share the cache data stored in their public base station. Our energy model supports both of the two cache strategies.

With the traditional ISO/OSI network, the stack is divided into seven layers, from the physical layer to the application layer. However, this partition cannot be directly migrated to the MSN with cache coherence, because more conditions should be considered carefully: (1) the data communication comes from the sensor's remote fetching operation, instead of application's *send* and *receive* functions; (2) if the remote data is written by an arbitrary node and has not been synchronized over the network, the received message might contain dirty data; Therefore, we redivide the upper layers of OSI network so that the cache coherence protocol becomes the kernel application in MSN. Based on the new network stack, our energy model of cache coherence protocol can be built to describe the energy dissipating over the network.

In this paper, we propose a new energy evaluating methodology of packet transmissions in MSNs, which is based on redividing network layers and describing the synchronous data flow with matrix form. Specifically, our main contributions are as follows.

(1) A new cache coherence protocol is inserted into the network stack of MSNs, based on which, all data communication in the MSNs can be replaced by the movements of state in the protocol.

(2) We propose a high accuracy probability-based energy model for the cache coherence protocol. It combines the energy model of network components, such as routing unit and link on base stations to measure the energy distribution of single message in MSNs.

(3) We propose a new energy comparison method to evaluate different types of cache coherence protocols for single message, which strongly depend on cache organizations, network design, and run-time environments.

The remainder of this paper is organized as follows. Architecture and data fetching procedure of the cache coherence protocol are presented in Section 2. Our probability-based energy model for the the protocol is described in Section 3. The new energy comparison method is depicted in Section 4. Experimental results are shown in Section 5. Section 6 contains some related work. Finally, Section 7 concludes the paper.

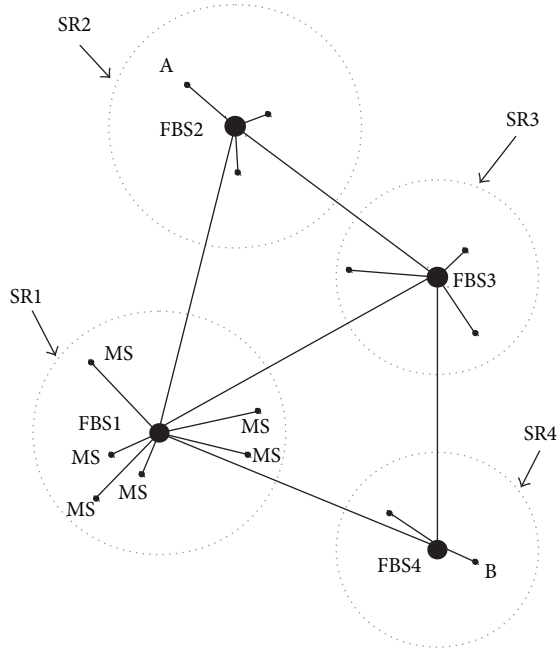
2. Architecture and Data Fetching Procedure

A typical mobile wireless sensor network includes some interconnected fixed base stations (FBSs), more mobile sensors (MSs) around these FBSs, and the sensing regions (SRs) of each FBS define the corresponding FBS' control area. Based on the architecture, a network stack is set up to provide the network services to the applications running in each MS. Using caching mechanism, the remote data reading and writing services can be provided by the cache coherence protocol layer in the stack, so that all applications can invoke the synchronization functions identically.

2.1. Architecture Overall. As mentioned above, components of an MSN, for example FBSs and MSs, are organized as a hierarchical fashion. Figure 1 provides an illustration of the typical architecture of the MSN. End MSs communicate with each other with the mechanisms available through their directly connected master FBS.

Mobile Sensor. Abstractly speaking, a sensor is a device capable of measuring a physical quantity and transforming it into a format that can be correctly interpreted by an instrument, namely, a computer or digital device. However, the current mobile sensors, such as the ends of mobile internet, have high performance on computing. They are able to process some simple and complicated tasks with local processors, which decreases the workload of communication, while demands more complex data synchronization mechanism for the distributed sensor. In this situation, the energy consumed by network is mainly used for fetching the remote data from other locations.

Fixed Base Station. A FBS stays in the center of an SR and gathers data from sensor nodes through shot-range communications, so that the energy consumption of each sensor is reduced, since fewer relays are needed for the sensor to send and receive its message to and from its master FBS [8]. The communication between a pair of FBSs is a more reliable channel than the one between MS and its master FBS, such as a wired connection based on TCP. In this architecture, we assume that the relative locations of any two FBSs are concrete so that they invoke fixed routing algorithm to find the target FBS and MS.



FBS: Fixed base station
MS: Mobile sensor
SR: Sensing region

Figure 1: The architecture of typical mobile sensor network.

In the architecture, different MSs communicate with each other by the FBSs. For example, in Figure 1, if an MS A in SR2 wants to communicate with another MS B which residents in SR4 remotely, A has to send the requesting message to its master fixed base station FBS2 as an agency, and FBS2 retransmits the message to FBS4 which is the master FBS of MS B through several hops in the network, for example, FBS2 → FBS3 → FBS4. The energy consumption of this procedure combines the energy of each network component in the transition path of the message. When the application on the sensors synchronize data with a high frequency, the remote connection above is set up over and over, and the energy is wasted if the remote data does not change. Therefore, a cache coherence protocol layer is added into the traditional MSN stack so that MS can get the remote data with a prompt response.

2.2. Protocol Stack of MSN. As a packet-switching network, MSNs need a protocol stack to support various communication services at different levels. However, limited resource and power on the mobile sensors can hardly sustain an integrated stack as complex as *open system interconnection* (OSI). Therefore, a simple network topology and light stack are needed to provide high-efficiency communication services with less energy consumption. Two significant differences exist between MSN and OSI protocol stacks: (1) the goal of designing OSI is to meet the communication demands of most network environments and conditions. However, in MSN design, the cache organization affects the communication, since different cache capability on each node determines the hit rate of the sensor's applications. Thus, this may affect

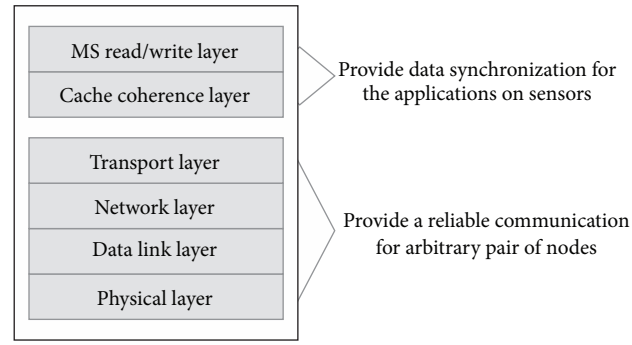


Figure 2: A new layer division of MSN stack.

communication patterns over the MSN. (2) The operating environment of OSI cannot be predicted precisely, and a series of safeguarding mechanisms are implemented to maintain the integrity and correctness of data units in each layer. However, MSN resides on distributed sensors and base station, and designers have to focus on how to utilize the limited resources and reduce the energy consumption.

As shown in Figure 2, we insert a cache coherence protocol layer to MSN stack to control the cache coherence transaction. When a read/write operation of a MS is launched, a data request is added to a request queue of its master FBS if the required data is missing in local FBS's cache. These requests will be sent into backbone network by the FBS' *network interface* (NI), and a remote data cache will respond with a synchronizing message to the requester. In this procedure, the memory address is mapped to the remote node's identifier and path to the remote node.

FBS cache missing is the reason of communication activities, and the knowledge of FBS cache is enough to set up an energy consumption model because the location of the remote responder can be deduced and the transferring path is determined by the cache coherence protocol. At any time, any node that produces a transaction can locate the required data in the shortest time. As one of the most widely used unit in cache coherence protocol, the directory table utilizes the distributed directory items to record the states of memory blocks, such as locations, sharing members, and owners. With this information, the requester traces the newest copy of a memory block, and a shortest path between the responder and the requester is reserved. We will focus on the energy consumption in this protocol because all location information can be determined and it is possible to determine the hops of a message in a specific network typology, considering fixed routing mechanism.

2.3. Data Fetching Procedure. The cache coherence protocol can set up a 3-way data fetching procedure to load the remote data into local FBS cache. We describe the fetching procedure as shown in Figure 3.

Request Step. The *request node* produces the *home node*'s location *i*; that is, the block directory information required by the *request node* may be stored into the *i* node's directory table. Then, the *request node* sends the *home node* a request

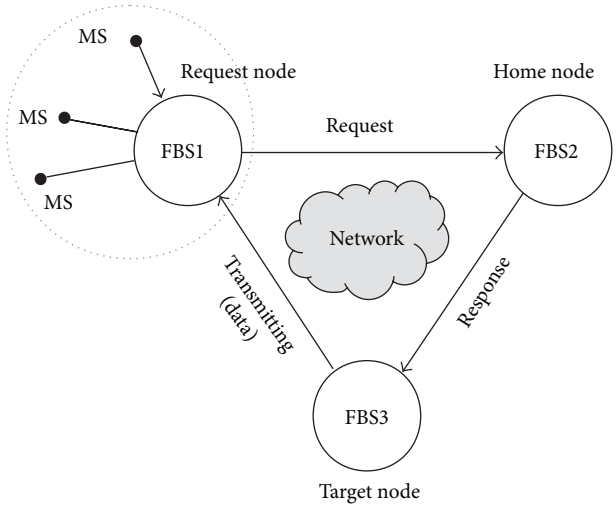


Figure 3: Procedure of data synchronization in MSN.

message which includes the memory address of the required block.

Response Step. After the *home node* receives the request message, it uses the block address to search the directory item in its directory table. If the directory item shows that there is an available image of the required memory block in a *target node*, the *home node* sends the *target node* a transfer request message which includes the required block address and the *request node*'s location.

Transmitting Step. When the *target node* receives a transfer request message, it sends a copy of the required memory block to the *request node*. Finally, remote data synchronization finishes.

For example, in Figure 3, MS launches a remote memory accessing to its master BFS1. When the FBS1 finds that the required data is not stored in the local cache, a request message to the home node of the data is sent, FBS2. The directory item in FBS2 indicates that the newest copy of the data can be obtained from a target node, FBS3, then FBS2 will notice the FBS3 to send a copy of the data to the original requester, FBS1. From the viewpoint of MS, this cache coherence protocol provides consistent accessing to any data over the network. For simplifying model, we neglect some competition-to-avoid mechanism, which is related to waiting time of the requester, while independent from the energy consumption model.

This fetching procedure is able to balance the network load when all directory information is fairly distributed in each FBS. We will further analyze the energy consumption of the 3-way data fetching procedure and build its energy evaluation model.

3. Energy Model of Data Fetching

As a widely used cache coherence unit, a directory item includes a 3-way locating procedure to obtain the newest copy of memory blocks from remote nodes, as shown in Section 2.3. This procedure simplifies the communication because write and read operations are performed in the same transaction, and each transaction also corresponds to a single packet travelling in the network. Our energy consumption model takes into account each step of the data transfer procedure.

3.1. Location of Remote Data. In this section, we use uniform memory address to locate the data in the whole MSN system. With the distributed architecture of the memory, each FBS is mapped to a piece of consecutive memory address. As the master of a set of nearby MSs, an FBS provides cache service to the MSs in its SR. Each MS in an SR can directly access the cache of its master FBS. Thus, message passing mechanism is used for an MS to access memory modules located at a remote node in the network. Therefore, cache access by an SM is not uniform since it depends on which memory address the SM wants to access. It is referred to a *nonuniform memory access* (NUMA) system. Many different cache mapping strategies have been designed for distributed cache organization to achieve a higher hit rate. We use a linear address model that is simple but scalable to find directory items in a remote FBS node.

A directed graph MSN(S_N, L) is used to model an MSN with N FBS nodes, where $s_i \in S_N$ is the ith routing nodes in the graph, i represents the location of node s_i, $0 \le i \le N-1$, and $l_{i,j} \in L$ is a directed edge between s_i and s_j. A one-dimensional flat memory can be expressed as a two-tuple $M(l, b)$, where l is the number of the basic blocks in the memory space and b is the number of parts, and each part includes l/b memory blocks. A number of distributed directory record tables map all blocks in memory and store their current states, while all records items of a part are stored in the same *home node*. If the number of parts equals the number of nodes, that is, $b = N$, then a mapping function from a block address in memory to its *home node*'s location in MSN can be represented as

$$i = \left\lfloor block\_addr \times \frac{b}{l} \right\rfloor, \quad (1)$$

where *block_addr* is the basic address of the block in memory. The addresses of all memory units in one block use the same *block_addr*. This equation can determine the location of *home node* where the block's directory record is stored. Then, the *request node* needs to launch a fetching process to obtain the state information of the block in the *home node*. We will discuss the fetching process in the following section.

3.2. Distribution of Request and Response. Our energy consumption model is based on the transitivity of packet sending probability between the request and response steps. We utilize the independency between the request and response steps to provide the matrix expressions of the transitive relationship. We also describe how to combine these probability matrices with the energy consumption of a single packet to estimate the energy distribution on node pairs of an FBS network. In MSN, all request messages requiring remote data can be seen as random events. Thus, we use $q_{i,j}$ to represent the probability of FBS i's SM asking for the block whose record item

is stored in node j. Then we utilize a matrix $\mathbf{Q}$ to describe the request sending probability distribution of the whole MSN. Specifically, (1) if the diagonal elements $q_{i,i} > 0$, a part of requests from node i hits the local cache of the master FBS, and node i does not need to look up the directory item in remote nodes. If $q_{i,i} = 0$, the required data block is always found in the remote FBS' cache. (2) Since $\mathbf{Q}$ is a probability matrix, it is natural that the sum of a row in $\mathbf{Q}$ must equal 1. (3) $q_{i,j} (i \neq j)$ could be any value between 0 and 1.

Matrix $\mathbf{Q}$ is a probability matrix that contains more hints about the transmission energy. If requests from node i are mainly requests to nearby nodes rather than to distant ones, the energy consumed in the request process could be lower. For example, in a 2×2 mesh network, consider a probability vector $\mathbf{q} = (q_{1,0}, q_{1,1}, q_{1,2}, q_{1,3})$, a row of the 4×4 matrix $\mathbf{Q}$. If the probability of this row is mainly contributed by $q_{1,0}$, $q_{1,1}$, and $q_{1,2}$, then the energy dissipation from the requests to distant nodes, $q_{1,3}$, cannot increase the overall energy consumption significantly. However, if $q_{1,3}$ is dominant, the requests from node 1 to node 3 can seriously impact overall energy consumption.

We adopt a similar approach to describe the response probability distribution. A matrix $\mathbf{P}$ can be used to express the relationship between home and target nodes. Its element $p_{j,k}$ is the probability with which the home node j sends the transfer request message to the target node k. In some specific implementations of the cache coherence protocol, designers employ different swap-out methods to delete the trashy image or spread a block's image in a limited range, where some blocks can never appear in a node or will always be accessed by a node. These optimal strategies are viewed as special probability distributions in our matrix space, which may lead to better performances, such as fewer average transmission hops and lower energy consumption. We do not intend to restrict this model in a specific implementation, so it is reasonable to assume that an image of any memory block can resident in any node, and a specific strategy always has corresponding probability distribution matrix $\mathbf{P}$.

We also list the conditions for matrix $\mathbf{P}$: (1) if the diagonal elements $p_{j,j} \neq 0$, a part of transfer request messages hits in the local home node; that is, node j is also the target node during the *response step*. If $p_{j,j} = 0$, the available copy of required data block is never found in the home node j, so all transfer request messages will be sent to the remote nodes for a copy of memory image. (2) Since $\mathbf{P}$ is a probability matrix, the sum of a row in $\mathbf{P}$ must equal 1. (3) $p_{j,k} (j \neq k)$ could be any value between 0 and 1, and that depends on the run-time environment and the swap-out strategy of each node.

In the following section, we will show how to build the energy consumption distribution using $\mathbf{Q}$ and $\mathbf{P}$.

3.3. Energy Consumption on FBS Pairs. A packet traveling through MSN motivates the activities of the components in its routing path, such as routers and links. Routing algorithms always choose the shortest path between a pair of source and sink nodes for a packet travelling through MSN. Thus, we can count the number of hops between the source node and the destination node to estimate the energy incurred by the transfer of a packet. If each node in the routing path has a homogeneous routing unit and works in the same frequency, then the energy consumption by a packet E_{packet} can be expressed as

$$E_{\text{packet}} = (h + 1) E_R + h E_L, \tag{2}$$

where h is the hop along the path of the packet transferring between a node pair; E_R and E_L are the average energy consumption of a single packet passing through a routing unit and a link, respectively.

Consider a mesh backbone MSN network with $R \times C$ nodes connected together, and these nodes take identifiers from 0 to $R \times C - 1$. If a packet travels from node i to node j, the shortest Manhattan path between the two nodes should be chosen, which includes a constant number of routers and links. On the mesh network, in order to reach node j, the packet has to go through $|(j - i)\mathbf{mod}(c)| + 1$ routers along the horizontal axis. Yet, along the vertical axis, the shortest path contains $|\lfloor j/c \rfloor - \lfloor i/c \rfloor| + 1$ routers. Thus, from (2), the energy consumed by a single packet between node i and j is

$$e_{i,j} = \begin{cases} \left[|j - i| \bmod (c) + \left| \left\lfloor \frac{j}{c} \right\rfloor - \left\lfloor \frac{i}{c} \right\rfloor \right| \right] (E_R + E_L) + E_R; & i \neq j, \\ 0; & i = j. \end{cases} \tag{3}$$

The key idea is to count the number of hops of a Manhattan path and then sum up all the energy from the components that transport a packet. In general, there are several shortest paths between two nodes that contain the same number of routers and links. However, the energy consumption is not affected by the choice of different shortest paths.

3.4. Energy Consumption of Data Transmitting. Since the energy of transmitting step takes the greatest portion of energy in MSN design, we need to decide the probability distribution of single packet's transition which includes the block data for cache synchronization.

3.4.1. Independence between Request and Response Steps. For directory table-based cache systems, synchronization is always a costly operation. A request node has to look up the directory items stored in home node before a data transmission action starts. That means, on the one hand, the requester does not know where an available copy of the block is stored in the network until the search finishes in the home node. On the other hand, when a node is searching in its directory table, it does not need to reference the location of the requester on the network. If we use one random event $A_{i,j}$ to represent that node i sends a request to node j and another random event $B_{j,k}$ to represent that node j finds a needed data stored in node k, then the events $A_{i,j}$ and $B_{j,k}$ are statistically *independent* from each other:

$$\Pr\left(A_{i,j} \cap B_{j,k}\right) = \Pr\left(A_{i,j}\right) \Pr\left(B_{j,k}\right). \tag{4}$$

In some protocol implementations, several copies of one cache block may be distributed in different nodes. These

Require: A MSN prototype; Sequence of memory accessing operation (Opt) from each MS.
Ensure: Data transmission probability matrix **T**.
(1) Get a cache missing event Opt_{RN} in request node RN
(2) $ADDR \leftarrow$ accessing address of Opt_{RN}
(3) Home node $HN \leftarrow ADDR$ with (1)
(4) $req\_home_{RN,HN} + +; req\_total_{RN} + +$
(5) Send request message to HN
(6) **if** $ADDR$ is cold **then**
(7) Forward request message to main memory
(8) **else**
(9) Get target node TN from directory table
(10) $trans\_target_{HN,TN} + +; trans\_total_{TN} + +$
(11) **end if**
(12) Send transfer request message to TN
(13) **for** Each group of three nodes $i, j, k \in NoC$ **do**
(14) $Q_{i,j} = req\_home_{i,j}/req\_total_i$
(15) $P_{j,k} = trans\_target_{j,k}/trans\_total_j$
(16) $T_{k,i} + = Q_{i,j} * P_{j,k}$
(17) **end for**
(18) Send data to RN

Algorithm 1: Forming a data transmission matrix **T**.

nodes constitute a candidate set. Any node in the set could be a potential target node in the transmitting step. We only select one node located closest to the requester in order to shorten the transmission distance, which leads to lower energy consumption. Since the candidate set is *independent* from the location of the requester, the selected target node's position is also independent from the requester, no matter how close the target node will be from the request node. Based on this independence, we construct the data transmission energy model in the following section.

3.4.2. Energy Consumption of a Single Packet Transmission. The final goal of the cache coherence protocol is to send the latest memory block image from the target node to the request node. Considering that if a packet is transferring from node k to node i, two independent events, $A_{i,j}$ and $B_{j,k}$, defined in Section 3.4.1, have happened before the data transfer. The probability of a packet transferring from node k to node i can be formulated as

$$\Pr\left(C_{k,j}\right) = \sum_{j=0}^{N-1} \Pr\left(A_{i,j}\right) \Pr\left(A_{i,j} \mid B_{j,k}\right), \tag{5}$$

thus

$$\Pr\left(C_{k,j}\right) = \sum_{j=0}^{N-1} \Pr\left(A_{i,j}\right) \Pr\left(B_{j,k}\right), \tag{6}$$

where $C_{k,i}$ is the event that a packet transfers from node k to node i. We use an $N \times N$ matrix **T** to represent the probability distribution of data transferring; then **T** can be deduced from (6) as

$$\mathbf{T} = (\mathbf{QP})^T. \tag{7}$$

The result needs to be transposed to fit the correct direction of data transmission. The meaning of elements of matrix **T** is similar to that of **Q** and **P**. Element $t_{k,i}$ is the probability of sending a packet from node k to node i. Algorithm 1 shows the procedure forming the data transmission matrix **T**.

In (3), we use $e_{i,j}$ to denote energy consumed by a packet travelling from node i to node j. Thus, $e_{i,j}$ could be a fixed parameter of the shortest path between a pair of nodes in the network. We multiply $e_{i,j}$ to each element of the data transferring probability matrix **T**, shown in (8), and get the energy distribution of fetching a single block over MSN:

$$\mathbf{E} = \left(e_{i,j}t_{i,j}\right)_{N\times N}. \tag{8}$$

The matrix **E** can be used to display the energy consumption distribution of data transferring in network. In particular, (1) the sum of a row $\sum_{i=0}^{N-1} e_{k,i}t_{k,i}$ is the expected energy when sending a packet through MSN from node k. (2) The diagonal element $e_{k,k}t_{k,k}$ equals 0, because $e_{k,k} = 0$ (see (3)).

With our energy consumption model above, the data update can be achieved conveniently. Referring the description in Section 2, a remote data update includes three steps. At first, a copy of remote data is fetched from a remote node, which can be seen as a regular data fetching procedure in our model. Then, the data is written in the local memory. If the data is set with a write-through flag, this copy of data should be sent back to the remote data so that the other node is able to use the latest data version, which is also another regular data fetching procedure:

$$E_{\text{update}} = 2 * E_{\text{fetching}} + E_{\text{localwrite}}. \tag{9}$$

Therefore, the energy consumption of a data update procedure is the sum of two regular data fetching energy and one local data writing energy, see (9).

4. Energy Estimation with Random Sending

To use our energy model to evaluate different cache organizations and application algorithms on sensors, we present a random sending solution that does not include any optimizing information. By comparing with the random sending solution, we can determine the optimization gain of a given solution. Using the expected energy of a single packet in directory protocols, we can define a random sending solution which is a complete, fair packet sending instance that satisfies $q_{i,j} = q_{i,k}$ and $p_{i,j} = p_{i,k}$ for any node identifier i, j, and k. In a random sending solution, all elements in matrix **P**, **Q**, and **T** equal $1/N$, where N is the node quantity in the MSN. This is because $q_{i,j} = p_{j,k} = 1/N$ for any node i, j, and k. Then $t_{k,i} = q_i p_k = N \times (1/N)^2 = 1/N$, where q_i and p_k are the ith row vector of matrix **Q** and kth column vector of matrix **P**, respectively.

In the random sending solution, the sending probability between any pair of FBS nodes is uniform, so there is no information about data aggregation and address optimization in this solution. In other words, it's unnecessary to adopt a solution whose energy performance is worse than the random one. We use a scalar α to represent the energy optimization gain for a given solution, and $e_{i,j}^{\text{given}}$ and $e_{i,j}^{\text{random}}$ are elements of matrix **E** of a given and random solution, respectively

$$\alpha = \frac{\sum_{i,j} e_{i,j}^{\text{random}}}{\sum_{i,j} e_{i,j}^{\text{given}}}. \tag{10}$$

Based on the uniformization, α is able to evaluate the energy optimal depth of any given solution because the random solution can be calculated without any measurement. A bigger α indicates that the given solution can save more energy during the transmission of a single packet from target node to the requester. We will use this ratio to evaluate different energy optimal solutions in our experiments section.

5. Model Evaluation

In order to verify the effectiveness of our energy model, we perform two experimental cases. Firstly, we use simulator to mimic the behaviours of cache coherence protocol and test the accuracy that characters the performance of this model in a more practical context. Secondly, the model is inserted into mobile sensors network to help to measure the energy consumption of a set of application we developed.

5.1. Evaluation Methodology. In the evaluation, we developed our simulator as our experimental platform. Our simulator contains 16 FBSs organized as a 4 × 4 mesh network. Each of the FBS nodes master a random number of MS, from 1 to 255, and a directory table. The cache stores the cached data by the FBSs; the directory table keeps the locating records for part of memory. We use (1) to map a specific address to the position of its directory table. To demonstrate the accuracy of our energy model, we implement a random packet sending system as a reference. Three packet sending patterns are used to test the model in different environments.

Evaluation Methodology. The basic method to measure the energy consumption of the MSN is to set an energy monitor in each link and router. When a packet arrives at a monitor of a link, a basic energy value is added to the link's energy property. When a packet enters a router, the energy consumption of router's components, such as *buffer* and *crossbar* are also accounted for using an average energy value for one packet. The statistic data of links and routers are used at the end of simulation to calculate the overall energy consumption of the MSN system.

Configuration of Network. As an event-driven network simulator, our simulator provides flexible interfaces to create various network units and sample these units' parameters. In our 4 × 4 mesh network, each pair of neighbouring nodes is connected with a link which has 10 Mbps bandwidth. This bandwidth guarantees that there is no packet loss. The link's basic energy value mentioned above is also initiated based on this active frequency. When a data transmission event occurs, the synchronous packets have a length of 512 bytes which fit one data block of memory and cache line. In the NI of a target node, a packet is divided into 4 messages, and the routers in their transfer path only launch the routing algorithm for the head message. After the tail flit leaves the router, all resources allocated to this packet, such as buffer space and crossbar path, are released. We use the simple X-Y routing algorithm in the mesh to determine the shortest path between a pair of source and sink nodes.

Packet Sending Patterns. During a specific interval, node is either in the bursting state or in the idling state. In the experiments, we use the proportion of bursting time in one second to control the expected sending rate. Since a time interval is set to one second, the expectations of bursting time $E(\text{BT})$ and idling time $E(\text{IT})$ of one node satisfy $E(\text{BT}) + E(\text{IT}) = 1$. The sending probability of a node can be calculated as $E(\text{BT})/[E(\text{BT}) + E(\text{IT})] = E(\text{BT})$. We design three packet sending patterns to represent different task mapping results of an application: 3 × 3, 5 × 5, and 7 × 7 windows. For example, when the 3 × 3 window is chosen, the node in position of (i, j), where i and j are the row and column identifiers, sends packets only to the nodes in $(i-1, j-1)$, $(i-1, j)$, $(i-1, j+1)$, $(i, j-1)$, $(i, j+1)$, $(i+1, j-1)$, $(i+1, j)$, and $(i+1, j+1)$, forming a 3 × 3 sending window. This type of sending model satisfies the random feature in the sending window of node (i, j) area. The basic idea of this kind of design is that, if a task is mapped to several neighbouring nodes, the communication between the neighbours will happen with much higher probabilities than the remote node pair.

5.2. Experimental Setup and Applications. Our simulators were conducted using Intel Core i5-2410M CPU which integrates 2.30 GHz quad core. The system memory used in this machine is an 8 GB DDR3 memory with 1600 MHz clock frequency. This machine runs Linux with the 3.5.0-17-generic Kernel.

Our experiments are designed based on NS-2 to set up the simulation environment, which includes the architecture of our network design and the infrastructures of the MSN system, such as node sensors, wireless devices microcontroller,

Table 1: Single packet energy comparison and errors.

Sending type	Packet number	Platform energy (J)	Model average energy (J)	Mean error	Maximum error
7×7	6,700,208	95.6	95.8	0.18%	1.76%
5×5	5,022,234	78.3	78.2	0.07%	0.69%
3×3	2,357,410	51.2	51.3	0.14%	1.35%

and power source. There are two kinds of node sensors in the MSN system: *base station* and *movable device*. In our experiments, there are 16 *base stations* which are organized as a 4 by 4 mesh network. Each pair of neighbor base stations is connected with a 10 MHz frequency Ethernet. The positions of these base stations are stable so that the energy consumption of wireless communications between base station and movable device can be computed quickly. The movable devices in our simulator are initialed with random positions in a rectangle area. Their positions keep changing with every time interval so that the distances with their base station also vary with time. The energy consumption counting system in our simulator is designed based on packet level. When a packet passes through the network components, such as the link and router, an activity count is accumulated to compute the energy consumption for the component. If the packet arrive its target base station, the energy consumption of the wireless communication is calculated based on the current distance between the base station and the sink movable device. The similar case also happens in the packet sending action.

5.3. Accuracy Comparison of Energy Consumption. The simulation results in Table 1 shows that the stable single packet energy could be obtained without the effects from different sending probability. The different values of packets' hops only came from the different sending patterns. In 7×7 pattern, some packets are sent to farther nodes; then the average hop count is greater. While, in 3×3 pattern, all packets are limited in a small area, which leads to a shorter average transmission range. Table 1 shows the energy simulation results of the three types of sending patterns in Section 5.1. Since our simulator sends hundreds of thousands of packets to transfer the data in the 16-node MSN system, the energy results are very close to the model's forecast. Even if in the worst case, the error between the simulation's measurements and model's prediction is less than 2%. This outcome is acceptable as a validation of the energy evaluation model.

The model's stability is also important in evaluating the energy consumption of MSN. We also run the simulator with different packet sending probabilities for each type of pattern, from 0.05 to 0.95. The errors between the platform and our model are shown in Figures 4, 5, and 6. The results shows that a higher sending speed, or a larger probability, always results in less variances. This illustrates a better prediction of energy consumption. The reason is rooted in the law of large numbers. When the sending probability is higher than 0.8 each node, the simulation results stably corresponded to the model predictions and maximum errors are less than 0.2%. This results show that our model's prediction approaches the *limit* of a single packet's energy consumption.

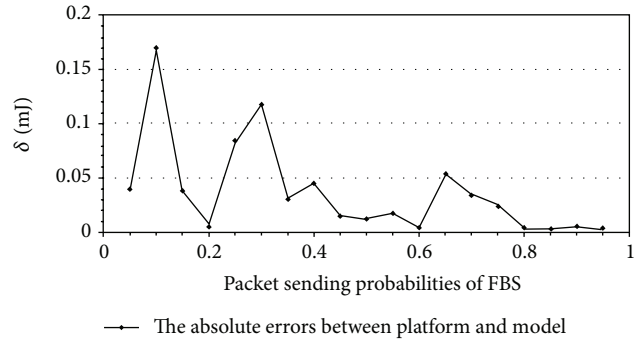


Figure 4: 7×7 window's packet energy errors between platform and model.

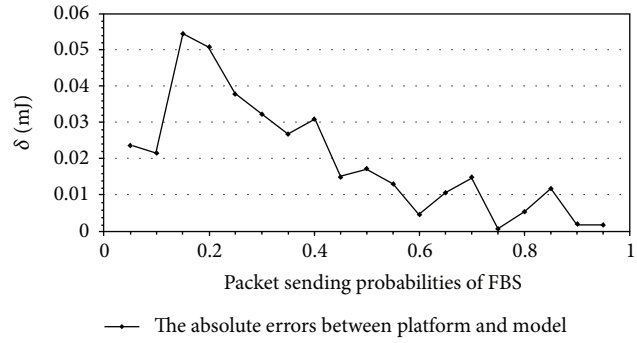


Figure 5: 5×5 window's packet energy errors between platform and model.

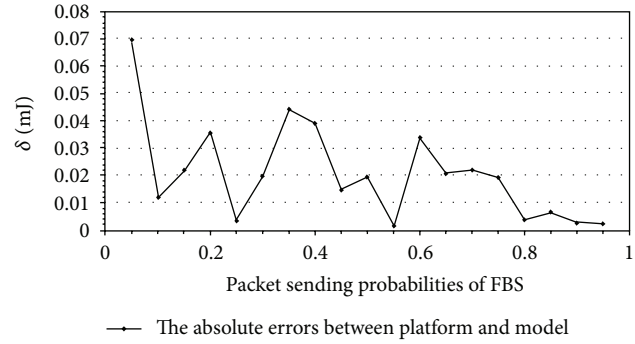


Figure 6: 3×3 window's packet energy errors between platform and model.

We implement the static energy consumption analyser for a distributed MSN system. The analyser takes the data fetching operations as inputs to record each sensor's cache access. Starting from the records, we identify the remote data accesses from local data hit and generate the remote accessing

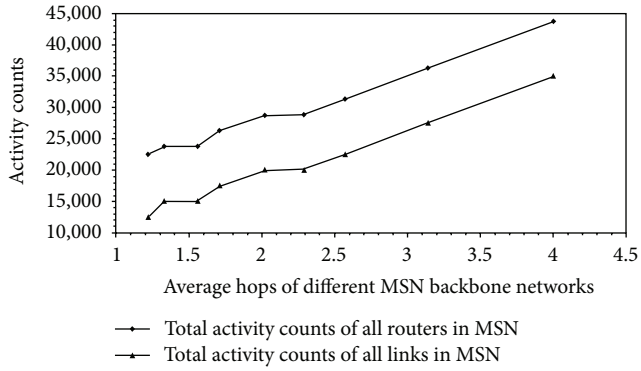


Figure 7: Energy consumption optimization for different transmitting hops.

list for each thread. During the energy evaluating stage, we iterate different shapes of windows and various locations of main thread with our energy model to get the best thread mapping pattern. The reason for iterating the location of master sensor is that master sensor always has the most communication with other subsensors, which intensively impacts the energy consumption of MSN. Figure 7 shows the relationship between average hop and the activity count of components, such as routers and links. We can conclude from the figure that the energy consumption has a linear increasing with the average hop of the network, which is irrelevant with the topology of network.

6. Related Works

Research works on energy model of mobile sensor network mainly focus on energy modelling and estimation. Reference [6] proposes a comprehensive energy model to estimate the overall energy consumption through power dissipation, for both static and mobile sensors in the potential sensor fields. Their model can calculate the total remaining lifetime of mobile sensor networks from the total power dissipation, and an optimum network strategy can be designed for a given application. Reference [9] gives an efficient hybrid method for message relaying and load balancing which is proposed in low mobility wireless sensor networks. Taking a mathematical approach, the system parameters are adjusted so that all the sensor nodes dissipate the same amount of energy, so that the problem of losing connectivity due to the fast power drainage of the closest node to the fixed sink is resolved. A 3D model of energy consumption for deploying nodes proposed in [10] describes a mathematical model for the power consumption of mobile node in wireless sensor networks. Each source node must send all its locally generated data to the other nodes and vice versa. To maximize mobile node's lifetime, it is essential to have optimum monitored region and radio range of each source node of wireless sensor networks. Reference [11] proposes a transmission scheme for power-adjustable radio to optimize transmit energy efficiency subject to given overflow and delay constraints. An analytical model is developed to estimate the unit energy, data throughput, and delay for a sensor node only in the single-hop case. Some work in [12] proposes to split the lifetime of a sensor network into equal periods of time referred to as rounds and model the energy constrained routing during a round as polynomial-time solvable flow problems. The flow information from an optimum solution to a flow problem is then used as a basis for an energy-efficient routing protocol. Reference [13] proposes a mobile cluster which is applied to a vehicle equipped with a sensor node and consists of a mobile cluster head and mobile cluster members. Their analytical results show that a mobile cluster applied to the vehicle can perform data transmission using less power than direct communication applied to the vehicle. In [14], they explore an optimal barrier coverage-based sensor deployment for object tracking wireless sensor networks where a dual-sink model was designed to evaluate the energy performance of all the static sensors, static sink, and mobile sink simultaneously. Reference [15] gives a distributed target localization and pursuit scheme based on discrete measurements of the energy intensity field produced by mobile targets. In their new strategy, all robots are categorized into two groups: the leaders, responsible for the target pursuit, and the followers, responsible for the formation and connectivity maintenance. Most of these previous work have the ability to describe the activities of wireless components and get the energy consumption in each router node or link. Some of them achieve high accuracy for single device energy consumption when simulating some applications of wireless sensor network. However, they are weak in predicting the energy consumption of the overall MSN system, because they only set up their energy model on hardware layers so that the software (coherence protocol) impact on energy consumption is ignored by the model.

7. Conclusions

Energy consumption of MSNs is a vital factor in the design of a large-scale network. A low power consumption design of wireless sensor network has to satisfy heavy workload on communication and high power-efficiency at the same time. In this paper, we analyse the data fetching procedure of a 3-way cache coherence protocol and proposed a methodology to build the energy model for it. In this model, an independence assumption is provided to guarantee that any application can be seen as a sequence of data fetching action on the wireless sensor network, and the fetching procedure is decided by the first two steps of the cache protocol. Then, we use our model to evaluate any given application that transits copy of data between different node in the MSNs system. Our model is also suitable for measuring and comparing the energy consumption from different implements of hardware and software. Our experimental results shows that our approach achieves a precision with less than 2% error and provides a credible quantitative criterion for energy optimization of cache coherence protocols with MSNs system.

Acknowledgments

Jihe Wang and Bing Guo are supported in part by the National Natural Science Foundation of China under Grant nos. 61272104 and 61073045; Sichuan Science Fund for

Distinguished Young Scholars under Grant no. 2010JQ0011; the Fund from State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences under Grant no. ICT-ARCH201003. Meikang Qiu is supported by NSF CNS-1249223 and NSFC 61071061.

References

[1] H. C. Chen, H. L. Fu, P. Lin, and C. H. Hsu, "Energy-aware transmission scheduling in mobile sensor networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '11)*, pp. 1–15, December 2011.

[2] M. Qiu, C. Xue, Z. Shao, M. Liu, and E. H.-M. Sha, "Energy minimization for heterogeneous wireless sensor networks," *Journal of Embedded Computing*, vol. 3, no. 2, pp. 109–117, 2009.

[3] M. Qiu, J. Liu, J. Li, Z. Fei, Z. Ming, and E. H. M. Sha, "A novel energy-aware fault tolerance mechanism for wireless sensor networks," in *Proceedings of IEEE/ACM International Conference on Green Computing and Communications (GREENCOM '11)*, pp. 56–61, Washington, DC, USA, 2011.

[4] S. Choudhury, S. Akl, and K. Salomaa, "Energy efficient cellular automaton based algorithms for mobile wireless sensor networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '12)*, pp. 2341–22346, April 2012.

[5] Q. Meikang and W. Jiande, "Energy saving for memory with loop scheduling and prefetching," in *Proceedings of the 18th ACM Great Lakes Symposium on VLSI (GLSVLSI '08)*, pp. 155–158, New York, NY, USA, March 2008.

[6] M. Tariq, M. Macuha, Y. J. Park, and T. Sato, "An energy estimation model for mobile sensor networks," in *Proceedings of the 4th International Conference on Sensor Technologies and Applications (SENSORCOMM '10)*, pp. 507–512, July 2010.

[7] T. P. Sharma, R. C. Joshi, and M. Misra, "Dual radio based cooperative caching for wireless sensor networks," in *Proceedings of the 16th IEEE International Conference on Networks (ICON '08)*, pp. 1–7, December 2008.

[8] O. Jerew and W. Liang, "Prolonging network lifetime through the use of mobile base station in wireless sensor networks," in *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM '09)*, pp. 170–178, New York, NY, USA, December 2009.

[9] G. F. Zaki, H. M. Elsayed, H. H. Amer, and M. S. El-Soudani, "Energy balanced model for data gathering in wireless sensor networks with fixed and mobile sinks," in *Proceedings of the 18th International Conference on Computer Communications and Networks (ICCCN '09)*, pp. 1–6, August 2009.

[10] R. Dutta and I. Bhattacharya, "Generalize 3d model of energy consumption for deploying nodes in sensor network," in *Proceedings of IEEE Recent Advances in Intelligent Computational Systems (RAICS '11)*, pp. 124–128, September 2011.

[11] C. Wang and P. Ramanathan, "Energy efficient transmission scheme for data-gathering in mobile sensor networks," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad hoc Communications and Networks (Secon '06)*, vol. 2, pp. 498–507, September 2006.

[12] S. Gandham, M. Dawande, and R. Prakash, "An integral flow-based energy-efficient routing algorithm for wireless sensor networks," *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 4, pp. 2341–2346, 2004.

[13] I. Joe and M. Shin, "An energy-efficient mobile cluster-based approach for vehicular wireless sensor networks," in *Proceedings of the 6th International Conference on Networked Computing (INC '10)*, pp. 1–5, May 2010.

[14] J. Chen, M. B. Salim, and M. Matsumoto, "Modeling the energy performance of object tracking in wireless sensor network using dual-sink," in *Proceedings of the 16th Asia-Pacific Conference on Communications (APCC '10)*, pp. 204–209, November 2010.

[15] J. Hu, L. Xie, and C. Zhang, "Energy-based multiple target localization and pursuit in mobile sensor networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 1, pp. 212–220, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 814807, 12 pages
http://dx.doi.org/10.1155/2013/814807

Research Article

Energy-Efficient Soft Real-Time Scheduling for Parameter Estimation in WSNs

Senlin Zhang,[1] Zixiang Wang,[1] Meikang Qiu,[2] and Meiqin Liu[1]

[1] *College of Electrical Engineering, Zhejiang University, 38 Zheda Road, Hangzhou 310027, China*
[2] *Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY 40506, USA*

Correspondence should be addressed to Zixiang Wang; aronlennon@yahoo.cn

Received 12 January 2013; Accepted 25 March 2013

Academic Editor: Jianwei Niu

In wireless sensor networks (WSNs), homogeneous or heterogenous sensor nodes are deployed at a certain area to monitor our curious target. The sensor nodes report their observations to the base station (BS), and the BS should implement the parameter estimation with sensors' data. Best linear unbiased estimation (BLUE) is a common estimator in the parameter estimation. Due to the end-to-end packet delay, it takes some time for the BS to receive sufficient data for the estimation. In some soft real-time applications, we expect that the estimation can be completed before the deadline with a probability. The existing approaches usually guarantee the real-time constraint through reducing the number of hops during data transmission. However, this kind of approaches does not take full advantage of the soft real-time property. In this paper, we proposed an energy-efficient scheduling algorithm especially for the soft real-time estimations in WSNs. Through the proper assignment of sensors' state, we can achieve an energy-efficient estimation before the deadline with a probability. The simulation results demonstrate the efficiency of our algorithm.

1. Introduction

Wireless sensor networks (WSNs) are emerging technologies, which can be widely applied in medicine, military, surveillance, and aerospace fields. Several sensors collaborate to accomplish high-level tasks. A WSN typically consists of a Base Station (BS) and several homogeneous or heterogenous sensor nodes. The sensor nodes are responsible for sampling the analog signal and transmit their local data to the BS. The BS acquires data from sensor nodes and does some relevant applications.

The parameter estimation is an important task in WSNs. Because the sensors' observations are corrupted by the noise, the BS should estimate the real value with the corrupted observed data. An estimator which achieves an acceptable estimation Mean Square Error (MSE) should be designed at the BS. Best Linear Unbiased Estimation (BLUE) [1] is a popular estimator in parameter estimation. Due to the bandwidth constraint in WSNs, authors in [2] propose the Quasi-Best Linear Unbiased Estimation (Quasi-BLUE). The estimator is simple and can give unbiased estimation. The works [2–6] are examples that employ Quasi-BLUE as the estimator in WSNs.

Since sensors may be deployed in hostile or remote areas, sometimes, the batteries replacement is impossible. To a certain sensor network, there is an upper bound on the lifetime [7]. Therefore, energy saving is very important in the applications of WSNs. The communication is the primary source of energy consumption [8]. The data transmission from sensor nodes to the BS can be directly sensor-to-destination scheme or multihop routing scheme. Due to the transmission power is proportional to the αth power of the transmission distance [9, 10], multihop routing scheme is widely applied in WSNs. In order to save energy, not all the sensor nodes in the network need to send their observations to the BS. In [11], the authors proposed a new topology management scheme by switching the state of the sensors. The radios of nodes can be turned off in a so-called "monitoring" state and will be switched to the "transfer" state when required. The transfer state nodes report their observations to the BS and the monitoring state nodes will not send any packet to the BS. Many works employ the idea of [11] and

schedule the state of sensor nodes to reduce energy consumption [12–15]. In this paper, an energy-efficient state scheduling scheme is designed especially for Quasi-BLUE in WSNs.

The BS should collect sufficient data from sensor nodes to implement the Quasi-BLUE. Because of the delay during data transmission, it takes some time for the BS to implement the Quasi-BLUE. The performance metric event detection delay (EDD) is used to describe the time when sufficient number of packets are delivered to the BS [16]. Because of stochastic behavior of end-to-end delay in WSNs, the previous works usually use a probabilistic model to describe the delay [17, 18]. The probability distribution of the end-to-end delay is researched in [17, 18]. That the EDD is less than a bound also satisfies a probability distribution. In some real-time applications, long EDD is not expected. However, the existing researches of Quasi-BLUE in WSNs do not consider the timing constraint. It calls for a scheme that can implement the real-time Quasi-BLUE. The real-time can be classified into hard real-time and soft real-time. In hard real-time, the system needs to finish a task before a hard deadline. The soft real-time, on the other hand, just requires the task be accomplished before the deadline with a probability. In this paper, we focus on the scheduling for the soft real-time estimation. Because the more number of hops during data transmission results in longer delay [19], the existing works usually reduce the number of hops during data transmission [20–24]. However, these approaches do not take advantage of property of soft real-time estimation. The soft timing constraint only requires a task to be finished before the deadline with a probability. In the Quasi-BLUE with an MSE constraint, more sensor nodes' data will increase the probability that the timing constraint is satisfied. Through turning some redundant nodes to transfer state, the soft timing constraint can still be guaranteed. In this paper, we add some redundant transfer state nodes to guarantee the soft timing constraint rather than reducing the number of hops during data transmission.

In this paper, we focus on soft real-time parameter estimation of WSNs. We employ Quasi-BLUE to implement the parameter estimation at the BS. The packets that carry the observations of sensors are transmitted the BS through multihop. The multihop path is the energy minimum path that can be obtained through Dijkstra's algorithm. We propose the *MSE* constraint function based nodes assignment (MBNA) algorithm to schedule the state of sensor nodes. MBNA schedules the state of sensor nodes to implement the soft real-time estimation with an MSE constraint. The contributions of this paper can be concluded as follows.

(i) We first consider the real-time for the Quasi-BLUE in WSNs.

(ii) The probability that the EDD is less than the timing constraint is quite difficult to calculate. Our approach takes advantage of linear property of Quasi-BLUE and calculates the probability in a heuristic way.

(iii) Our MBNA can achieve low energy consumption under MSE and soft timing constraints.

The paper is organized as follows. Section 2 provides some related works. In Section 3, we introduce the system model and give some assumptions in this paper. In Section 4, we show the possibility of energy reduction through adding redundant transfer state nodes. In Section 5, the energy-efficient scheduling algorithm for soft real-time estimation, MBNA, is introduced; the performance of MBNA is shown in Section 6. In Section 7, we conclude the paper.

2. Related Works

A lot of researches have been done on parameter estimation in WSNs. BLUE is a popular estimator for the parameter estimation [1, 25]. Luo makes some adjustments on BLUE, and proposes the Quasi-BLUE [2]. In Quasi-BLUE, the data is quantized to several bits, and the estimation is implemented with the quantized data. Although MSE through Quasi-BLUE increases compared to BLUE, Quasi-BLUE is quite suitable for the digital communication environment. In order to save energy, not all the sensor node will send their observations. Only part of sensors will report the observations to the BS according to the demand [11]. The estimation cannot be implemented until the BS receives sufficient data from sensor nodes, because the packet that is transmitted from source to the BS suffers an end-to-end delay. In some real-time applications, the estimation should be finished before a deadline. It requires sufficient data arrives at the BS before the deadline, and the packet delay should be considered.

Because of the randomness of wireless communication, the end-to-end packet delay shows the stochastic characterization. Many researches try to describe the delay through statistics method. In the studies in [26–28], the worst case end-to-end delay is analyzed. The low delay routing algorithms always guarantee the worst case of delay. But due to the large variance of end-to-end delay in WSNs, the worst case cannot accuratly describe the end-to-end delay. The works in [16–18, 29] employ a probability distribution to describe the delay. The delay distribution is built in [17, 18, 29], and the probabilistic description is quite suitable for the delay analysis. In this paper, we follow the probabilistic model of delay and implement our scheduling based on the results in [16–18, 29].

In order to guarantee the timing constraint, many works focus on designing the low delay routing algorithm [20–22, 24]. In WSNs, the delay during data transmission consists of the queueing delay, the transfer delay and the processing delay. Since more number of hops will increase the delay, the routing scheme decreases the delay by decreasing the number of hops. However, the energy consumption increases at the same time. The tradeoff between delay and energy is the major topic. But most low delay routing schemes do not take advantage of the probabilistic property of the delay. The approaches in [20–22, 24] are designed for the fixed delay bound and are not suitable for soft real-time scenario. The energy consumption sometimes can drop a lot while employing the soft timing constraint [30]. Through a proper scheduling scheme, heterogenous sensor nodes can cooperatively implement tasks under soft timing constraint. The

works in [13, 15, 30] are examples that implement the optimization.

In this paper, we guarantee the soft timing constraint of Quasi-BLUE through adding redundant transfer state nodes. The BS just requires sufficient data from sensors in an area for the estimation but does not specify a certain sensor. So one sensors' data can be replaced by the other sensors. If there are enough transfer state nodes, the estimation can still be finished before the deadline with a high probability. The depth-first search method is suitable for the multilevel soft real-time scheduling problem [15, 30, 31]. However, the node state scheduling problem of Quasi-BLUE is a single-level scheduling problem, and there are multiple equivalent nodes in the same level. The approaches in [15, 30, 31] are not suitable for this kind of problem. The problem is also not easy to solve through breadth-first search because a huge number of node state combinations should be listed. Our MBNA algorithm, on the other hand, does not employ the traditional search method to implement the optimization. It exploits the properties of Quasi-BLUE in WSNs, and provides the energy-efficient scheduling in a heuristic way.

3. System Model

3.1. Network Model. In this paper, we assume the WSN consists of many sensor nodes and a BS. The sensors are uniformly distributed in the sensing area. The sensor node has two states: transfer state and monitoring state. In transfer state, sensors detect the environment and transmit the observed value to the BS. In monitoring state, sensors detect the environment but do not communicate with others. The mode of a sensor node can be switched according to the command from the BS. The transfer sensors will send their observations to the BS, and the BS implements the estimation with the observed data. The state of sensor nodes is determined by the BS. Based on some performance metrics, the BS comes up with the scheduling of sensor nodes and sends the scheduling command to the sensor nodes. The sensor nodes change their states according to the command.

The sensor nodes can communicate with each other in the network. In order to save energy, the packets will be transmitted to the BS through multihop. Some nodes will be selected as the intermediate nodes during multihop packet relay. Because the BS is usually powered by the external electric source, we do not care about the energy consumption of the BS. Therefore, the BS communicates with sensor nodes directly without any intermediate nodes. In the wireless communications, we assume that the quadrature amplitude modulation (QAM) is employed. The sensor node or the BS sends an L-bit message by using QAM with a constellation size 2^L.

3.2. Quasi-BLUE. The sensors keep observing the curious parameters. The observation z_k on the real-value x made by the kth sensor s_k is corrupted by noise θ_k, which can be interpreted as

$$z_k = x + \theta_k. \tag{1}$$

If the variance σ_k of the noise θ_k is known, the BLUE estimator [1] for the real-value x is

$$\widehat{x} = \frac{\sum_{k=1}^{n} \left(x_k/\sigma_k^2\right)}{\sum_{k=1}^{n} \left(1/\sigma_k^2\right)}. \tag{2}$$

The MSE of BLUE estimator is

$$E(\widehat{x} - x)^2 = \frac{1}{\sum_{k=1}^{n} \left(1/\sigma_k^2\right)}. \tag{3}$$

The BLUE gives us a relatively accurate estimation, but it is impractical in a WSN system because of the bandwidth and energy limitation [2]. Therefore, the data is quantized to some bits at each sensor, and the estimations are implemented with the quantized data.

Suppose the value z_k observed by sensor s_k is bounded by $[-W, W]$, and it is quantized to L_k bits

$$m_k\left(z_k, L_k\right) = \begin{cases} -W + iM, & \left|z_k - iM\right| < 0.5M \\ -W + (i+1)M, & 0.5M \le \left|z_k - iM\right| < M, \end{cases} \tag{4}$$

where $0 \le i \le 2^{L_k} - 2$, $M = 2W/(2^{L_k} - 1)$.

We employ Quasi-BLUE to construct a linear estimator of x similar to BLUE estimator, and the estimator $\widehat{x}$ based on quantization is [5]

$$\widehat{x} = \frac{\sum_{k=1}^{n} \left(m_k/\left(\sigma_k^2 + \delta_k^2\right)\right)}{\sum_{k=1}^{n} \left(1/\left(\sigma_k^2 + \delta_k^2\right)\right)}, \tag{5}$$

where $\delta_k^2 = (W^2/(2^{L_k} - 1)^2)$, and the variance is

$$D = E(\widehat{x} - x)^2 = \frac{\sum_{k=1}^{n} \left(E(m_k - x)^2/\left(\sigma_k^2 + \delta_k^2\right)^2\right)}{\sum_{k=1}^{n} \left(1/\left(\sigma_k^2 + \delta_k^2\right)\right)^2}. \tag{6}$$

If we round the quantized value to the nearest endpoint of 2^{L_k} intervals, the MSE is [2]

$$D = \frac{1}{\sum_{k=1}^{n} \left(1/\left(\sigma_k^2 + \delta_k^2\right)\right)}. \tag{7}$$

From (7), it can be found that more sensors lead to more accurate estimation.

3.3. Energy Model. The energy consumption of a sensor node contains two main parts: (1) the communication energy and (2) the circuit energy. In long-range application, the data transmission consumes most of the energy in a WSN and the other energy can be neglected compared to the communication energy. Therefore, we only consider the communication energy in this paper.

When a sensor s_k finishes detecting and quantization, an L_k-bit length data will be transmitted to the BS. In a simplified model, the transmission energy can be described as a function of the data length and the transmission distance. The

channel between two nodes experiences a pathloss proportional to $a = d^{\alpha}$, where d is the transmission distance and pathloss $\alpha \geq 2$ is the pathloss exponent. If an L_k-bit packet is transmitted with the distance of d, the energy consumption using QAM with a constellation size 2^{L_k} is [9, 10]

$$E = ca\left(2^{L_k} - 1\right), \tag{8}$$

where E is the energy consumption and c is a constant during transmission. Equation (8) is the energy consumption to transmit L_k-bit length data for one hop. The energy consumption to send a packet from source node to the BS is the summation of multihop energy consumption. We denote by E_k the energy consumption which corresponds to the source node s_k.

3.4. Probabilistic Delay. Within the communication range, a link can be built between two nodes. For two sensor nodes s_i and s_j, we denote (i, j) the link between s_i and s_j. In a WSN, each link (i, j) is associated with an end-to-end delay $T_{(i,j)}$. $T_{(i,j)}$ is not stationary, and it will change during the system running. Because of the randomness in wireless communication, the end-to-end packet delay is usually described as a probabilistic model [16–18, 29]. If we know the probability density function (PDF) of $T_{(i,j)}$, the delay of (i, j) satisfies

$$P\left(T_{(i,j)} < T\right) = \int_0^T p_{(i,j)}(t)\, dt, \tag{9}$$

where $p_{(i,j)}$ is the PDF of $T_{(i,j)}$. A packet is transmitted from the source node to the BS through a multihop path, and the packet will suffer the multihop end-to-end packet delay. We denote by T_k the delay of the packets transmitted from the sensor node s_k. The delay satisfies a probability distribution. We denote by g_k the cumulative density functions (CDF) of T_k.

The probability that the delay T_k satisfies timing constraint is

$$P\left(T_k < T_d\right) = g_k\left(T_d\right), \tag{10}$$

where T_d is the timing constraint.

4. Node State Scheduling for Soft Real-Time Estimation

4.1. Motivational Example. In the multihop transmission, increasing the number of hops will increase the delay. More hop means extra processing delay, queueing delay, and transmission delay. Transmitting the packets along a path with less number of hops is a method to guarantee the timing constraint [20–24]. We call this kind of approaches the delay sensitive energy aware (DSEA) routing scheme. Through the tradeoff between energy and delay, a path will be generated based on the timing constraint. However, this method has the two drawbacks.

(1) The energy minimum path planning with timing constraint is an NP-complete problem. The existing approaches can only provide the near-optimal solution.

(2) In order to decrease the path delay, the path that satisfies the soft timing constraint has less number of hops, which will increase the communication energy.

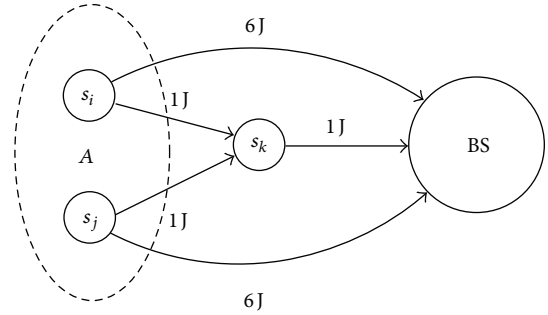


Figure 1: A simple sensor network.

In this paper, on the other hand, we try to guarantee the soft timing constraint through adding redundant nodes. In the estimation process, the BS only requires sufficient data but does not care for the source of the data. Transmitting redundant data is able to increase the $P(\text{EDD} < T_d)$. Sometimes this approach is more energy-efficient compared to planning a new path. It can be illustrated in the following example. As shown in Figure 1, there is a sensor network that consists of three sensor nodes and one BS. The BS requires at least one piece of data from sensors in the area A. Either s_i or s_j is candidate to send observations to the BS. The energy consumption for transmitting the packet through each link is shown in Figure 1. In the energy minimum routing, both the two sensor nodes transmit their data to s_k at first. Then s_k relays the data to the BS.

The delay of the two paths with the intermediate node s_k is denoted by T_i and T_j. Assume the BS requires that the data from A within 50 ms with the probability 0.7. If T_i and T_j have the following probability

$$\begin{aligned} P\left(T_i < 50\right) &= 0.5, \\ P\left(T_j < 50\right) &= 0.5, \end{aligned} \tag{11}$$

these two paths cannot guarantee soft timing constraint. The conventional approach is to generate a new path that satisfies the soft timing constraint. In this example, either s_i or s_j will transmit data directly to the BS. The direct data transmission will consume 6J energy per sensor. However, if both s_i and s_j transmit data to the BS through the energy minimum path, the probability that the BS receive the packet from s_i or s_j within 50 ms is

$$P(T < 50) = 1 - P\left(T_i \geq 5\right) P\left(T_j \geq 50\right) = 0.75. \tag{12}$$

The soft timing constraint is satisfied when redundant data is transmitted to the BS. The total energy consumption that both s_i and s_j transmit data to the BS along the energy minimum path is 4J. It can be found that adding redundant nodes can

achieve low energy consumption while satisfies the soft timing constraint.

We should still note that the approach through adding redundant transfer state nodes may not perform better than DSEA routing. The performance is tightly related to the value of end-to-end packet delay. In the Section 6, we will discuss the problem in detail.

4.2. CDF of End-to-End Delay. For a source node, the energy minimum path to the BS can be obtained through Dijkstra's algorithm with energy metric. Each path is associated with an end-to-end delay distribution. Because each sensor node corresponds to a path, we can assume that the end-to-end packet delay distributions with the same source node are identical.

The packets are sent from source node to the BS through multihop relay. In the end-to-end delay analysis, the CDFs of multihop end-to-end delay are similar among the works in [17, 29]. Because there is a physical limit in how short a delay can be (shorter than that it is impossible that a message arrives at the other end), the end-to-end delay will be larger than a lower bound. The lower bound of delay is denoted by $T_{\min}$ in this paper. A packet may be lost during transmission. In this situation, the end-to-end packet delay can be thought as infinite. Based on the experimental results in [17, 29], the end-to-end delay approximately satisfies the negative exponential distribution in the range $[T_{\min}, +\infty)$. For the packets transmitted from the source node s_k, the CDF of multihop end-to-end delay satisfies

$$g_k(t) = 1 - e^{-\mu_k t} + T_{\min}. \tag{13}$$

The parameter μ_k can be estimated through moment estimation method. During the network system running, the BS can record the end-to-end delay with different source nodes. When a sensor node send a packet, the time information will be added to the packet. The BS calculate the end-to-end packet delay based on the time information. If the the delay of different packets from s_k is $T_1, T_2, \ldots, T_n$, the estimated $\widehat{\mu}_k$ through moment estimation is

$$\widehat{\lambda}_k = \frac{1}{\sum_{i=1}^{n} T_i} - T_{\min}. \tag{14}$$

The value of $\widehat{\mu}_k$ is always updated during the system running.

4.3. Guarantee Soft Timing Constraint with Redundant Nodes. Suppose the BS requires data from an area A to implement the estimation on a parameter. The MSE constraint for the estimation is D_r. Multiple transfer state nodes will provide the observed data for the estimation. We will add several redundant transfer state nodes to guaranteed the soft timing constraint.

We denote on S_A the set that contains all the sensor nodes in the area A. With the PDFs of different path delays and the timing constraint T_d, we can calculate $P(T_k > T_d)$ of the sensor s_k. If $P(T_k > T_d) = 1$, it means the path can never satisfy the timing constraint. This kind of node will never be selected to send data to the BS. We delete this kind of node from S_A. Then we randomly select several nodes from S_A to guarantee the soft timing constraint. We denote by S_r the transfer state sensor node set. The node $s_k \in S_r$ will transmit data to the BS. At first, we should choose several transfer state nodes to implement the BLUE while satisfying the MSE constraint. If the soft timing constraint is satisfied with the set S_r, no redundant node are required. Otherwise, we should add some redundant nodes to S_r. We define a subset $\Omega \subseteq S_r$, and the sensors in Ω can provide the sufficient data for the estimation, that is,

$$\frac{1}{\sum_{s_k \in \Omega} \left(1/\left(\sigma_k^2 + \delta_k^2\right)\right)} \leq D_r. \tag{15}$$

If the data from the sensors in Ω can guarantee the soft timing constraint, we have

$$\prod_{s_k \in \Omega} P\left(T_k < T_d\right) > \gamma. \tag{16}$$

For the set S_r, we can find more than one Ω that satisfies (15). Therefore, the probability $P(\mathrm{EDD} < T_d)$ can be expressed as

$$P\left(\mathrm{EDD} < T_d\right) = \sum_{\Omega \subseteq S_r} \prod_{s_k \in \Omega} P\left(T_k < T_d\right). \tag{17}$$

Through scheduling the state of sensor nodes, $P(\mathrm{EDD} < T_d)$ can be controlled to a certain level and the soft timing constraint can be guaranteed.

4.4. Energy-Efficient Soft Real-Time Parameter Estimation. In the parameter estimation process, the BS should provide the accurate estimation with sensors' data. In this paper, we employ the MSE between the estimated value and the actual value to evaluate the accuracy of the estimation. In order to save energy, not all the sensor nodes need to send data to the BS. We just need some sensors' data to accomplish the estimation with a certain MSE constraint.

The BS collects sufficient data from different sensors, and implements the estimation. There is an event detection delay (EDD) for the WSNs [32]. The EDD is the time when sufficient number packets are delivered to the BS for the data fusion. In some real-time applications, the EDD should not be too large. A packet that is transmitted from source node to the BS corresponds to an end-to-end delay distribution [16, 17, 29]. In the network, the transfer state nodes send their packets to the BS, and the EDD is determined by the end-to-end delay distribution of each packet. In order to guarantee the soft timing constraint, the EDD should be less than a bound with a probability, that is,

$$P\left(\mathrm{EDD} < T_d\right) > \gamma, \tag{18}$$

where T_d is the timing constraint.

The assignment of transfer state nodes will affect EDD. If the transfer state nodes are not enough, the Quasi-BLUE cannot be finished within the soft timing constraint. On the other hand, if we turn too many nodes to transfer state, the energy consumption will increase. We need to schedule the

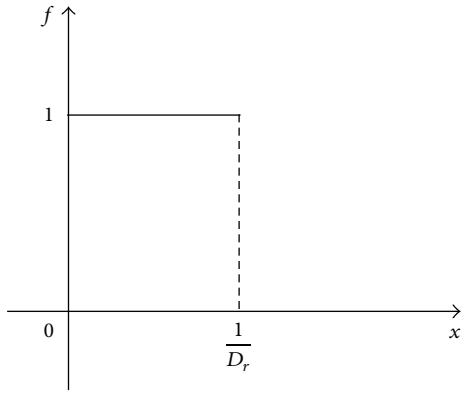


Figure 2: The MSEPF with static D_r.

state of node to achieve low energy soft real-time estimation, that is,

$$\begin{aligned} &\min \sum_{s_k \in S_r} E_k, \\ &\text{s.t. } \mathrm{D} \le \mathrm{D_r}, \\ &P\left(\mathrm{EDD} < T_d\right) > \gamma, \end{aligned} \tag{19}$$

where S_r is the transfer state node set, D_r is the MSE constraint, and T_d is the timing constraint.

5. Redundant Nodes Assignment

$P(\mathrm{EDD} < T_d)$ is the summation of all the probability of $\prod_{s_k \in \Omega} P(T_k < T_d)$. Before calculating $\prod_{s_k \in \Omega} P(T_k < T_d)$, we must list all the possible Ω. The process is time consuming. In this paper, we use a heuristic method to calculate $P(\mathrm{EDD} < T_d)$. We propose the *MSE* constraint function (MSECF) and calculate $P(\mathrm{EDD} < T_d)$ through the MSECF. Then the transfer state node set can be determined based on $P(\mathrm{EDD} < T_d)$.

5.1. MSE Constraint Function. In Quasi-BLUE, more sensors' data results in small MSE. Under a certain MSE constraint, the BS has to wait for sufficient data to guarantee the MSE constraint. Thus, the timing constraint for Quasi-BLUE is not satisfied and can also be expressed as *the estimation cannot be finished with the data that arrives before the deadline*. Therefore, $P(\mathrm{EDD} < T_d)$ is equivalent to *the probability that the estimation cannot be finished before the deadline*.

In this paper, we define the *MSE* constraint function (MSECF) $f(x)$ as

$$f(x) = P\left(\frac{1}{D_r} \ge x\right). \tag{20}$$

The function means the probability that *the reciprocal of MSE constraint is larger than x*. Within the timing constraint, the MSE achieved is denoted by D. Then $f(1/D)$ is the probability that the MSE constraint is not satisfied, and $P(\mathrm{EDD} < T_d) = f(1/D)$.

If the MSE constraint is a static value, we have

$$f(x) = \begin{cases} 1, & x \le \dfrac{1}{D_r}, \\ 0, & \text{else.} \end{cases} \tag{21}$$

The function (21) can be plotted as shown in Figure 2.

The Quasi-BLUE is implemented with sensors' data, and there is an estimation MSE D. If $1/D > 1/D_r$, we have $f(1/D) = 0$. It means that the D can guarantee the MSE constraint with the probability 1.

In the Quasi-BLUE of WSNs, the MSE constraint is usually a certain value, and the MSECF can be formulated as (21). In order to guarantee the MSE constraint, we should determine a S_r that satisfies

$$\frac{1}{D} = \sum_{s_k \in S_r} \frac{1}{\sigma_k^2 + \delta_k^2} > \frac{1}{D_r}. \tag{22}$$

When a packet from s_k is transmitted to the BS, two possible events may happen.

(i) G_k: the packet reaches the BS before the deadline.

(ii) $\overline{G}_k$: the packet does not reach the BS before the deadline.

We denote the probability that G_k happens as p_k and the probability $\overline{G}_k$ happens as q_k. With the CDF of end-to-end delay, p_k and q_k can be calculated.

$$\begin{aligned} p_k &= g_k\left(T_d\right), \\ q_k &= 1 - p_k, \end{aligned} \tag{23}$$

where g_k is the CDF of end-to-end packet delay whose source node is s_k.

The missing data will not be used while calculating $1/D$. Since the packet from each transfer state sensor node corresponds to probabilistic delay, there is a corresponding p_k for the packet transmitted from $s_k \in S_r$.

5.2. Probability for Satisfying MSE Constraint. At first, we assume that all the packets can reach the BS before the deadline, and original reciprocal MSE is

$$\frac{1}{D}^{(0)} = \sum_{s_k \in S_r} \frac{1}{\sigma_k^2 + \delta_k^2}. \tag{24}$$

If one packet transmitted from a transfer state node does not reach the BS before the deadline, it can be thought that the packet is missing. The BS has to implement the estimation without the data in that packet. Then the data will not make contribution to the estimation. If the missing packet is transmitted from s_k, the contribution of s_k should be subtracted from $(1/D)^{(0)}$. The achieved $1/D$ without data from s_k is $(1/D)^{(0)} - 1/(\sigma_k^2 + \delta^2)$. The process is equivalent to add the MSE constraint by $1/(\sigma_k^2 + \delta^2)$. Thus, the MSECF will be converted to

$$f(x) = \begin{cases} 1, & x \le \dfrac{1}{D_r} + \dfrac{1}{\sigma_k^2 + \delta^2}, \\ 0, & \text{else.} \end{cases} \tag{25}$$

The function (25) right shifts (21) for $1/(\sigma_k^2 + \delta^2)$. Because the probability for $\overline{G}_k$ is q_k, the MSECF that considers possible data missing of s_k is

$$f(x) = p_k f^{(0)}(x) + q_k f^{(0)}\left(x - \frac{1}{\sigma_k^2 + \delta^2}\right), \quad (26)$$

where $f^{(0)}(x)$ is the MSECF without considering the data missing. We introduce the operator "$\oplus$", and express (26) as

$$f^{(1)}(x) = f^{(0)}(x) \oplus G_k. \quad (27)$$

After one $\oplus$ operation, the MSECF is converted to Figure 3. $f^{(1)}(1/D)$ is the probability that the estimation cannot be finished within timing constraint while considering the possibility of $\overline{G}_k$.

Theorem 1. *Consider the following:*

$$f(x) \oplus G_i \oplus G_j = f(x) \oplus G_j \oplus G_i. \quad (28)$$

Proof. Consider the following:

$$\begin{aligned} & f(x) \oplus G_i \oplus G_j \\ &= \left(p_i f(x) + q_i f\left(x - \frac{1}{\sigma_i^2 + \delta_i^2}\right)\right) \oplus G_j \\ &= p_j\left(p_i f(x) + q_i f\left(x - \frac{1}{\sigma_i^2 + \delta_i^2}\right)\right) \\ &\quad + q_j\left(p_i f\left(x - \frac{1}{\sigma_j^2 + \delta_j^2}\right)\right. \\ &\qquad \left. + q_i f\left(x - \frac{1}{\sigma_i^2 + \delta_i^2} - \frac{1}{\sigma_j^2 + \delta_j^2}\right)\right) \\ &= \left(p_j f(x) + q_j f\left(x - \frac{1}{\sigma_j^2 + \delta_j^2}\right)\right) \oplus G_i \\ &= f(x) \oplus G_j \oplus G_i. \end{aligned} \quad (29)$$

□

According to Theorem 1, the order of $\oplus$ operation will not affect the final MSECF. All the packets in S_r may arrive at the BS after the deadline, so the above process should be applied to all sensors. After n $\oplus$ operation, the MSECF is converted to

$$f^{(n)}(x) = f^{(0)}(x) \oplus G_1 \oplus G_2 \oplus \cdots \oplus G_n. \quad (30)$$

Hence,

$$P(\text{EDD} < T_d) = f^{(n)}\left(\frac{1}{D}\right). \quad (31)$$

5.3. Nodes Assignment through MSECF. $P(\text{EDD} < T_d)$ can be calculated through MSECF. Based on the probability $P(\text{EDD} < T_d)$, we propose the *MSECF* based nodes assignment (MBNA) algorithm to determine the transfer state nodes to implement the soft real-time BLUE. The detail steps of MBNA is shown in Algorithm 1. At first, an original S_r is generated. The sensor nodes in S_r can provide sufficient data for the Quasi-BLUE with the MSE constraint. The original S_r does not have redundant nodes. So all data of sensors in S_r should arrive before the deadline. Then we calculate the probability $P(\text{EDD} < T_d)$ through $f^{(n)}(1/D)$. If $f^{(n)}(1/D) > \gamma$, the estimation under the MSE constraint D_r can be implemented with the soft timing constraint. Otherwise, we should add a redundant node to S_r and check whether $f^{(n+1)}(1/D) > \gamma$. The process continues until we obtain a $f(x)$ that satisfies $f(1/D) > \gamma$.

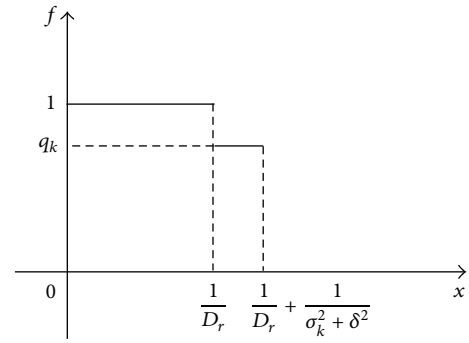


Figure 3: The MSEPF after one $\oplus$ operation.

6. Simulation Results

In this section, we present the simulation results for the MBNA algorithms.

6.1. Simulation Setup. In the simulations, we randomly deploy 100 sensor nodes in a 1000 m × 1000 m area. We make the BS located at the (500,500). The two constants of the communication energy in (8) is set as $c = 1$ and $\alpha = 2$. The maximum communication range of the sensor node is set as 500 m. The nodes can communicate with others within the communication range.

We assume that the sensors observe a parameter with the range of $[-16, 16]$ and the observations are quantized to 4-bit. The noise variance is assumed to be a stochastic value between $[1, 4]$. Each sensor corresponds to a noise variance. We schedule the states of the sensor nodes to implement Quasi-BLUE. The transfer state nodes will report their observations to the BS. At first, an original transfer state sensor set can be generated without considering the timing constraint. We randomly generate some transfer state nodes that are able to provide sufficient data for the Quasi-BLUE, and these nodes formulate the original transfer state sensor set. In soft real-time estimations, the EDD should be less than the timing constraint T_d with a probability γ. The DSEA routing approaches usually reduce the number of hops to guarantee the timing constraint and construct a low delay path. The packets travel along the low delay path so that the EDD can be reduced. This kind of scheme guarantees the timing constraint by considering the worst case end-to-end delay. The

Require: observed value range $[-W, W]$, quantization data length L, timing constraint T_d with probability γ, and MSE constraint D_r;
Ensure: transfer state sensor set S_r;
(1) $\delta = W/\left(2^L - 1\right)$;
(2) calculate p_k of every sensor node;
(3) $S_r = \emptyset$;
(4) **do**
(5) add a transfer state node to S_r;
(6) $1/D = \sum_{s_k \in S_r} 1/\left(\sigma_k^2 + \delta^2\right)$;
(7) **while** $1/D < 1/D_r$;
(8) calculate $f(x)$ according to (21);
(9) **for** $s_k \in S_r$
(10) $f(x) = f(x) \oplus G_k$;
(11) **end for**
(12) **while** $f\,(1/D) \leq \gamma$
(13) add a transfer state node to S_r;
(14) **for** $s_k \in S_r$
(15) $f(x) = f(x) \oplus G_k$;
(16) **end for**
(17) $1/D = \sum_{s_k \in S_r} 1/\left(\sigma_k^2 + \delta^2\right)$;
(18) **end while**

Algorithm 1: *MSECF Based Nodes Assignment* (MBNA) Algorithm.

MBNA, on the other hand, tries to guarantees the soft timing constraint through turning redundant nodes to the transfer state. The path of MBNA is still the energy minimum path. In this paper, we employ the approach in [19] to implement DSEA routing. We simulate the energy consumption for our MBNA and compare the results of MBNA with DSEA routing. Because the worst case of single hop delay is required for DSEA routing, we assume that the largest single hop delay is

$$t = F^{-1}\,(0.99) + \text{rand}, \tag{32}$$

where $F(x)$ is the CDF of single hop delay, $F^{-1}(x)$ is the inverse function of $F(x)$, and rand is a random value between 0 and 1. In (32), the single hop delay will be larger than $F^{-1}(0.99)$ with the probability 0.99. While considering the variation of single hop delay, we add a random value rand to $F^{-1}(0.99)$ and approximate the worst case of single hop delay as (32).

6.2. Normal Distribution Single Hop Delay Case. Normal distribution single hop delay is a common assumption in the delay analysis in WSNs. In this subsection, we simulate the performance of MBNA with the normal distribution single hop delay. The single hop delay is assumed to satisfy the normal distribution with the PDF $(1/\sqrt{2\pi})e^{(t-15)^2/18}$. In the soft real-time parameter estimation in WSNs, three factors will affect the system's energy consumption: (1) timing constraint T_d; (2) MSE constraint D_r; (3) probability for satisfying timing constraint γ.

We investigate the performance of MBNA versus T_d, D_r, and γ. We make $\gamma = 0.8$ and the MSE constraint $D_r = 0.3$ and investigate the energy consumption versus T_d. The simulation is repeated for 100 times and the result is shown in Figure 4.

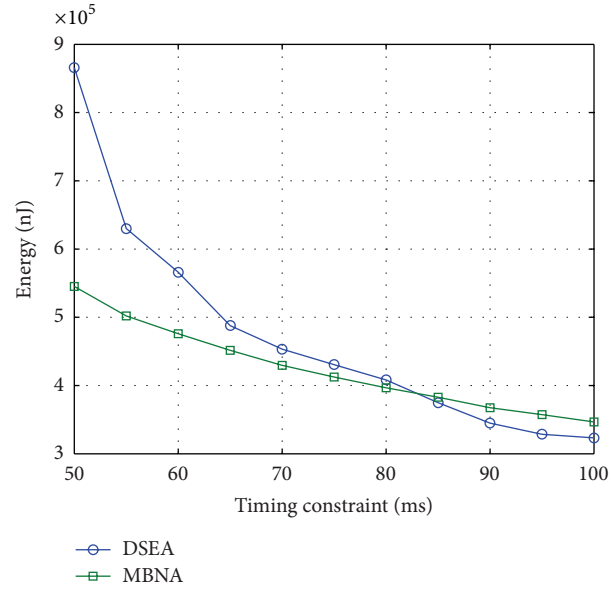


Figure 4: Energy consumption versus timing constraint. $\gamma = 0.8$ and $D_r = 0.3$.

The two curves in Figure 4 represent the average energy consumption required to implement the Quasi-BLUE. Short timing constraint means the low probability that the packet can reach the BS before the deadline. Therefore, when the timing constraint increases, the energy consumption decreases. Compared to DSEA routing, MBNA has lower

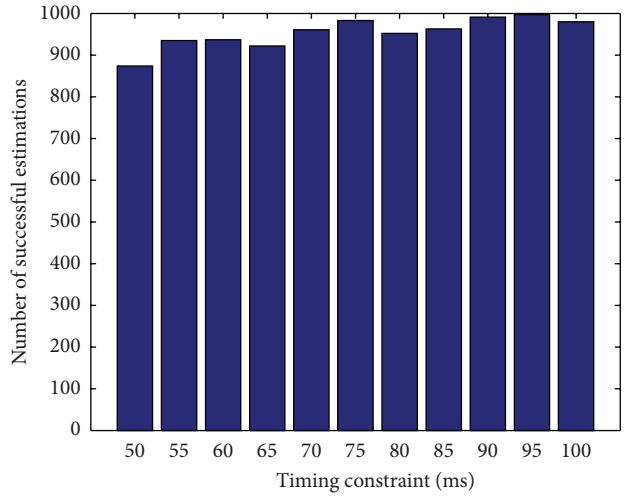


FIGURE 5: Number of successful estimation before deadline. $\gamma = 0.8$ and $D_r = 0.3$.

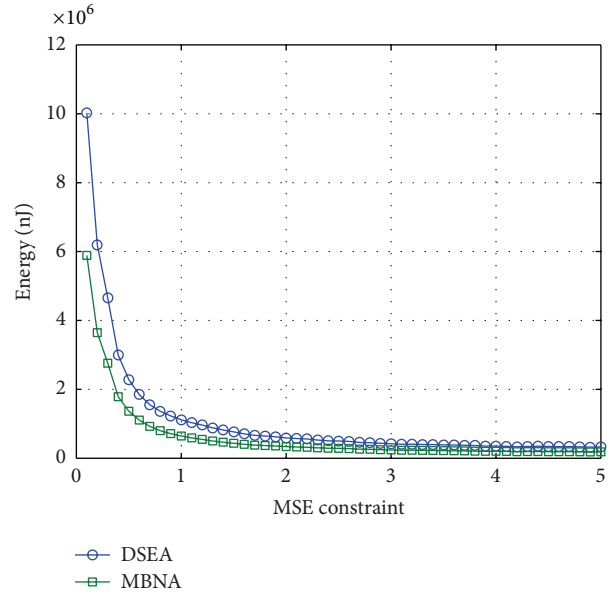


FIGURE 6: Energy consumption versus MSE constraint. $\gamma = 0.8$ and $T_d = 100$ ms.

energy consumption when the timing constraint is small. In Figure 4, DSEA routing achieves lower energy consumption than MBNA when $T_d > 82$ ms. The phenomenon is easy to understand. Because T_d is large, the packets will travel along the energy minimum path through DSEA routing. Therefore, with the same source node, the multihop path is identical for both DSEA routing and MBNA. Because MBNA requires extra transfer state nodes to guarantee the soft timing constraint, MBNA may consume more energy for Quasi-BLUE when T_d is large.

MBNA is designed for the soft timing constraint. We need the estimation to be implemented before the deadline with a probability γ. To verify that MBNA can guarantee the soft timing constraint, the number of successful estimations should be investigated. The successful estimation can be expressed as the MSE constraint is satisfied when the data arrives before the deadline. Figure 5 shows the number of successful estimations before deadline. We choose 11 different timing constraints from 50 ms to 100 ms and simulate the estimation process for 1000 times per timing constraint. We let $D_r = 1$ and $\gamma = 0.8$ during simulation. We record the number of successful estimations in the 100 times estimations. In Figure 5, the height of the bar represents the number of successful estimations. We can find that the number of successful estimation is larger than 800 for each timing constraint. It means that the Quasi-BLUE can be finished before the deadline with the probability that is larger than 0.8. The soft timing constraint can be guaranteed through MBNA.

Then we investigate the MSE constraint's influence on the performance of MBNA. We make $\gamma = 0.8$ and the timing constraint $T_d = 100$ ms. γ and T_d keep unchanged during simulation. The energy consumption with different MSE constraints is shown in Figure 6.

The result in Figure 6 represents the average energy consumption with MBNA and DSEA routing. MBNA can achieve lower energy consumption when $D_r < 4.6$. When $D_r \geq 4.6$,

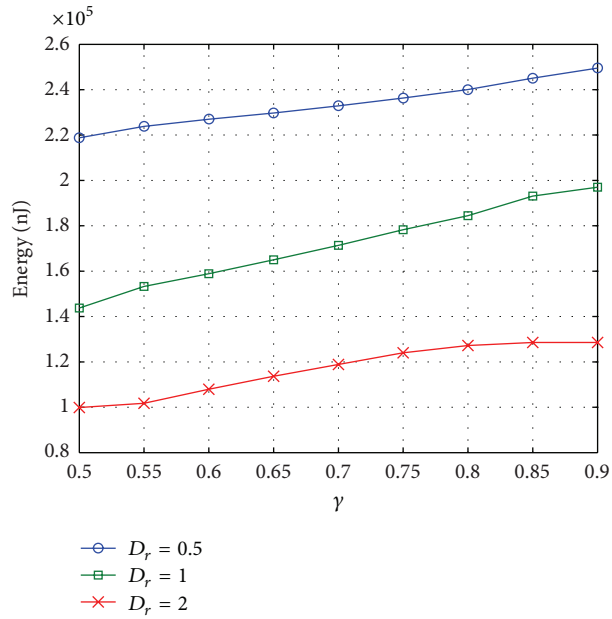


FIGURE 7: Energy consumption versus γ for $T_d = 100$ ms.

MBNA and DSEA routing have the same energy consumption. The reason is that when the MSE constraint is large, the Quasi-BLUE can be finished with few sensors' data. According to (17), $P(\text{EDD} < T_d)$ will increase when the size of S_r is small. When D_r is large enough, DSEA routing does not need to reduce the number of hops and MBNA will not add redundant transfer state node.

The probability γ affects the number of redundant transfer state nodes. We make $T_d = 100$ ms and compare the results of MBNA with different γ. We choose three value of γ and

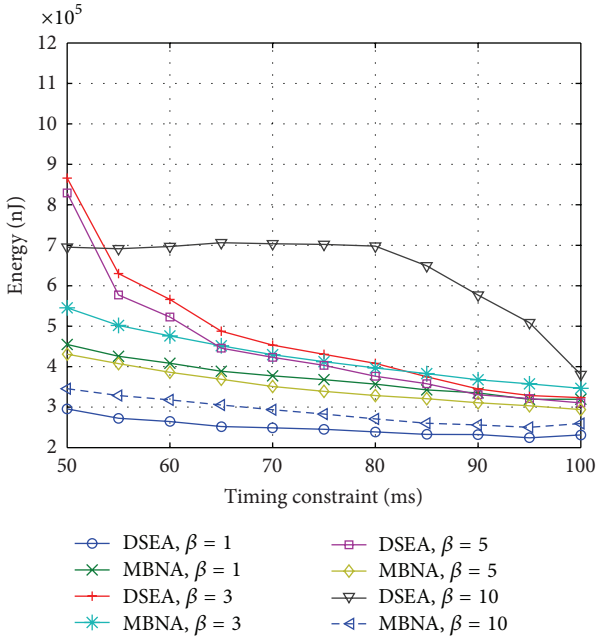


Figure 8: Energy consumption with normal distribution single hop delay.

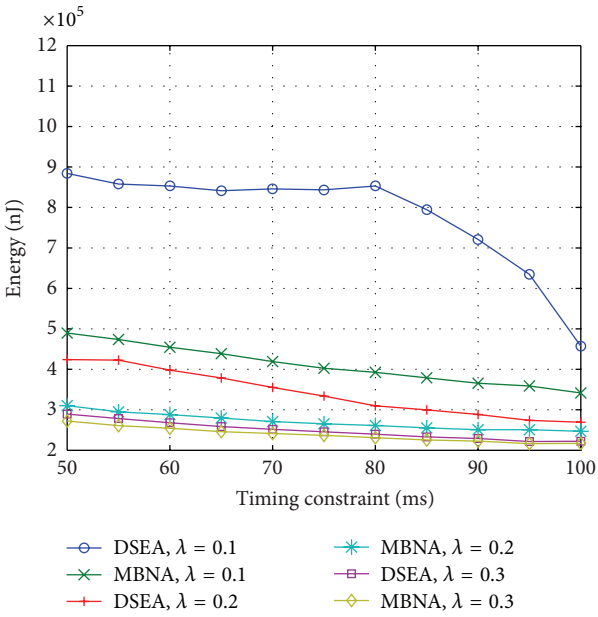


Figure 9: Energy consumption with negative exponential distribution single hop delay.

simulate our MBNA with different γ. For each γ, the simulations is repeated for 100 times. We record the average energy consumption for the Quasi-BLUE. The simulation results is shown in Figure 7. γ represents the probability that the estimation should be finished before the deadline. If γ is small, MBNA will not add many redundant nodes to guarantee the timing constraint. As a result, the energy consumption will decrease as γ decreases.

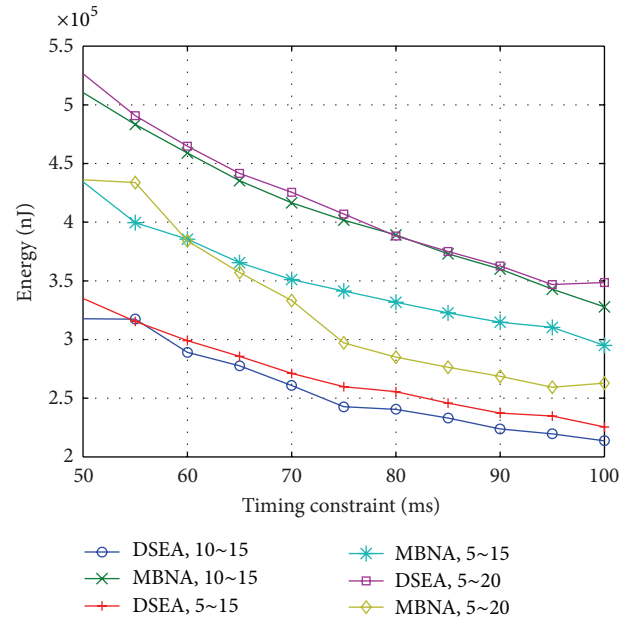


Figure 10: Energy consumption with uniform distribution single hop delay.

6.3. Simulation for Different Single Hop Delay Distributions. The single hop delay distribution will affect the performance of DSEA routing. In DSEA routing, the worst case of single hop delay is used in the route planning. If the worst case is not far away from the common case, DSEA routing may achieve lower energy consumption than MBNA. The general hypothesis of single hop delay distribution are normal distribution, negative exponential distribution and uniform distribution. We make $D_r = 1$, $\gamma = 0.8$ and simulate the performance of DSEA routing and MBNA under the three single hop delay distributions. The performances of DESA and MBNA with different single hop delay distributions are shown in Figures 8, 9 and 10. Figure 8 shows the energy consumption for normal distribution single hop delay, Figure 9 shows the energy consumption for negative exponential distribution single hop delay, and Figure 10 shows the energy consumption for uniform distribution single hop delay.

In the normal distribution case, the single hop distribution is assume to satisfy the $N(15, \beta^2)$, β^2 is the variance of the distribution. In Figure 8, when $\beta = 3, 5, 10$, MBNA achieves lower energy consumption than DSEA routing. When $\beta = 1$, DSEA routing performs better than MBNA. In the negative exponential distribution case, the CDF single hop distribution is assumed to be $1 - e^{-\lambda t}$. With different values of λ, the performances of DSEA routing and MBNA changes. When λ is small, MBNA shows great energy-efficiency over DSEA routing. When $\lambda = 0.3$, MBNA and DSEA routing have similar energy consumption. In the uniform distribution case, we let the single hop delay vary in a range. In Figure 10, the single hop delay varies in $[10, 15]$, $[5, 15]$, and $[5, 20]$, respectively. In the uniform distribution case, we find that MBNA shows no advantage when the single hop delay varies

in [10, 15] and [5, 15]. When the single hop delay varies in [5, 20], MBNA provides lower energy consumption.

The above results reflect the fact that DSEA routing guarantees the hard timing constraint. In DSEA routing, the packets travel along a path whose maximum end-to-end delay is less than the timing constraint. In general, the worst case will happen with a small probability, and DSEA routing over considers the end-to-end delay. Therefore, DSEA routing is not energy-efficient compared to MBNA. However, if the end-to-end packet delay varies in a small range, that is, the variance of delay distribution is small, the worst case of delay will not be far from the mean value of delay. In this case, the property of soft timing constraint is not notable, and DSEA routing may achieve lower energy consumption than MBNA. In the three single hop distribution cases, MBNA can reduce the energy consumption a lot when the variance is large. For the network traffic with large uncertainty, the single hop delay usually varies in a large range. And our MBNA can achieve lower energy consumption in this situation.

7. Conclusion

In this paper, we focused on the energy-efficient scheduling for soft real-time parameter estimation in WSNs. The estimator at the BS is Quasi-BLUE, which is a quite common estimator in WSNs. In order to save energy, not all the sensor nodes will send the data to the BS. Only part of sensor nodes will be at the transfer state so that the Quasi-BLUE can be implemented with an MSE constraint. In some real-time applications, we always expect the estimation can be finished before a deadline with a high probability. The EDD describes the time that the BS receives sufficient data from sensor nodes to implement the estimation. The traditional approaches usually try to reduce the number of hops to decrease the end-to-end packet delay, which will increase the communication energy. However, in the scenario of Quasi-BLUE, the BS just needs the data from an area instead of a unique sensor. A sensor node's data can be replaced by another sensors. Therefore, adding some redundant transfer state nodes will increase the probability that EDD is less than the timing constraint, that is, $P(\text{EDD} < T_d) > \gamma$.

Because a packet from a sensor node corresponds to a delay distribution, the calculation of $P(\text{EDD} < T_d) > \gamma$ with packets from different sensor nodes is difficult. In this paper, we proposed the MSEPF and employ the MSEPF to calculate $P(\text{EDD} < T_d) > \gamma$. The approach takes advantages of the linear property of Quasi-BLUE and it is easy to implement. Once $P(\text{EDD} < T_d) > \gamma$ is obtained, we proposed the MBNA algorithm to schedule the transfer state sensor nodes for the soft real-time Quasi-BLUE. We compared our MBNA with the existing DESA routing approaches in the soft real-time Quasi-BLUE. The simulation results show that MBNA is more energy-efficient while satisfying the soft timing constraint in the Quasi-BLUE.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61222310, 61174142, 61071061, 61134012, and 60874050, the Zhejiang Provincial Natural Science Foundation of China under Grants R1100234 and Z1090423, the Program for New Century Excellent Talents (NCET) in University under Grant NCET-10-0692, the Fundamental Research Funds for the Central Universities under Grant 2011QNA4036, the ASFC under Grant 20102076002, the Specialized Research Fund for the Doctoral Program of Higher Education of China (SRFDP) under Grants 20100101110055, and 20120101110115, the Zhejiang Provincial Science and Technology Planning Projects of China under Grants 2012C21044 and the Marine Interdisciplinary Research Guiding Funds for Zhejiang University under Grant 2012HY009B. This work was also supported by the "151 Talent Project" of Zhejiang Province. The work is also partially supported by NSF CNS-1249223 (M. Qiu).

References

[1] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[2] Z. Q. Luo, "Universal decentralized estimation in a bandwidth constrained sensor network," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 2210–2219, 2005.

[3] J. Li and G. AlRegib, "Network lifetime maximization for estimation in multihop wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2456–2466, 2009.

[4] J. Y. Wu, Q. Z. Huang, and T. S. Lee, "Energy-constrained decentralized best-linear-unbiased estimation via partial sensor noise variance knowledge," *IEEE Signal Processing Letters*, vol. 15, pp. 33–36, 2008.

[5] J. Xiao, A. Ribeiro, Z. Luo, and G. Giannakis, "Power scheduling of universal decentralized estimation in sensor networks," *IEEE Signal Processing Magazine*, vol. 54, no. 2, pp. 413–422, 2006.

[6] J. Li and G. AlRegib, "Distributed estimation in energy-constrained wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 3746–3758, 2009.

[7] M. Bhardwaj, T. Garnett, and A. P. Chandrakasan, "Upper bounds on the lifetime of sensor networks," in *Proceedings of the International Conference on Communications (ICC '01)*, pp. 785–790, June 2001.

[8] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[9] S. Cui, A. Goldsmith, and A. Bahai, "Joint modulation and multiple access optimization under energy constraints," in *Proceedings of the IEEE Global Telecommunications Conference*, pp. 151–155, December 2004.

[10] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2349–2360, 2005.

[11] C. Schurgers, V. Tsiatsis, and M. Srivastava, "Stem: topology management for energy efficient sensor networks," in *Proceedings of the Aerospace Conference Proceedings*, pp. 1099–1108, 2002.

[12] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, "Scheduling sleeping nodes in high density cluster-based sensor networks," *Mobile Networks and Applications*, vol. 10, no. 6, pp. 825–835, 2005.

[13] M. Qiu, C. Xue, Z. Shao, Q. Zhuge, M. Liu, and E. H. M. Sha, "Efficient algorithm of energy minimization for heterogeneous

wireless sensor network," in *Proceedings of the IEEE/IFIP International Conference on Embedded And Ubiquitous Computing*, pp. 25–34, 2006.

[14] M. Qiu, J. Liu, J. Li, Z. Fei, Z. Ming, and E. H.-M. Sha, "A novel energyaware fault tolerance mechanism for wireless sensor networks," in *Proceedings of the IEEE/ACM GreenCom*, pp. 56–61, 2011.

[15] M. Qiu, C. Xue, Z. Shao, M. Liu, and E. H. M. Sha, "Energy minimization for heterogeneous wireless sensor networks," *Journal of Embedded Computing*, vol. 3, no. 2, pp. 109–117, 2009.

[16] Y. Wang, M. C. Vuran, and S. Goddard, "Analysis of event detection delay in wireless sensor networks," in *Proceedings of the IEEE INFOCOM*, pp. 1296–1304, Shanghai, China, April 2011.

[17] Y. Wang, M. C. Vuran, and S. Goddard, "Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 305–3318, 2012.

[18] M. Xie and M. Haenggi, "Towards an end-to-end delay analysis of wireless multihop networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 849–861, 2009.

[19] T. F. Abdelzaher, S. Prabh, and R. Kiran, "On real-time capacity limits of multihop wireless sensor networks," in *Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS '04)*, pp. 359–370, December 2004.

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

[21] K. Akkaya and M. Younis, "Energy-aware delay-constrained routing in wireless sensor networks," *International Journal of Communication Systems*, vol. 17, no. 6, pp. 663–687, 2004.

[22] A. Pourkabirian and A. T. Haghighat, "Energy-aware, delay-constrained routing in wireless sensor networks through genetic algorithm," in *Proceedings of the 15th International Conference on Software, Telecommunications and Computer Networks (SoftCOM '07)*, pp. 1–5, Split-Dubrovnik, Hvar, September 2007.

[23] D. Pompili, T. Melodia, and I. F. Akyildiz, "Distributed routing algorithms for underwater acoustic sensor networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, pp. 2934–2944, 2010.

[24] S. C. Ergen and P. Varaiya, "Energy efficient routing with delay guarantee for sensor networks," *Wireless Networks*, vol. 13, no. 5, pp. 679–690, 2007.

[25] J. J. Xiao, S. Cui, Z. Q. Luo, and A. J. Goldsmith, "Joint estimation in sensor networks under energy constraints," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON '04)*, pp. 264–271, October 2004.

[26] J. B. Schmitt, F. A. Zdarsky, and L. Thiele, "A comprehensive worst-case calculus for wireless sensor networks with in-network processing," in *Proceedings of the 28th IEEE International Real-Time Systems Symposium (RTSS '07)*, pp. 193–202, Tucson, Ariz, USA, December 2007.

[27] A. Burchard, J. Liebeherr, and S. D. Patek, "A min-plus calculus for end-to-end statistical service guarantees," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4105–4114, 2006.

[28] A. Koubaa, M. Alves, and E. Tovar, "Modeling and worst-case dimensioning of cluster-tree wireless sensor networks," in *Proceedings of the 27th IEEE International Real-Time Systems Symposium (RTSS '06)*, pp. 412–421, Rio de Janeiro, Brazil, December 2006.

[29] R. S. Oliver and G. Fohler, "Probabilistic estimation of end-to-end path latency in wireless sensor networks," in *Proceedings of the IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (MASS '09)*, pp. 423–431, Macau, China, October 2009.

[30] M. Qiu and E. H. M. Sha, "Cost minimization while satisfying hard/soft timing constraints for heterogeneous embedded systems," *ACM Transactions on Design Automation of Electronic Systems*, vol. 14, no. 2, article 25, 2009.

[31] J. Niu, Y. Gao, M. Qiu, and Z. Ming, "Selecting proper wireless network interfaces for user experience enhancement with guaranteed probability," *Journal of Parallel and Distributed Systems*, vol. 72, no. 12, pp. 1565–1575, 2012.

[32] V. C. Gungor, O. B. Akan, and I. F. Akyildiz, "A real-time and reliable transport $(RT)^2$ protocol for wireless sensor and actor networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 359–370, 2008.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 181426, 14 pages
http://dx.doi.org/10.1155/2013/181426

Research Article

Adaptive Computing Resource Allocation for Mobile Cloud Computing

Hongbin Liang,[1,2] Tianyi Xing,[3] Lin X. Cai,[4] Dijiang Huang,[3] Daiyuan Peng,[5] and Yan Liu[6]

[1] *State Key Laboratory of Information Security, Institute of Information Engineering, The Chinese Academy of Sciences, Beijing 100093, China*
[2] *School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, Sichuan 610031, China*
[3] *Arizona State University, 699 S Mill Avenue, Suite 464, Tempe, AZ 85281, USA*
[4] *Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1*
[5] *School of Information Science and Technology, Southwest Jiaotong University, Chengdu, Sichuan 610031, China*
[6] *School of Software and Microelectronics, Peking University, Beijing 102600, China*

Correspondence should be addressed to Yan Liu; ly@ss.pku.edu.cn

Received 6 January 2013; Accepted 24 February 2013

Academic Editor: Jianwei Niu

Mobile cloud computing (MCC) enables mobile devices to outsource their computing, storage and other tasks onto the cloud to achieve more capacities and higher performance. One of the most critical research issues is how the cloud can efficiently handle the possible overwhelming requests from mobile users when the cloud resource is limited. In this paper, a novel MCC adaptive resource allocation model is proposed to achieve the optimal resource allocation in terms of the maximal overall system reward by considering both cloud and mobile devices. To achieve this goal, we model the adaptive resource allocation as a semi-Markov decision process (SMDP) to capture the dynamic arrivals and departures of resource requests. Extensive simulations are conducted to demonstrate that our proposed model can achieve higher system reward and lower service blocking probability compared to traditional approaches based on greedy resource allocation algorithm. Performance comparisons with various MCC resource allocation schemes are also provided.

1. Introduction

Cloud computing is a new computing service model with characteristics such as resource on demand, pay as you go, and utility computing [1]. It provides new computing models for both service providers and individual customers, which can be broadly classified into infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Furthermore, smart phones are expected to overtake PCs and become the most common web access entities worldwide by 2013 as predicted by Gartner [2]. Since mobile devices (MDs) have more advantages such as mobility, flexibility, and sensing capabilities over fixed terminals, integrating mobile computing and cloud computing techniques is a natural and predictable approach to build new mobile applications, which has attracted a lot of attention in both academia and industry community. As a result, a new research field, called mobile cloud computing (MCC), is emerging.

In [3], Huang et al. presented a new MCC infrastructure, called MobiCloud, where dedicated virtual machines (VMs) are assigned to mobile users to improve the security and privacy capability. In such an MCC environment, the system computational resources, such as CPU, storage, and memory, are partitioned into several service provisioning domains based on the cluster geographical distribution. Each domain consists of multiple VMs, and each VM handles parts of cloud computing resource (i.e., CPU, storage and memory, etc.). When the MCC service provisioning domain receives a service request from a mobile device, it needs to make a decision on (1) whether to accept the request; and (2) how much Cloud resources should be allocated if the request is accepted. Although the Cloud resource can be considered

as unlimited compared with the computing resource in a single mobile device, in practice, a geographically distributed cloud system usually contains limited resource at a local service provisioning domain. When all the Cloud resources are occupied within the local service provisioning domain, the service request from mobile device will be rejected (or migrated to a nonlocal service provisioning domain) due to the resource unavailability. The rejection of a service request not only degrades the user satisfaction level (i.e., resulting in a long service delay due to the nonlocal service provisioning or service migration to other remote domain), but also reduces the system reward which is usually defined as a metric that includes the system net income and cost.

In general, the Cloud income increases with the number of the accepted services. However, it is definitely not true that cloud service provider (CSP) would like to acccept service requests as many as possible, since more accepted services occupy more cloud resources, and more likely a new request will be rejected when the network resource is limited, which degrades the QoS level of users. The rewards of the most existing Cloud resource allocation methods only consider the income on behalf of the CSP. To obtain a comprehensive system reward of MCC, the customer QoS and user satisfaction level should be taken into account in the system reward as well. Therefore, our research goal is to address the following questions: how to obtain the maximal overall system rewards by taking into account from both the service provider side and the customer side while satisfying a certain QoS level.

In this paper, we present an adaptive MCC resource allocation model based on semi-Markov decision process (SMDP) to achieve the objective mentioned above. Our proposed MCC model considers not only the incomes of accepting services, but also the cost resulted from VM occupation in the Cloud. Moreover, other factors including service precessing time of both Cloud and MD battery consumption of mobile device are also taken into account. Thus, the overall economic gain is determined by a comprehensive approach which considers all the factors mentioned above.

The contributions and essence of this proposed model are listed as follows.

(i) Semi-Markov decision process (SMDP) is applied to derive the optimal resource allocation policy for MCC.

(ii) The proposed model allows adaptive resource allocations, that is, multiple Cloud resources (i.e., the number of VMs) can be allocated to a service request based on the available Cloud resource in the service domain in order to maximize the resource utilization and enhance the user experience.

(iii) The maximal system rewards of Cloud can be achieved by using the proposed model and by taking into the considerations the expenses and incomes of both Cloud and mobile devices.

The rest of this paper is organized as follows. We present the related work in Section 2. In Section 3, the basic system model is described. The semi-Markov decision process model for MCC system is presented in Section 4. Based on our proposed model, we analyze the probabilities for each adaptive allocation scheme and rejection probability in Section 5. We evaluate the performance of the proposed economic model in Section 6 and conclude this paper and discuss the future work in Section 7.

2. Related Work

Recent research work for Cloud computing has shifted its focus from the Cloud for fixed user to Cloud for mobile devices [4], which enables a new model of running applications between resource-constrained devices and Internet-based Cloud. Moreover, resource-constrained mobile devices can outsource computation/communication/storage intensive tasks onto the Cloud. CloneCloud [5] focuses on execution augmentation with less consideration on user preference or device status. Elastic applications for mobile devices via Cloud computing were studied in [6]. In [3], Huang et al. presented an MCC model that allows the mobile device related operations residing either on mobile devices or dedicated VMs in the Cloud. [7] proposes a way using traffic-aware virtual machine (VM) placement to improve the network scalability by optimizing the placement of VMs on host machines.

Although resource management in wireless networks has been extensively studied [8–10], there are few previous works focusing on resource management of Cloud computing and especially mobile cloud computing. In [11], an economic mobile cloud computing model is presented to decide how to manage the computing tasks with a given configuration of the Cloud system. That is, the computing tasks can be migrated between the mobile devices and the Cloud servers. A game theory-based resource allocation model to allocate the Cloud resources according to users' QoS requirements is proposed in [12]. In the past few years, some research work focused on application of specific resource management in Cloud computing using virtual machines or end servers in data center. In [13], authors propose a new operating system which enables resource-aware programming while permitting high-level reusable resource management policies for context-aware applications in Cloud computing. Lorincz et al. [14] address the problem of resource management in semantic event processing applications in Cloud computing. Tesauro et al. [15] propose a reinforcement learning based management system for dynamic allocation of servers trying to maximize the profit of the host data center in Cloud computing. In [16], Boloor et al. propose a generic request allocation and scheduling scheme to achieve desired percentile service level agreements (SLA) goals of consumers and to increase the profits to the cloud provider.

The works discussed above target to achieve a higher Cloud system profit and/or to meet a better service level agreement (SLA). However, they model the problem from service provider's perspective without considering the costs and profits of mobile devices. Therefore, the overall system rewards derived in previous works are sufficient. Generally, a Cloud-based application can be assigned with

multiple resources in terms of VMs (can be in different domains/clusters) to obtain more computation/storage and other capacities. However, to our best knowledge, in the previous literature, none of them addresses the following emerging research problems: (1) how to construct a reward model of MCC system for resources allocation purpose by considering the rewards from both Cloud system and mobile users; (2) how to allocate system resources to service requests to maximize the user satisfaction level of mobile users while obtaining the maximal overall system and user rewards under a given QoS level.

3. System Model

A major benefit of MCC over the traditional client-server mode is that MDs can have more capabilities and better performance (i.e., less processing time, energy saving, etc.) when they outsource their tasks onto the Cloud. The outsourcing procedure can be implemented by using weblets (application components) to link the services between the Cloud and the mobile devices. A weblet can be platform independent such as using Java or .Net bytecode or Python script or platform dependent, using a native code. Some research work [5] focuses on the algorithm to decide whether to offload the weblet from MD to the Cloud (i.e., run on one or more virtual nodes offered by an IaaS provider) or run the weblet on the MD itself. In this way, a mobile device can dynamically expand its capabilities, including computation power, storage capacity, and network bandwidth, by offloading an elastic application service to the Cloud. The choice made by mobile device on whether to offload the task onto the Cloud can refer to the mobile device's status such as CPU processing capability, battery power level, and network connection quality and security. In this paper, the service scenario of the proposed model is the task offloading from MD onto the Cloud. Also, the task offloading procedure can be done in a way that MD sends a service request to the Cloud firstly, then the task is further offloaded to the Cloud once the service request is accepted by the Cloud.

As shown in Figure 1, a VM is responsible for managing the weblet's loading, unloading, and processing in the mobile Cloud. Each VM has the capacity to hold one weblet at a time for handling migrated weblet request, and two types of service requests are defined to be handled by a VM: (i) *paid*: a paid weblet service request is sent to the service provisioning domain from a mobile device; (ii) *free*: a free weblet service request is sent to the service provisioning domain from a mobile device. Figure 1 demonstrates the relationship between the *paid*/*free* service requests and the VMs of the service provisioning domain.

In this paper, the MCC service architecture is based on the MobiCloud framework presented in [3], in which a VM can handle a portion of Cloud system resources (CPU, memory and storage, etc.) that can satisfy the minimal resource requirement to process an application offloading service in the MCC system. Within the local MobiCloud service provisioning domain, the resource capacity, in terms of the number of VMs, is limited. Thus, if the demands of the arriving service requests exceed the number of available VM resources in a certain service domain, the following service requests will be rejected (or migrated to a remote service provisioning domain). On the other hand, if the demands of the arriving service requests are lower than the number of the available VMs, more VMs can be assigned to one service request to maximally utilize the Cloud resource and achieve a better performance and QoS. Our analytical model is based on a single local service domain. The analysis of local service migrations to remote service domains is regarded as the future study.

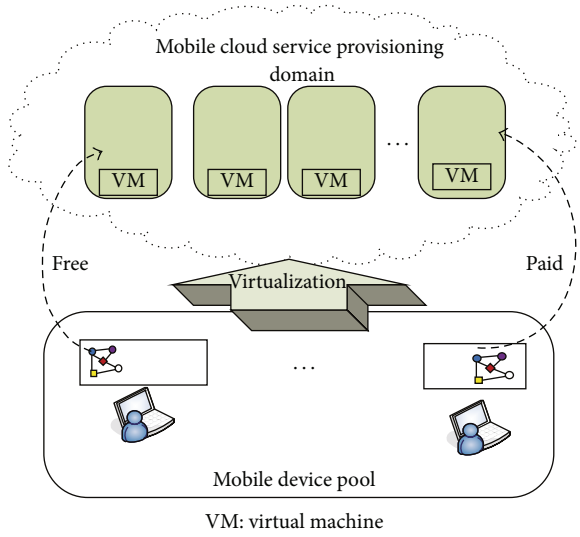


Figure 1: Reference model of mobile cloud computing.

3.1. System Description. An MCC system mainly consists of two entities, VM and physical MD. A VM is the minimum set of resources that can be allocated to an MD upon receiving its service request. Since an MD is a wireless node with limited computing capability and energy supply, it can outsource its mobile codes (i.e., weblet) of an application service to the Cloud. Then, the Cloud will decide a number of VMs to be allocated to the arriving service request if the decision for the service request made by the Cloud is accepted.

In this paper, we consider a service provisioning domain with K VMs. The maximum number of VMs that can be allocated to a Cloud service is c VMs (we denote as c allocation scheme), where $c \in \{1, 2, \ldots, C\}, C \leq K$. Generally, the duration for running a mobile application service in the Cloud depends on the number of VMs allocated to that service. The relationship between the processing time of an application service and the number of allocated VMs in the Cloud can be expressed as a function denoted as $\xi(c)$. Assume that the time to process an application service by using one VM in a service provisioning domain is θ_s, therefore the time to handle the service is $\xi(c)\theta_s$ if c VMs are allocated to that service. The higher computing speed for an application service in a service provisioning domain means the higher user satisfaction level, which is the major part of the whole system reward of the Cloud. Thus, in order to improve the whole system reward of a service provisioning domain by

increasing the user satisfaction level, the traditional greedy algorithm [17] always decides to allocate maximal VMs to the service. But on the other hand, if the Cloud computing resources (denoted by the number of VM) allocated to the current service by the service provisioning domain are too high, then the following several arrival service requests may be rejected by the service provisioning domain because of insufficient available Cloud computing resources, which decease the user satisfaction level. As a result, the system rewards of that MCC service provisioning domain degrade as well.

It can be more complicated when we consider both the rewards and costs of mobile devices. Cost involved in the MD side should not be neglected, which means that the whole system reward should consider not only the rewards of the mobile Cloud itself, but also the incomes and the costs of MD, such as the saved battery energy if the service is processed in the mobile Cloud and the expense of the battery energy and the processing time of MD if the application service is processed on the MD locally.

To model this complex dynamic MCC resource allocation process, without loss of generality, we assume that the arrival rates of both *paid* and *free* service requests follow Poisson distributions with mean rate of λ_p and λ_f, respectively. The life time of services follows exponential distributions. The mean holding time of a service which is allocated only one VM in the service provisioning domain is $1/\mu$. Thus, the holding time of the service allocated c VMs in the domain is $\xi(c)/\mu$, which implies that the mean departure rate of finished service is $\mu/\xi(c)$.

Since the decision making epoch is randomly generated in the system, we use semi-Markov decision process (SMDP) to model the dynamic MCC resource allocation process based on the system description we presented above. SMDP is a stochastic dynamic programming method, which can be used to model and solve optimal dynamic decision making problems. There are six following elements in the SMDP model: (a) *system states*; (b) *action sets*; (c) *the events that cause the decisions*; (d) *decision epoches*; (e) *transition probabilities*; and (f) *reward*. In the following, we first present the system states, the actions, the events, and the reward model for the MCC system.

3.2. System States. According to the assumption, there are total K VMs in one service provisioning domain, and c VM can be allocated to the service request, which is from 1 to C, where $C \leq K$. However, the arrival of *paid* application service request and *free* application service request and the departure of the finished service are distinct events. Thus, the system states can be described by the number of the running Cloud services which occupy the same number of VMs and the events (including both arrival and departure events) in the service provisioning domain. Here, we use c to indicate the number of VMs allocated to one application service (denoted as c allocation scheme as presented in Section 3.1), $c \in \{1, 2, \ldots, C\}$. Therefore, the number of the running Cloud services which occupy c VMs in one service provisioning domain can be denoted as s_c.

In the MCC system model, we can define two types of service events: (1) a *paid* or *free* service request arrives from an MD, denoted by A_p and A_f, respectively; and (2) the departure of a finished application service occupying c VMs in the current service provisioning domain, denoted by F_c. Thus, the event e in the MCC system can be described as $e \in \{A_p, A_f, F_1, F_2, \ldots, F_C\}$. Therefore, the system state can be expressed as

$$S = \{s \mid s = \langle s_1, s_2, \ldots, s_C, e\rangle\}, \tag{1}$$

where $\sum_{c=1}^{C}(s_c * c) \leq K$.

3.3. Actions. For a system state of the service provisioning domain with an incoming service request from an MD (i.e., A_p or A_f), the mobile Cloud needs to make a decision on whether to accept the service request and what is the allocation scheme (i.e., how many VMs to allocate to the MD) if the decision is acceptance. If the decision is acceptance, then the c allocation scheme is assigned to the arriving service request; thus, the action to assign the c allocation scheme can be denoted as $a(s) = c$. While if the decision is rejection based on the whole system reward, which means no VM will be assigned, thus the *paid* or *free* service request will be rejected and the application will run on the MD itself. Then, the action to reject the service request can be denoted as $a(s) = 0$.

And for the departure of a finished service in the service provisioning domain (i.e., $e = F_c$), the action for this event can be considered as to calculate the current available Cloud resources and denoted as $a(s) = -1$. Therefore, the action space can be defined as $a(s) \subseteq \mathrm{Act}_s$, where

$$a(s) = \begin{cases} \{0, 1, \ldots, C\}, & e \in \{A_p, A_f\}, \\ -1, & e \in \{F_1, F_2, \ldots, F_C\}. \end{cases} \tag{2}$$

3.4. Reward Model. Based on the system state and its corresponding action, we can evaluate the whole mobile Cloud system reward (denoted by $r(s, a)$), which is computed based on the income and the cost as follows:

$$r(s, a) = w(s, a) - g(s, a), \quad e \in \{A_p, A_f, F_1, F_2, \ldots, F_C\}, \tag{3}$$

where $w(s, a)$ is the net lump sum income for the Cloud and MDs and $g(s, a)$ denotes the system cost.

The net lump sum income should consider the payment from MD to the mobile Cloud, the saved battery energy of MD, and the consumed time of mobile Cloud to process the service if the service is run in the mobile Cloud, the consumed battery energy, and the consumed time of MD if the service is run on MD locally.

Thus, the net lump sum income $w(s, a)$ is computed as

$$w(s, a) = \begin{cases} 0, & a(s) = -1, e \in \{F_1, F_2, \ldots, F_C\} \\ -\gamma_d U_d - \theta_d \beta, & a(s) = 0, e \in \{A_p, A_f\} \\ E_d - \delta_d \beta - \xi(c)\theta_s \beta, & a(s) = c, e = A_p \\ -\delta_d \beta - \xi(c)\theta_s \beta, & a(s) = c, e = A_f. \end{cases} \tag{4}$$

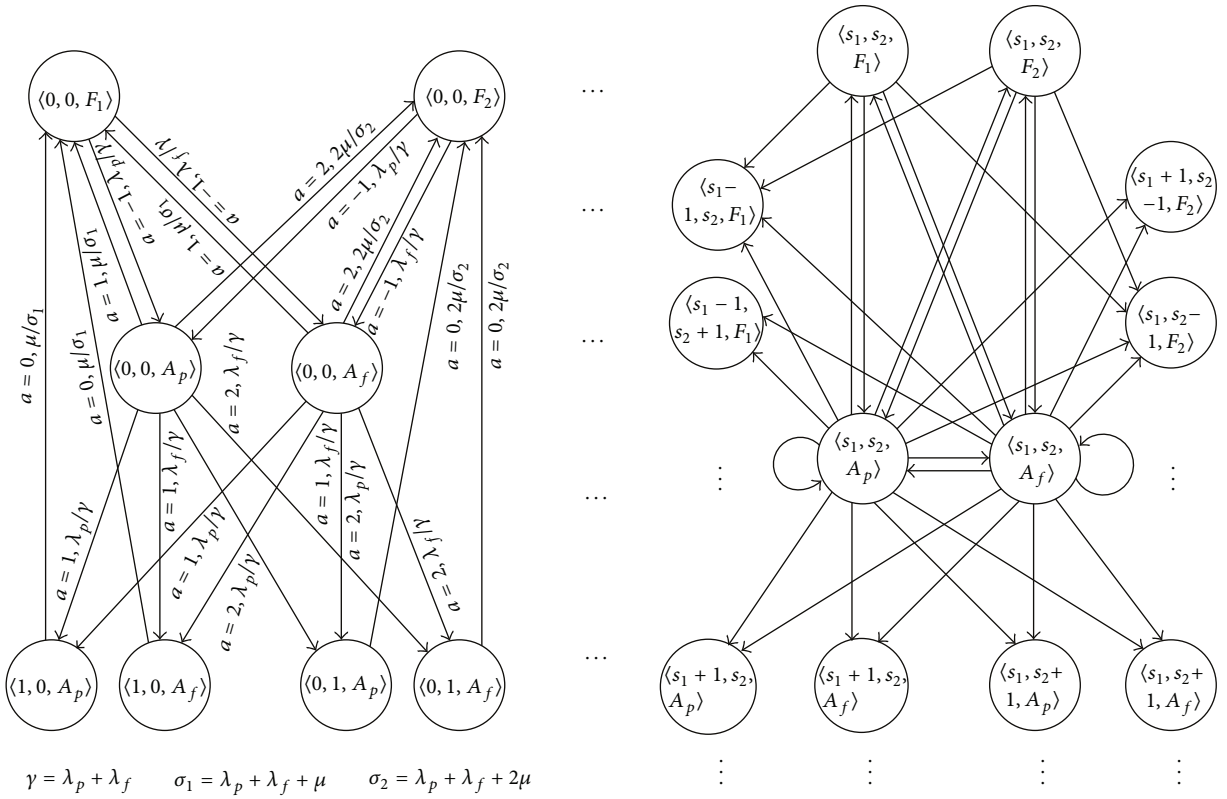


FIGURE 2: An example of state transition probabilities for two allocation schemes. The first item represents the action and the second item represents the state transition probability.

In (4), E_d is the income of the service provisioning domain obtained from the MD when it accepts a *paid* service request from the MD. δ_d denotes the time consumed on transmitting the service request from MD to the service provisioning domain through wireless connection, while β denotes the price per unit time, which has the same measurement unit as the income. Thus, $\delta_d\beta$ denotes the expense measured by the time consumed on transmitting the service request from MD to the service provisioning domain. U_d represents the expense measured by the battery energy consumed by the MD when the service request is rejected by the service provisioning domain and run on the MD locally, which has the same measurement unit as the income. γ_d is the weight factor that satisfies $0 \le \gamma_d \le 1$. Let θ_d denote the time to process an application service by using one mobile device, then $\theta_d\beta$ represents the expense measured by the time consumed to process the application using one mobile device. Similarly, $\theta_s\beta$ denotes the expense measured by the time consumed to process the service using one VM in a service provisioning domain. Therefore, $\xi(c)\theta_s\beta$ denotes the expense measured by the time consumed to process the service using c VMs in a service provisioning domain.

In (3), $g(s, a)$ is given by

$$g(s,a) = \tau(s,a)\, o(s,a), \quad a(s) \in \mathrm{Act}_s. \tag{5}$$

In (5), $\tau(s, a)$ is the average expected service time when the system state transfers from current state s to the next potential state j and the decision a is made; $o(s, a)$ is the cost rate of the service time and it is defined as the number of all occupied VMs; thus, it can be computed as

$$o(s,a) = \sum_{c=1}^{C} \left(s_c * c\right). \tag{6}$$

4. SMDP-Based Mobile Computing Model

Based on the SMDP model, we have already defined the *system states*, *action sets,* the *events,* and *reward* for the MCC system in the last section, then we need to define the *decision epoches* and obtain the *transition probabilities* to calculate the maximum long-term whole system reward.

There are three types of events in the MCC system (i.e., an arrival of a *paid* service request, an arrival of a *free* service request, and a departure of a finished service). The next decision epoch occurs when any of the three types of events takes place. Based on our assumption, the arrival of service request follows Poisson distribution and the departure of finished service follows exponential distribution. Thus, the expected time duration between two decision epoches (i.e., $\tau(s, a)$) follows exponential distribution as well. Then,

Table 1: States transition probabilities of system model at $C = 2$. $\gamma = \lambda_p + \lambda_f + s_1\mu + 2s_2\mu$, $\sigma_1 = \lambda_p + \lambda_f + (s_1 + 1)\mu + 2s_2\mu$, $\sigma_2 = \lambda_p + \lambda_f + s_1\mu + 2(s_2 + 1)\mu$.

Current state	Next state	Action (a)	Transition probability
$\langle s_1, s_2, A_p\rangle$	$\langle s_1, s_2, A_p\rangle$	0, −2	λ_p/γ,
	$\langle s_1, s_2, A_f\rangle$	0, −2	λ_f/γ,
	$\langle s_1 - 1, s_2, F_1\rangle$	0, −2	$s_1\mu/\gamma$,
	$\langle s_1 - 1, s_2 + 1, F_1\rangle$	2	$s_1\mu/\sigma_2$,
	$\langle s_1, s_2, F_1\rangle$	1	$(s_1 + 1)\mu/\sigma_1$,
	$\langle s_1, s_2, F_2\rangle$	2	$2(s_2 + 1)\mu/\sigma_2$,
	$\langle s_1 + 1, s_2 - 1, F_2\rangle$	1	$2s_2\mu/\sigma_1$,
	$\langle s_1, s_2 - 1, F_2\rangle$	0, −2	$2s_2\mu/\gamma$,
	$\langle s_1 + 1, s_2, A_p\rangle$	1	λ_p/σ_1,
	$\langle s_1 + 1, s_2, A_f\rangle$	1	λ_f/σ_1,
	$\langle s_1, s_2 + 1, A_p\rangle$	2	λ_p/σ_2,
	$\langle s_1, s_2 + 1, A_f\rangle$	2	λ_f/σ_2,
$\langle s_1, s_2, A_f\rangle$	$\langle s_1, s_2, A_p\rangle$	0	λ_p/γ,
	$\langle s_1, s_2, A_f\rangle$	0	λ_f/γ,
	$\langle s_1 - 1, s_2, F_1\rangle$	0	$s_1\mu/\gamma$,
	$\langle s_1 - 1, s_2 + 1, F_1\rangle$	2	$s_1\mu/\sigma_2$,
	$\langle s_1, s_2, F_1\rangle$	1	$(s_1 + 1)\mu/\sigma_1$,
	$\langle s_1, s_2, F_2\rangle$	2	$2(s_2 + 1)\mu/\sigma_2$,
	$\langle s_1 + 1, s_2 - 1, F_2\rangle$	1	$2s_2\mu/\sigma_1$,
	$\langle s_1, s_2 - 1, F_2\rangle$	0	$2s_2\mu/\gamma$,
	$\langle s_1 + 1, s_2, A_p\rangle$	1	λ_p/σ_1,
	$\langle s_1 + 1, s_2, A_f\rangle$	1	λ_f/σ_1,
	$\langle s_1, s_2 + 1, A_p\rangle$	2	λ_p/σ_2,
	$\langle s_1, s_2 + 1, A_f\rangle$	2	λ_f/σ_2,
$\langle s_1, s_2, F_1\rangle$	$\langle s_1 - 1, s_2, F_1\rangle$	−1	$s_1\mu/\gamma$,
$\langle s_1, s_2, F_2\rangle$	$\langle s_1, s_2, A_p\rangle$	−1	λ_p/γ,
	$\langle s_1, s_2, A_f\rangle$	−1	λ_f/γ,
	$\langle s_1, s_2 - 1, F_2\rangle$	−1	$2s_2\mu/\gamma$.

the mean rate (denoted as $\gamma(s,a)$) of expected time can be represented as

$$\gamma(s,a) = \tau(s,a)^{-1} = \begin{cases} \lambda_p + \lambda_f + \sum_{c=1}^{C} \dfrac{s_c\mu}{\xi(c)}, & e \subseteq \{F_1, F_2, \ldots, F_C\} \\ & \text{or } e \subseteq \{A_p, A_f\},\ a = 0, \\ \lambda_p + \lambda_f + \sum_{c=1}^{C} \dfrac{s_c\mu}{\xi(c)} + \dfrac{\mu}{\xi(c)}, & e \subseteq \{A_p, A_f\},\ a = c. \end{cases} \tag{7}$$

Thus, the expected discounted reward (denoted as $r(s,a)$) during $\tau(s,a)$ can be obtained based on the discounted reward model defined in [18, 19],

$$\begin{aligned} r(s,a) &= w(s,a) - o(s,a) E_s^a \left\{ \int_0^\tau e^{-\alpha t} dt \right\} \\ &= w(s,a) - o(s,a) E_s^a \left\{ \frac{[1 - e^{-\alpha\tau}]}{\alpha} \right\} \\ &= w(s,a) - \frac{o(s,a)}{\alpha + \gamma(s,a)}, \end{aligned} \tag{8}$$

where α is a continuous-time discounting factor and $w(s,a)$, $o(s,a)$, and $\gamma(s,a)$ are defined in (4), (6), and (7), respectively.

Then the only element left to be calculated is the transition probabilities. To calculate the transition probabilities, we show an example in Figure 2.

In this example, without loss of generality, we assume that there are only two allocation schemes, which means $C = 2$. Thus, the transition probabilities in this example can be obtained in Table 1.

From the example, the transition probabilities of C allocation schemes can be deduced. Let $q(j \mid s,a)$ denote the state transition probability from the current state s to the next state j when action a is chosen. Then, the transition probability $q(j \mid s,a)$ can be expressed as following.

For the state $s = \langle s_1, s_2, \ldots, s_c, \ldots, s_C, A_p\rangle$, $q(j \mid s,a)$ can be obtained as

$$q(j \mid s,a) = \begin{cases} \dfrac{\lambda_p}{\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_C, A_p\rangle,\ a = 0 \\ \dfrac{\lambda_f}{\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_C, A_f\rangle,\ a = 0 \\ \dfrac{s_c\mu}{\varepsilon(c)\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_c - 1, \ldots, s_C, F_c\rangle, \\ \dfrac{(s_c + 1)\mu}{\varepsilon(c)\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_c, \ldots, s_C, F_c\rangle,\ a = c \\ \dfrac{s_m\mu}{\varepsilon(m)\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_m - 1, \ldots, s_c + 1, \ldots, \\ & \qquad S_C, F_m\rangle,\ s_m \geq 1,\ m \neq c,\ a = c \\ \dfrac{\lambda_p}{\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_c + 1, \ldots, s_C, A_p\rangle, \\ & \qquad s_c \leq C - 1,\ a = c \\ \dfrac{\lambda_f}{\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_c + 1, \ldots, s_C, A_f\rangle, \\ & \qquad s_c \leq C - 1,\ a = c, \end{cases} \tag{9}$$

where $c \subseteq \{1, 2, \ldots, C\}$, $m \subseteq \{1, 2, \ldots, C\}$, $m \neq c$.

For the states $s = \langle s_1, s_2, \ldots, s_c, \ldots, s_C, A_f \rangle$, $q(j \mid s, a)$ can be obtained

$$q(j \mid s,a) = \begin{cases} \dfrac{\lambda_p}{\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_C, A_p \rangle,\ a = 0 \\ \dfrac{\lambda_f}{\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_C, A_f \rangle,\ a = 0 \\ \dfrac{s_c \mu}{\varepsilon(c)\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_c - 1, \ldots, s_C, F_c \rangle, \\ & s_c \geq 1,\ a = 0 \\ \dfrac{(s_c+1)\mu}{\varepsilon(c)\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_c, \ldots, s_C, F_c \rangle, \\ & a = c \\ \dfrac{s_m \mu}{\varepsilon(m)\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_m - 1, \ldots, \\ & s_c + 1, \ldots, s_C, F_m \rangle, \\ & s_m \geq 1,\ m \neq c,\ a = c \\ \dfrac{\lambda_p}{\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_c + 1, \ldots, s_C, A_p \rangle, \\ & s_c \leq C - 1,\ a = c \\ \dfrac{\lambda_f}{\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_c + 1, \ldots, s_C, A_f \rangle, \\ & s_c \leq C - 1,\ a = c, \end{cases} \tag{10}$$

where $c \subseteq \{1, 2, \ldots, C\}$, $m \subseteq \{1, 2, \ldots, C\}$, and $m \neq c$.

For the states $s = \langle s_1, s_2, \ldots, s_c, \ldots, s_C, F_c \rangle$, the action for this departure state is always -1 which means $a = -1$, then the transition probability $q(j \mid s, a)$ can be obtained as

$$q(j \mid s,a) = \begin{cases} \dfrac{\lambda_p}{\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_C, A_p \rangle \\ \dfrac{\lambda_f}{\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_C, A_f \rangle \\ \dfrac{s_c \mu}{\xi(c)\gamma(s,a)}, & j = \langle s_1, s_2, \ldots, s_c - 1, \\ & \ldots, s_C, F_c \rangle,\ s_c \geq 1, \end{cases} \tag{11}$$

where $c \subseteq \{1, 2, \ldots, C\}$.

Then, the maximal long-term discounted reward is obtained based on the discounted reward model defined in [18, 19] and can be denoted as

$$v(s) = \max_{a \in \mathrm{Act}_s} \left\{ r(s,a) + \lambda \sum_{j \in S} q(j \mid s,a)\, v(j) \right\}, \tag{12}$$

where $\lambda = (\gamma(s,a))/(\alpha + \gamma(s,a))$, and $r(s,a)$ and $q(j \mid s,a)$ can be obtained in (8), (9), (10), and (11).

In the reward equation (8), the first part is that the revenue is a lump earnings of the reward and the second part is that the cost is a continuous-time payment of the reward. Thus, the reward function needs to be uniformized to obtain the uniformized long-term reward, then the discrete-time discounted Markov decision process can be used in this model. Based on the *assumption 11.5.1* in [19], we need to find a constant ω satisfying $[1 - q(s \mid s,a)]\gamma(s,a) \leq \omega < \infty$ to obtain the uniformized long-term reward by utilizing *(11.5.8)* in [19]. Let $\omega = \lambda_f + \lambda_p + K * C * \mu$ and $\overline{q}(j \mid s,a)$, $\overline{v}(s)$, $\overline{\gamma}(s,a)$ denote the uniformized transition probability, the long-term reward, and the reward function, respectively.

Thus, the transition probability can be uniformized as

$$\overline{q}(j \mid s,a) = \begin{cases} 1 - \dfrac{[1 - q(s \mid s,a)]\gamma(s,a)}{\omega}, & j = s \\ \dfrac{q(j \mid s,a)\gamma(s,a)}{\omega}, & j \neq s. \end{cases} \tag{13}$$

For the state $s = \langle s_1, s_2, \ldots, s_c, \ldots, s_C, A_p \rangle$, the uniformized transition probability $\overline{q}(j \mid s,a)$ is rewritten as

$$\overline{q}(j \mid s,a) = \begin{cases} \dfrac{(\omega + \lambda_p - \gamma(s,a))}{\omega}, & j = \langle s_1, s_2, \ldots, s_C, A_p \rangle,\ a = 0 \\ \dfrac{\lambda_f}{\omega}, & j = \langle s_1, s_2, \ldots, s_C, A_f \rangle,\ a = 0 \\ \dfrac{s_c \mu}{\xi(c)\omega}, & j = \langle s_1, s_2, \ldots, s_c - 1, \ldots, s_C, F_c \rangle, \\ & s_c \geq 1,\ a = 0 \\ \dfrac{(s_c+1)\mu}{\xi(c)\omega}, & j = \langle s_1, s_2, \ldots, s_c, \ldots, s_C, F_c \rangle,\ a = c \\ \dfrac{s_m \mu}{\xi(m)\omega}, & j = \langle s_1, s_2, \ldots, s_m - 1, \ldots, \\ & s_c + 1, \ldots, s_C, F_m \rangle,\ s_m \geq 1, \\ & m \neq c,\ a = c \\ \dfrac{\lambda_p}{\omega}, & j = \langle s_1, s_2, \ldots, s_c + 1, \ldots, s_C, A_p \rangle, \\ & s_c \leq C - 1,\ a = c \\ \dfrac{\lambda_f}{\omega}, & j = \langle s_1, s_2, \ldots, s_c + 1, \ldots, s_C, A_f \rangle, \\ & s_c \leq C - 1,\ a = c \\ \dfrac{(\omega - \gamma(s,a))}{\omega}, & j = s,\ a = c. \end{cases} \tag{14}$$

Similarly, for the state $s = \langle s_1, s_2, \ldots, s_c, \ldots, s_C, A_f \rangle$, the uniformized transition probability $\overline{q}(j \mid s,a)$ can be rewritten

as

$$\overline{q}(j \mid s,a) = \begin{cases} \dfrac{\left(\omega + \lambda_p - \gamma(s,a)\right)}{\omega}, & j = \left\langle s_1, s_2, \ldots, s_C, A_p \right\rangle,\ a = 0 \\ \dfrac{\lambda_f}{\omega}, & j = \left\langle s_1, s_2, \ldots, s_C, A_f \right\rangle,\ a = 0 \\ \dfrac{s_c \mu}{\xi(c)\omega}, & j = \left\langle s_1, s_2, \ldots, s_c - 1, \ldots, s_C, F_c \right\rangle,\ s_c \geq 1,\ a = 0 \\ \dfrac{(s_c + 1)\mu}{\xi(c)\omega}, & j = \left\langle s_1, s_2, \ldots, s_c, \ldots, s_C, F_c \right\rangle,\ a = c \\ \dfrac{s_m \mu}{\xi(m)\omega}, & j = \left\langle s_1, s_2, \ldots, s_m - 1, \ldots, s_c + 1, \ldots, s_C, F_m \right\rangle,\ s_m \geq 1,\ m \neq c,\ a = c \\ \dfrac{\lambda_p}{\omega}, & j = \left\langle s_1, s_2, \ldots, s_c + 1, \ldots, s_C, A_p \right\rangle,\ s_c \leq C - 1,\ a = c \\ \dfrac{\lambda_f}{\omega}, & j = \left\langle s_1, s_2, \ldots, s_c + 1, \ldots, s_C, A_f \right\rangle,\ s_c \leq C - 1,\ a = c \\ \dfrac{(\omega - \gamma(s,a))}{\omega}, & j = s,\ a = c. \end{cases} \tag{15}$$

And for the state $s = \langle s_1, s_2, \ldots, s_c, \ldots, s_C, F_c \rangle$, the uniformized transition probability $\overline{q}(j \mid s,a)$ is rewritten as

$$\overline{q}(j \mid s,a) = \begin{cases} \dfrac{\lambda_p}{\omega}, & j = \left\langle s_1, s_2, \ldots, s_C, A_p \right\rangle \\ \dfrac{\lambda_f}{\omega}, & j = \left\langle s_1, s_2, \ldots, s_C, A_f \right\rangle \\ \dfrac{s_c \mu}{\xi(c)\omega}, & j = \left\langle s_1, s_2, \ldots, s_c - 1, \ldots, s_C, F_c \right\rangle,\ s_c \geq 1, \\ \dfrac{(\omega - \gamma(s,a))}{\omega}, & j = s. \end{cases} \tag{16}$$

Using the uniformization equations presented above, then the expected maximal long-term reward in (12) can be uniformized as

$$\overline{r}(s,a) = r(s,a)\,\frac{\gamma(s,a) + \alpha}{(\alpha + \omega)} \tag{17}$$

and the parameter λ can be uniformized as $\overline{\lambda} = \omega/(\omega + \alpha)$.

Thus, according to the uniformization equations (14), (15), (16), and (17), the uniformized maximal long-term expected reward is obtained as

$$\overline{v}(s) = \max_{a \in \mathrm{Act}_s} \left\{ \overline{r}(s,a) + \overline{\lambda} \sum_{j \in S} \overline{q}(j \mid s,a)\, \overline{v}(j) \right\}. \tag{18}$$

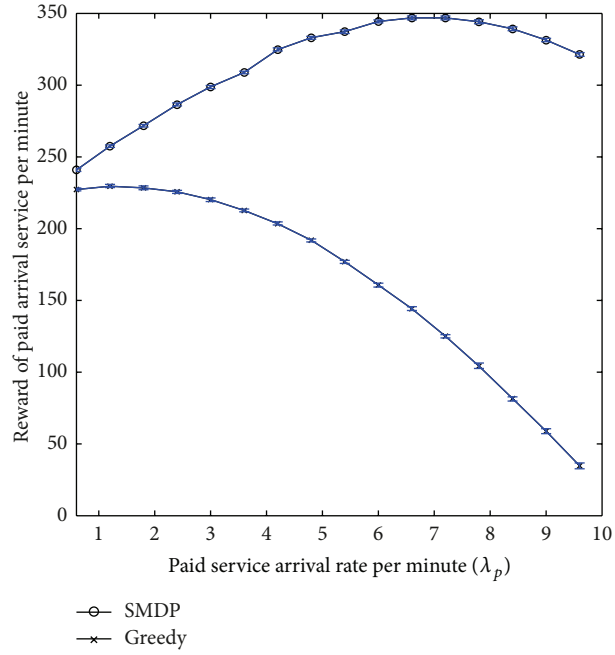


Figure 3: System reward of *paid* service compared between SMDP model and greedy method, varying with the arrival rate of *paid* service requests ($\lambda_f = 2.4$, $\mu = 6.6$, $K = 6$).

5. Performance Analysis

The probability of allocation scheme c, which is defined as the probability that c VMs are allocated for a cloud service, is an important performance metric for ensuring the user satisfaction level and the Cloud resource utilization ratio. It is very useful for the operator to manage the system capacity/utilization status based on the system parameters of the service provisioning domain (such as arrival rate, departure rate, and the VM number of Cloud resource). Meanwhile, blocking service request does not only mean the loss of whole system reward, but also means the degradation of users' satisfaction level. Then, the blocking probability, which is the probability that blocking the cloud service requests from mobile device, is another important performance metrics for the service provisioning domain. In this section, we analytically derive the probabilities of each allocation scheme and blocking probability for the proposed economic mobile computing model based on SMDP.

From the reward function (18) and probability equations (14), (15), and (16), the expected total discounted reward $\overline{v}(s)$ at state $s \in S$ is related with the arrival rates of *paid* service request (λ_p) and *free* service request (λ_f), the departure rate ($\mu/\xi(c)$) of each allocation scheme, the occupied Cloud resource expressed by the number of being occupied VMs $\sum_{c=1}^{C}(s_c * c)$, and the capability of the service provisioning domain (i.e., the total number of VMs-K). For a given service provisioning domain and a certain system state of an arrival of service request (i.e., $\langle s_1, s_2, \ldots, s_C, A_p \rangle$ or $\langle s_1, s_2, \ldots, s_C, A_f \rangle$), the above parameters λ_p, λ_f, $\mu/\xi(c)$,

$\sum_{c=1}^{C}(s_c * c)$, and K are fixed. As a result, the steady-state probability of each state can be obtained from the probability equations (14), (15), and (16). Thus, the probabilities of each allocation scheme and blocking probability can also be achieved through the steady-state probability of each state.

Let π_s denote the steady-state probability of the system state s in the service provisioning domain. From the example in Figure 2 and Table 1, the steady-state probability of $\pi_{\langle s_1,s_2,\ldots,s_C,e\rangle}$ can be classified as three types: (1) the arrival of a *paid* service request; (2) the arrival of a *free* service request; (3) the departure of a finished service with c allocation scheme. Based on the probability equations (14), (15), and (16), the steady-state probabilities $\pi_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}$ and $\pi_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}$ can be derived as follows

$$\begin{aligned}
&\pi_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}\\
&\quad=\frac{\lambda_p}{\gamma(s,a)}\rho_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}\pi_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}\\
&\qquad+\frac{\lambda_p}{\gamma(s,a)}\rho_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}\pi_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}\\
&\qquad+\frac{\lambda_p}{\gamma(s,a)}\sum_{c=1}^{C}\rho_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_p\rangle}\pi_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_p\rangle}\\
&\qquad+\frac{\lambda_p}{\gamma(s,a)}\sum_{c=1}^{C}\rho_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_f\rangle}\pi_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_f\rangle}\\
&\qquad+\frac{\lambda_p}{\gamma(s,a)}\sum_{c=1}^{C}\pi_{\langle s_1,s_2,\ldots,s_C,F_c\rangle}
\end{aligned}\tag{19}$$

$$\begin{aligned}
&\pi_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}\\
&\quad=\frac{\lambda_f}{\gamma(s,a)}\rho_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}\pi_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}\\
&\qquad+\frac{\lambda_f}{\gamma(s,a)}\rho_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}\pi_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}\\
&\qquad+\frac{\lambda_f}{\gamma(s,a)}\sum_{c=1}^{C}\rho_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_p\rangle}\pi_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_p\rangle}\\
&\qquad+\frac{\lambda_f}{\gamma(s,a)}\sum_{c=1}^{C}\rho_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_f\rangle}\pi_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_f\rangle}\\
&\qquad+\frac{\lambda_f}{\gamma(s,a)}\sum_{c=1}^{C}\pi_{\langle s_1,s_2,\ldots,s_C,F_c\rangle},
\end{aligned}\tag{20}$$

where $\rho_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}$, $\rho_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}$, $\rho_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_p\rangle}$ and $\rho_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_f\rangle}$, are the parameters decided by the correlative actions respectively as follows:

$$\begin{aligned}
\rho_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}&=\begin{cases}1, & a_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}=0,\\ 0, & \text{otherwise},\end{cases}\\
\rho_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}&=\begin{cases}1, & a_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}=0,\\ 0, & \text{otherwise}.\end{cases}
\end{aligned}$$

$$\begin{aligned}
&\rho_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_p\rangle}\\
&\quad=\begin{cases}1, & a_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_p\rangle}=c,\ c\subseteq\{1,2,\ldots,C\},\\ 0, & \text{otherwise},\end{cases}\\
&\rho_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_f\rangle}\\
&\quad=\begin{cases}1, & a_{\langle s_1,s_2,\ldots,s_{c-1},\ldots,s_C,A_f\rangle}=c,\ c\subseteq\{1,2,\ldots,C\},\\ 0, & \text{otherwise}.\end{cases}
\end{aligned}\tag{21}$$

Similarly, the steady-state probability $\pi_{\langle s_1,s_2,\ldots,s_C,F_c\rangle}$ can be attained as

$$\begin{aligned}
&\pi_{\langle s_1,s_2,\ldots,s_C,F_c\rangle}\\
&\quad=\frac{(s_c+1)\mu}{\xi(c)\gamma(s,a)}\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,A_p\rangle}\pi_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,A_p\rangle}\\
&\qquad+\frac{(s_c+1)\mu}{\xi(c)\gamma(s,a)}\rho_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}\pi_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}\\
&\qquad+\frac{(s_c+1)\mu}{\xi(c)\gamma(s,a)}\sum_{m=1,m\neq c}^{C}\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_{m-1},\ldots,s_C,A_p\rangle}\\
&\qquad\qquad\times\pi_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_{m-1},\ldots,s_C,A_p\rangle}\\
&\qquad+\frac{(s_c+1)\mu}{\xi(c)\gamma(s,a)}\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,A_f\rangle}\pi_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,A_f\rangle}\\
&\qquad+\frac{(s_c+1)\mu}{\xi(c)\gamma(s,a)}\rho_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}\pi_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}\\
&\qquad+\frac{(s_c+1)\mu}{\xi(c)\gamma(s,a)}\sum_{m=1,m\neq c}^{C}\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_{m-1},\ldots,s_C,A_f\rangle}\\
&\qquad\qquad\times\pi_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_{m-1},\ldots,s_C,A_f\rangle}\\
&\qquad+\frac{(s_c+1)\mu}{\xi(c)\gamma(s,a)}\sum_{m=1}^{C}\pi_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,F_m\rangle},
\end{aligned}\tag{22}$$

where $\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,A_p\rangle}$, $\rho_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}$, $\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_{m-1},\ldots,s_C,A_p\rangle}$, $\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,A_f\rangle}$, $\rho_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}$, and $\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_{m-1},\ldots,s_C,A_f\rangle}$ are defined by the related actions

Table 2: Simulation parameters.

Parameter	Value
E_d	50
δ_d	30
β	1
γ_d	1
U_d	10
θ_d	60
θ_s	12

respectively as

$$\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,A_p\rangle} = \begin{cases} 1, & a_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,A_p\rangle} = 0, \\ 0, & \text{otherwise}, \end{cases}$$

$$\rho_{\langle s_1,s_2,\ldots,s_C,A_p\rangle} = \begin{cases} 1, & a_{\langle s_1,s_2,\ldots,s_C,A_p\rangle} = c,\ c \subseteq \{1,2,\ldots,C\}, \\ 0, & \text{otherwise}, \end{cases}$$

$$\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_{m-1},\ldots,s_C,A_p\rangle} = \begin{cases} 1, & a_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_{m-1},\ldots,s_C,A_p\rangle} = m,\ c \subseteq \{1,2,\ldots,C\}, \\ & \quad m \subseteq \{1,2,\ldots,C\},\ m \neq c, \\ 0, & \text{otherwise}, \end{cases}$$

$$\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,A_f\rangle} = \begin{cases} 1, & a_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_C,A_f\rangle} = 0, \\ 0, & \text{otherwise}, \end{cases}$$

$$\rho_{\langle s_1,s_2,\ldots,s_C,A_f\rangle} = \begin{cases} 1, & a_{\langle s_1,s_2,\ldots,s_C,A_f\rangle} = c,\ c \subseteq \{1,2,\ldots,C\}, \\ 0, & \text{otherwise}, \end{cases}$$

$$\rho_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_{m-1},\ldots,s_C,A_f\rangle} = \begin{cases} 1, & a_{\langle s_1,s_2,\ldots,s_{c+1},\ldots,s_{m-1},\ldots,s_C,A_f\rangle} = m,\ c = \{1,2,\ldots,C\}, \\ & \quad m \subseteq \{1,2,\ldots,C\},\ m \neq c, \\ 0, & \text{otherwise}. \end{cases} \tag{23}$$

Since the sum of the steady-state probabilities for all states equals to 1, we have

$$\sum_S \left(\pi_{\langle s_1,s_2,\ldots,s_C,A_p\rangle} + \pi_{\langle s_1,s_2,\ldots,s_C,A_f\rangle} + \pi_{\langle s_1,s_2,\ldots,s_C,F_c\rangle}\right) = 1. \tag{24}$$

Therefore, the steady-state probability of each state in an MCC service provisioning domain can be obtained by solving (19), (20), (22), and (24). Thus, as a result, for the service request arrival states (i.e., $\langle s_1, s_2, \ldots, s_C, A_p\rangle$ and $\langle s_1, s_2, \ldots, s_C, A_f\rangle$) in one service provisioning domain, the probability of each action can be achieved, which is the ratio of the sum of all steady-state probabilities with the same action to the sum of the steady-state probabilities of all service request arrival states (i.e., $\langle s_1, s_2, \ldots, s_C, A_p\rangle$ or $\langle s_1, s_2, \ldots, s_C, A_f\rangle$) in one domain. Let Pp_a and Pf_a denote the probability of each action for *paid* service request and *free* service request, respectively, then, Pp_a and Pf_a can be expressed as

$$Pp_a = \frac{\sum_{a_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}=a} \pi_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}}{\sum_{m=0}^{C}\left(\sum_{a_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}=m} \pi_{\langle s_1,s_2,\ldots,s_C,A_p\rangle}\right)}, \quad a = \{0,1,2,\ldots,C\}, \tag{25}$$

$$Pf_a = \frac{\sum_{a_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}=a} \pi_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}}{\sum_{m=0}^{C}\left(\sum_{a_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}=m} \pi_{\langle s_1,s_2,\ldots,s_C,A_f\rangle}\right)}, \quad a \subseteq \{0,1,2,\ldots,C\}. \tag{26}$$

Based on (26) and (25), the blocking probability for the service request arrival states (i.e, $\langle s_1, s_2, \ldots, s_C, A_p\rangle$ and $\langle s_1, s_2, \ldots, s_C, A_f\rangle$) in one service provisioning domain can be obtained and denoted as Pp_0 and Pf_0, respectively.

The high values of Pp_0 and Pf_0 do not only mean the loss of the whole system reward but also the decrease of the QoS of the service provisioning domain. Thus, the blocking probabilities Pp_0 and Pf_0 are very important metrics to measure the capability and QoS of a service provisioning domain. In the next section, we will illustrate the relationships between the blocking probability (i.e., Pp_0 and Pf_0) and the parameters (such as λ_p, λ_f, μ, and K) based on the simulation results.

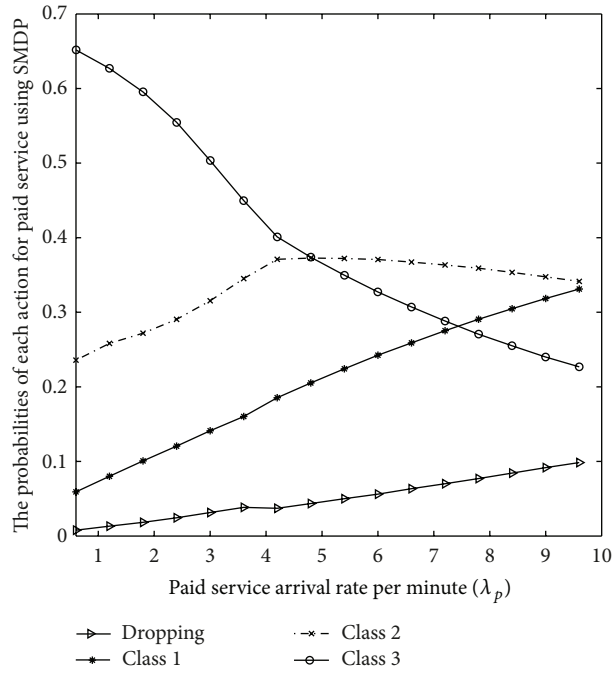


Figure 4: Probabilities for each action of *paid* service using SMDP model, varying with the arrival rate of *paid* service requests (λ_f = 2.4, μ = 6.6, K = 6).

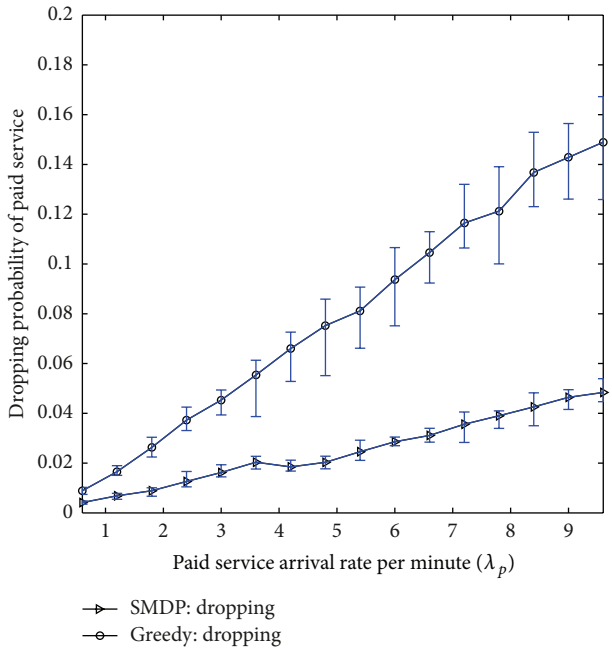


Figure 5: Dropping probability of *paid* service compared between SMDP model and greedy method, varying with the arrival rate of *paid* service requests ($\lambda_f = 2.4$, $\mu = 6.6$, $K = 6$).

Table 3: Resource allocation decision table for each state of paid service ($\lambda_p = 7.2$, $\lambda_f = 2.4$, $\mu = 6.6$, $K = 10$, $s_3 = 0$).

$s_1 \setminus s_2$	0	1	2	3	4	5
0	3	3	3	3	1	0
1	3	3	3	2	1	—
2	3	3	3	1	0	—
3	3	3	2	1	—	—
4	3	2	1	0	—	—
5	3	2	1	—	—	—
6	2	1	0	—	—	—
7	2	1	—	—	—	—
8	1	0	—	—	—	—
9	1	—	—	—	—	—
10	0	—	—	—	—	—

6. Performance Evaluation

In this section, we evaluate the performance of the proposed economic MCC model based on SMDP by using an event driven simulator compiled by Matlab [20] and compare our proposed model with the traditional greedy algorithm. Since the *paid* service demands a higher QoS level compared with other *free* services, thus our simulation mainly focuses on the performance of *paid* service.

In our simulation, the maximal number of VMs is $C = 3$, and the scheme that allocates $c_1 = 1$, $c_2 = 2$, and $c_3 = 3$ VMs to a service is denoted as allocation scheme c_i. The time to process an application service by the Cloud is assumed as a linear function of the number of VMs allocated to the service, which can be denoted as $\xi(c) = 1/c$. Thus, the value of $\xi(c_1)$, $\xi(c_2)$ and $\xi(c_3)$ can be obtained as $\xi(c_1) = 1$, $\xi(c_2) = 1/2$, and $\xi(c_3) = 1/3$. The total resource capability of the service provisioning domain is up to $K = 10$ VMs. Unless otherwise specified, the arrival rates of the *paid* and *free* service request are $\lambda_p = 7.2$ and $\lambda_f = 2.4$, respectively, and the departure rate of finished service occupying one VM is $\mu = 6.6$. Since the time to process the application service occupying one VM is $1/\mu$, then the departure rate of finished service occupying multiple VMs is $\mu/\xi(c)$ which is described in Section 3. Thus, the departure rates of finished service occupying one, two, and three VMs are $\mu_{c_1} = 6.6$, $\mu_{c_2} = 13.2$, and $\mu_{c_3} = 19.8$, respectively. To assure reward computation convergence, the continuous-time discounting factor α is set to be 0.1. The simulation results are collected with each experiment running 18000 s, and each experiment runs 1000 rounds. The other parameters used in this simulation are listed in Table 2.

Table 4: Resource allocation decision table for each state of paid service ($\lambda_p = 60$, $\lambda_f = 2.4$, $\mu = 6.6$, $K = 10$, $s_3 = 0$).

$s_1 \setminus s_2$	0	1	2	3	4	5
0	3	3	2	1	1	0
1	3	3	2	1	1	—
2	2	2	1	1	0	—
3	2	2	1	1	—	—
4	2	1	1	0	—	—
5	1	1	1	—	—	—
6	1	1	0	—	—	—
7	1	1	—	—	—	—
8	1	0	—	—	—	—
9	1	—	—	—	—	—
10	0	—	—	—	—	—

6.1. Optimal Actions. Tables 3 and 4 illustrate the actions of optimal resource allocation at each system state with different arrival rates of the *paid* service λ_p. The numbers in the tables represent the optimal decisions made on state $\langle s_1, s_2, s_3, e\rangle$. The symbol "—" in the tables denotes that the state does not exist. When no user is in the service provisioning domain, 3 VMs (which implies that the action $a = 3$ is made) are allocated to the *paid* service in both two scenarios, when a *paid* service request arrives. If there are $s_2 = 3$ services in the service provisioning domain, which means that the number of the occupied VMs is 6, thus, there are 4 unoccupied VMs available in the service provisioning domain. Our proposed model allocates 3 VMs ($a = 3$) to the *paid* service request when the arrival rate of *paid* service requests is low ($\lambda_p = 7.2$) and allocates 2 VMs ($a = 2$) to the *paid* service request when the arrival rate of *paid* service requests is high ($\lambda_p = 60$), which implies that when the arrival rate of *paid* service requests increases, our model becomes more conservative to allocate resources to the *paid* service requests. The reason is, for example, for the state $\langle 0, 3, 0, A_p\rangle$, the corresponding lump incomes $w(s, a)$ for c_1, c_2, and c_3 are 8, 14, and 16, respectively. Due to the small variance between the lump incomes obtained by allocating c_2 and c_3 VMs to the *paid*

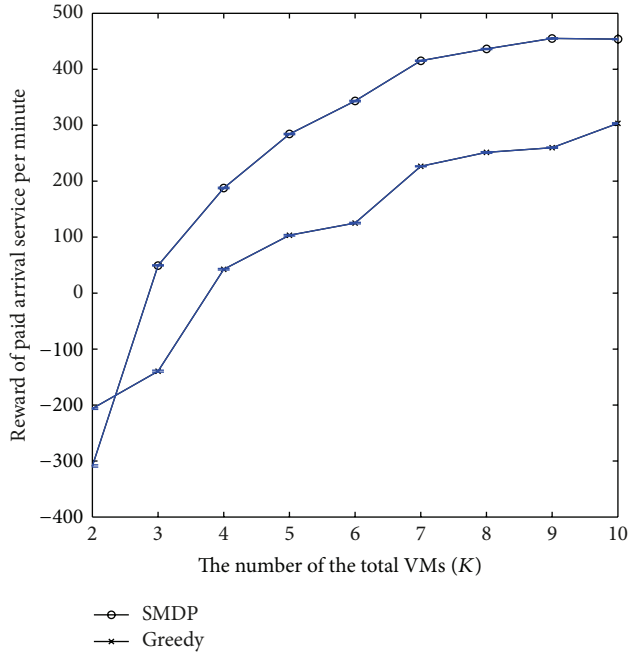


Figure 6: System reward of *paid* service compared between SMDP model and greedy method, varying with the number of VMs (K) ($\lambda_p = 7.2$, $\lambda_f = 2.4$, $\mu = 6.6$).

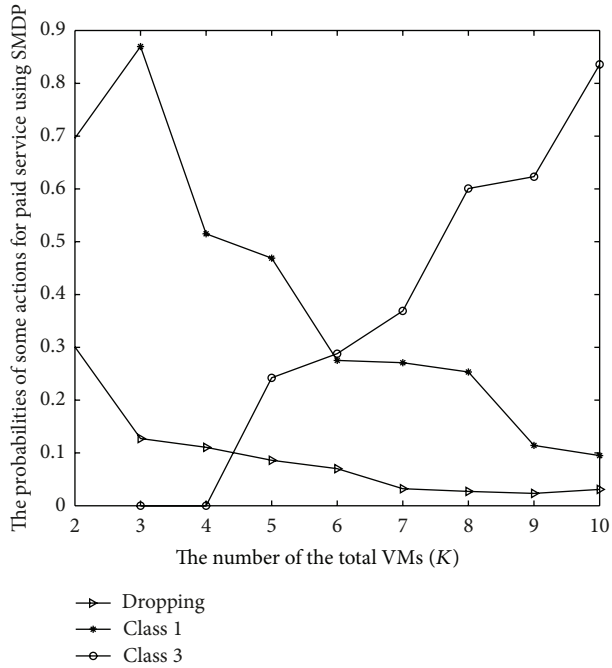


Figure 7: Probabilities for each action of *paid* service using SMDP model, varying with the number of VMs (K) ($\lambda_p = 7.2$, $\lambda_f = 2.4$, $\mu = 6.6$).

service request, when the arrival rate of *paid* service requests increases (i.e., $\lambda_p = 60$), our model prefers action $a = 2$ other than action $a = 3$, since action $a = 2$ can accommodate more *paid* services to gain higher rewards of the MCC system than action $a = 3$, which consumes more Cloud resources of the service provisioning domain.

6.2. System Rewards and Blocking Probability. To evaluate the performance of the proposed dynamic resource allocation model, we compare the long-term reward and blocking probability of the *paid* service between our model and greedy method in Figures 3, 4, and 5. In Figure 3, the reward of *paid* service of our model increases at the beginning, then falls down with the increase of the arrival rate of *paid* service requests (λ_p), while the reward of *paid* service using the greedy method declines always. It can be seen in this figure that the reward of the *paid* service of our proposed model performs much better than that of greedy method. In Figure 4, with the increase of the arrival rate of the *paid* service requests, our model would rather to allocate more c_1 and c_2 VMs to the *paid* service request other c_3 VM; thus, the dropping probability of our model is lower than that of the greedy method which can be seen in Figure 5 as well. As the rejection has more impact on the system lump income compared with acceptance (in our simulation, the lump income $w(s, a)$ or fine of rejection is -70, while the corresponding lump incomes $w(s, a)$ for c_1, c_2, and c_3 are 8, 14, and 16, resp.), thus the lower dropping probability of our model gains more rewards of *paid* service than the greedy method. We can also see in Figure 4 that when the arrival rate of the *paid* service requests is over 7, the probabilities to allocate c_1 and c_2 VMs (especially the probability of c_1 VM) exceed the probability to allocate c_3 VM, which explains the reason why the reward of *paid* service of our proposed model falls down when the arrival rate of *paid* service requests exceeds 7 as shown in Figure 3. In a word, our model can achieve higher reward of *paid* service while keeping lower dropping probability of *paid* service requests at the same time comparing with the greedy method, which are shown in Figures 3 and 5, respectively. Thus, our model outperforms the greedy method with the increase of arrival rate of *paid* service requests.

To further illustrate the performance of our model, we compare the reward of *paid* service and the blocking probability with the greedy method under the scenario of different number of VMs (K). In Figure 6, the rewards of both our model and greedy method increase with the increase of the number of total VMs in the service provisioning domain.

When the number of VMs (K) is less than 2, the rewards of both our model and greedy method are negative. This is because the absolute value of rejection cost (-70) is much higher than the net lump rewards of acceptance (8, 14, and 16 for c_1, c_2, and c_3, resp.) in our simulation.

When the number of total VMs in the service provisioning domain is low (1 and 2), the rejection probability of *paid* service requests is as high as 30% as shown in Figures 7 and 8, which results in the negative rewards for both our model and greedy algorithm. We also observed that when K is less than 3, the reward of *paid* service of our model is lower than that of the greedy method.

The reason is that our model does not only consider the instant and future long-term income but also the cost of

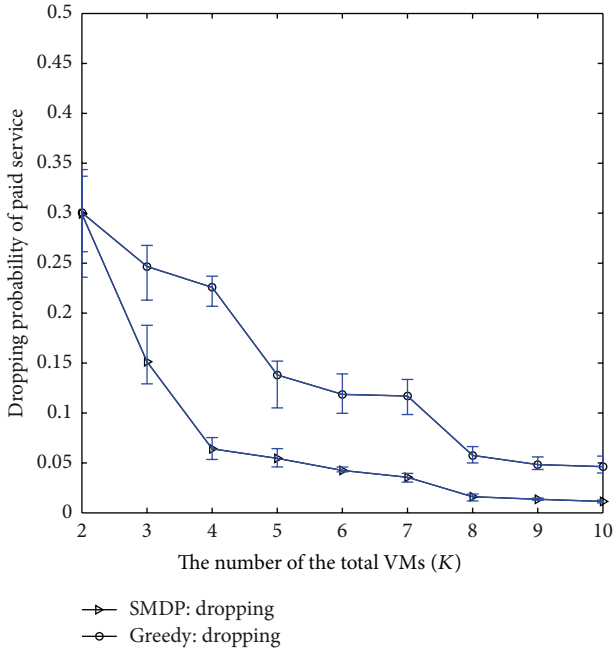


Figure 8: Dropping probability of *paid* service compared between SMDP model and greedy method, varying with the number of VMs (K) ($\lambda_p = 7.2$, $\lambda_f = 2.4$, $\mu = 6.6$).

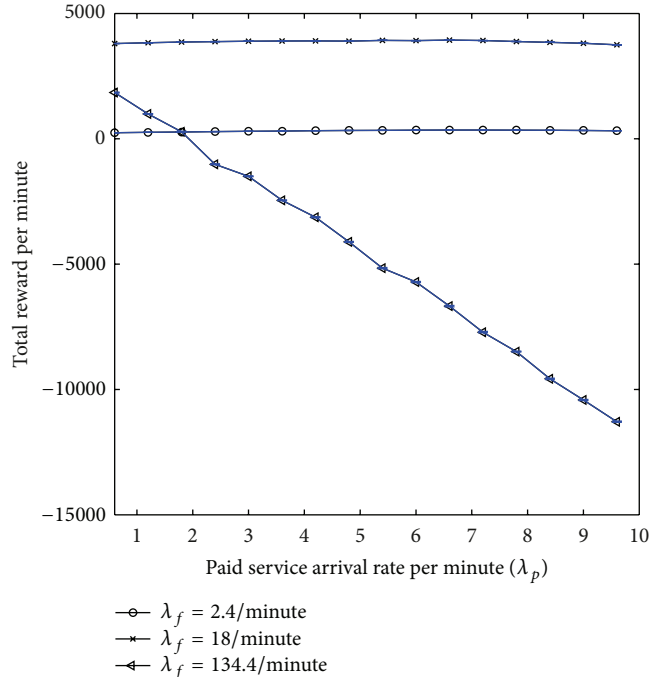


Figure 9: Total system reward with different arrival rate of *free* service requests using SMDP model, varying with the arrival rate of *paid* service requests ($\mu = 6.6$, $K = 6$).

resource occupation of all running services in the service provisioning domain when deciding to allocate the Cloud resources to the *paid* service request, while the greedy method only considers the current income of *paid* service of the service provisioning domain. Then, when the Cloud resource of the service provisioning domain is less than 3 VMs, our model is more conservative than the greedy method to allocate Cloud resources to the *paid* service request.

In Figure 6, we can also see that when the number of VMs (K) is less than 7, the reward of *paid* service of our model increases rapidly with the increase of K, while when K is greater than 7, the reward of *paid* service of our model increases slowly with the increase of K, which implies that when the Cloud resource of the service provisioning domain exceeds the threshold, for the given arrival rate and departure rate, it has limited impact to increase the reward of *paid* service through increasing the Cloud resource of the service provisioning domain. Comparing the rewards of *paid* service between our model and the greedy method in Figure 6, it can be seen that our model outperforms over 50% averagely than the greedy method. Meanwhile, as shown in Figure 8, the dropping probability of *paid* service requests of our model is lower than that of the greedy method over 50% averagely as well, which proves that our model performs better than the greedy method with the increase of the total number of VMs (or Cloud resources) of the service provisioning domain as well.

Figure 9 shows the total rewards (rewards of *paid* service plus *free* service) of different arrival rates of *free* service requests of our proposed model, varying with the increase of arrival rate of *paid* service requests in the service provisioning domain. It can be seen that when the values of the arrival rates between *paid* service request and *free* service request are comparable, the total reward of our model increases with the increase of arrival rate of *free* service requests. On the other hand, when the arrival rate of *free* service requests is much larger than that of *paid* service requests, the total reward decreases rapidly, which results from the large increase of the arrival rate of *free* service requests which may cause more rejections for the following service requests.

7. Conclusion

In this paper, we propose an SMDP-based model to adaptively allocate Cloud resources in terms of VMs based on requests from mobile users. By considering the benefits and expenses of both Cloud and mobile devices, the proposed model is able to dynamically allocate different numbers of VMs to mobile applications based on the Cloud resource status and system performance, thus to obtain the maximal system rewards and to achieve various QoS levels for mobile users. We further derive the Cloud service blocking probability and the probabilities of different Cloud resource allocation schemes in our proposed model. Simulation results show that the proposed model can achieve a higher system reward and a lower service blocking probability compared with the traditional greedy resource allocation algorithm. In the future, we will study a more complex decision making model with different types of mobile application services, for example, the mobile application services which require different serving priorities. We will also investigate the optimal Cloud resource planning by

determining the minimal Cloud network resources to achieve the maximal system rewards under given QoS constraints.

Acknowledgments

This work was supported in part by the State Key Development Program for Basic Research of China (Grant no. 2011CB302902), the "Strategic Priority Research Program" of the Chinese Academy of Sciences (Grant no. XDA06040100), the National Key Technology R&D Program (Grant no. 2012BAH20B03), US NSF Grants CNS-1029546, and the Office of Naval Research's (ONR) Young Investigator Program (YIP).

References

[1] M. Armbrust, A. Fox, R. Griffith et al., "Above the clouds: a berkeley view of cloud computing," Tech. Rep. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Calif, USA, 2009.

[2] M. Walshy, "Gartner: Mobile to outpace desktop web by 2013," Online Media Daily.

[3] D. Huang, X. Zhang, M. Kang, and J. Luo, "Mobicloud: a secure mobile cloud frame-work for pervasive mobile computing and communication," in *Proceedings of 5th IEEE International Symposium on Service-Oriented System Engineering*, 2010.

[4] X. H. Li, H. Zhang, and Y. F. Zhang, "Deploying mobile computation in cloud service," in *Proceedings of the 1st International Conference for Cloud Computing (CloudCom '09)*, p. 301, 2009.

[5] B. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution," in *Proceedings of the 12th USENIX HotoS*, 2009.

[6] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong, "Securing elastic applications on mobile devices for cloud computing," in *Proceedings of the ACM workshop on Cloud Computing Security*, pp. 127–134, 2009.

[7] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proceedings of the IEEE INFOCOM*, San Diego, Calif, USA, March 2010.

[8] L. X. Cai, L. Cai, X. Shen, and J. W. Mark, "Resource management and QoS provisioning for IPTV over mmWave-based WPANs with directional antenna," *ACM Mobile Networks and Applications*, vol. 14, no. 2, pp. 210–219, 2009.

[9] H. T. Cheng and W. Zhuang, "Novel packet-level resource allocation with effective QoS provisioning for wireless mesh networks," *IEEE TransacTions on Wireless Communications*, vol. 8, no. 2, pp. 694–700, 2009.

[10] L. X. Cai, X. Shen, and J. W. Mark, "Efficient MAC protocol for ultra-wideband networks," *IEEE Communications Magazine*, vol. 47, no. 6, pp. 179–185, 2009.

[11] H. Liang, D. Huang, and D. Peng, "On economic mobile cloud computing model," in *Proceedings of the International Workshop on Mobile Computing and Clouds (MobiCloud '10)*, 2010.

[12] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *The Journal of Supercomputing*, vol. 54, no. 2, pp. 252–269, 2009.

[13] K. Lorincz, B. R. Chen, J. Waterman, G. Werner-Allen, and M. Welsh, "Resource aware programming in the pixie os," in *Proceedings of the SenSys*, Raleigh, NC, USA, November 2008.

[14] K. Lorincz, B. Chen, J. Waterman, G. Werner-Allen, and M. Welsh, "A stratified approach for supporting high throughput event processing applications," in *Proceedings of the DEBS*, Nashville, Tenn, USA, July 2009.

[15] G. Tesauro, N. K. Jong, R. Das, and M. N. Bennani, "A hybrid reinforcement learning approach to autonomic resource allocation," in *Proceedings of the of ICAC*, Dublin, Ireland, June 2006.

[16] K. Boloor, R. Chirkova, Y. Viniotis, and T. Salo, "Dynamic request allocation and scheduling for context aware applications subject to a percentile response time sla in a distributed cloud," in *Proceedings of the 2nd IEEE International Conference on Cloud Computing Technology and Science*, Indianapolis, Ind, USA, November 2010.

[17] R. Ramjee, D. Towsley, and R. Nagarajan, "On optimal call admission control in cellular networks," *Wireless Networks*, vol. 3, no. 1, pp. 29–41, 1997.

[18] S. O. H. Mine and M. L. Puterman, *Markovian Decision Process*, Elsevier, Amsterdam, The Netherlands, 1970.

[19] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, New York, NY, USA, 2005.

[20] MathWorks, "Matlab," http://www.mathworks.com/.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 929031, 10 pages
http://dx.doi.org/10.1155/2013/929031

Research Article

Trajectory-Based Optimal Area Forwarding for Infrastructure-to-Vehicle Data Delivery with Partial Deployment of Stationary Nodes

Liang-Yin Chen,[1] Song-Tao Fu,[1] Jing-Yu Zhang,[1] Xun Zou,[1] Yan Liu,[2] and Feng Yin[3]

[1] *College of Computer Science, Sichuan University, Chengdu 610064, China*
[2] *School of Software and Microelectronics, Peking University, Beijing 102600, China*
[3] *Campus Network Management Center, Southwest University for Nationalities, Chengdu 610041, China*

Correspondence should be addressed to Yan Liu; ly@ss.pku.edu.cn

Received 11 January 2013; Revised 20 March 2013; Accepted 20 March 2013

Academic Editor: Jianwei Niu

This paper proposes a trajectory-based optimal area forwarding (TOAF) algorithm tailored for multihop data delivery from infrastructure nodes (e.g., Internet access points) to moving vehicles (infrastructure-to-vehicle) in vehicular ad hoc networks (VANETs) with partial deployment of stationary nodes. It focuses on reducing the delivery-delay jitter and improving the low reliability of infrastructure-to-vehicle communication. To adapt with the real world, TOAF supposes that stationary nodes are partially installed at intersections in VANETs, and nodes' trajectories can be calculated and predicted, such as using cloud services and GPS, to find the optimal area where the destination vehicle may receive a packet timely. AP selects the optimal area from the trajectory of the destination vehicle and determines the delivery sequence, which includes stationary and mobile nodes from the AP to the optimal area. During delivery, if a new node finds that the delay to the next stationary node is less than that of the current carrier, it can be added to the sequence, reducing the delivery delay. The addition of a new node continues until the packet reaches the optimal area and infrastructure-to-vehicle communication is achieved. The simulation results confirm that TOAF can improve the performance of delivery-delay jitters and reliability of infrastructure-to-vehicle communication with partial deployment of stationary nodes.

1. Introduction

Vehicular ad hoc networks (VANETs) have recently emerged as a promising area of research, since the existing systems cannot always cope with increasing demand in vehicular communication [1–4]. Some new approaches have been developed to lessen packet delay and to efficiently control urban traffic in VANETs. For example, when broadcast communication is not reliable at certain occasions, Access Points (APs) are used to forward a packet to a large number of vehicles. In these cases, each vehicle has a distinctive trajectory and is represented by a mobile sensor node [5]. In addition, with the development of cloud services and dynamic urban technology [6], nodes can compute and process information by working together or individually [7]. Traffic Control Center (TCC) [8] can easily collect road network conditions and maintain vehicle trajectories by using vehicle-to-infrastructure communication [9, 10]. Therefore drivers guided by the data from the TCC can select better driving paths in the VANETs [11]. Drivers can also get the data from the Access Points (APs), which are sparsely deployed in road networks and interconnected with each other to individual vehicles. Generally speaking, all of these applications need to reduce delivery-delay jitter and improve the reliability of infrastructure-to-vehicle system.

Currently, researchers have developed some mechanism to decrease the delivery delay with DSRC (the standardization of IEEE Dedicated Short Range Communications) [12], such as TSF (trajectory-based statistical forwarding) [11] and STDFS (shared trajectory-based data forwarding

scheme) [13]. For example, TSF [11] installs stationary nodes at every intersection of road networks to decrease delay of infrastructure-to-vehicle communication. Although the above-mentioned mechanism has achieved great effect, there is still room for improvement. None of them have studied the case with sparse stationary nodes, which are set in the intersections of the road networks.

Due to the good predictability and controllability in the choice of the next forwarding node, stationary nodes are more suitable for data delivery compared with mobile nodes [14]. However, increasing the number of stationary nodes is costly and hard to maintain; so it is not very realistic to install stationary nodes on every intersection in VANETs. In fact, many intersections of real world do not have to contain stationary nodes, and some stationary nodes may be located on roadsides rather than at intersections. The packet forwarding from infrastructure to vehicle under the partial deployment of stationary nodes is more generic and should be paid attention. In this work, we propose a reasonable packet forwarding strategy based on partial coverage of stationary nodes in VANETs. It includes a critical relay node selection method in infrastructure-to-vehicle data delivery.

It is not easy to select a suitable relay node sequence from AP to a mobile vehicle [15], especially when there are sparse stationary nodes with a large number of mobile nodes available in VANETs. A bottleneck in the selection of mobile nodes as relay nodes is the randomness of the position of mobile nodes. However, by means of GPS devices [16], vehicle trajectories are reported to AP via vehicle-to-infrastructure communication timely. So AP can decide which nodes should be set as relay nodes to improve the timeliness and reliability of delivery. In addition, cloud services provide the ability for mobile nodes to calculate delivery delay from their current position to a certain destination position based on their own trajectories. Through comparing with the delay that APs predicted, the mobile node can easily decide whether to add and forward the data packet or not. So, the delivery sequence could be altered by the addition of the new node which can provides less delay according to the actual VANET circumstances at that time.

This paper focuses on two key points during packet delivery: (a) creating a transmission scheme on the basis of vehicle trajectories and (b) adjusting the transmission scheme according to actual circumstance. To create the transmission scheme, we used two kinds of delay distributions in searching for the optimal area: (a) packet delivery-delay distribution from the AP to the optimal area and (b) the vehicle travel-delay distribution from the current position of the destination vehicle to the optimal area. Source nodes (i.e., AP) select the optimal area according to the trajectory of the destination vehicle. An optimal area is identified as a position in VANET, which could get packet from the APs and forward the packet to the destination vehicle, thereby minimizing packet delivery delay while satisfying the required packet delivery probability. Once the optimal area is decided, our TOAF determines the delivery sequence of nodes on the basis of the trajectories data given by TCC, namely, (a) the vehicle trajectories reported to APs and (b) predicted node distribution on the basis of traffic statistics. The system forwards the packet toward the optimal area and constantly adds new nodes with less delay than AP predicted, thus optimizing the delay from the current carrier to the next stationary node in the sequence until the optimal area is reached and infrastructure-to-vehicle communication is achieved. With the application of cloud services and GPS technology, nodes can calculate delivery delay from their current position to a certain position on the basis of its own trajectory and traffic statistics. Using vehicle trajectories instead of traffic statistics improve delay estimates. TOAF ensures effective sequence selection and rearrangement. The intellectual contributions of this study are as follows.

(1) An optimal area selection algorithm for infrastructure-to-vehicle data delivery on the basis of partial coverage of stationary nodes.

(2) An algorithm that determines the delivery sequence of nodes by using vehicle trajectories and traffic statistics.

(3) A strategy that decreases delivery delay from the AP to the optimal area.

The rest of the paper is organized as follows. Section 2 summarizes related studies regarding VANET communication and generates strategies in relay node selection. Section 3 analyzes the data delivery problem with stationary nodes partially installed in intersections. Section 4 presents the TOAF design. Section 5 shows the effectiveness of TOAF via simulation, and Section 6 concludes the paper.

2. Related Works

Data forwarding in VANET is different from that of traditional mobile ad-hoc networks. Vehicular assisted data dissemination (VADD) [9], static-node assisted adaptive data dissemination (SADV) [5], and trajectory-based data (TBD) [10] schemes process data delivery of vehicle-to-infrastructure communication. VADD utilizes vehicular traffic statistics to achieve data delivery with low delivery delay. SADV proposes a forwarding strategy that leverages stationary nodes in a network for reliable data delivery. TBD utilizes vehicle trajectories and vehicular traffic statistics to decrease communication delay and to increase delivery probability for vehicle-to-infrastructure communication. The above-existing schemes focus on data forwarding from vehicle to a stationary node, such as Access Point (AP).

Trajectory-based forwarding is a hybrid forwarding strategy of the source-based routing and greedy forwarding in ad hoc network [17]. The approximate trajectory is defined by the source node, and each intermediate node makes geographical greedy forwarding along the trajectory, such as DSR and LAR [17]. Unlike these two protocols, DREAM [18] is not an on-demand routing protocol; each node in this proactive location-based protocol maintains a location table for all other nodes in VANET. To maintain the table, each node transmits location packets to other nodes, thus increasing the control overhead. In contrast to DREAM, SIFT [4] proposes a technique where forwarding decisions are shifted from the transmitter to the receiver and are not based on location information but based on timers, which allow network nodes

to select themselves in an autonomous fashion as the most appropriate next forwarding node at each intermediate hop without exchanging any type of control messages. It merely uses the trajectory and the location of the last node that forwarded the packet to forward a data packet from a source to a destination. These protocols did not take advantage of the destination vehicles' statistical characteristics.

Trajectory-based statistical forwarding (TSF) [11] presents the first attempt to investigate how to effectively utilize the statistical characteristics of the destination vehicles for infrastructure-to-vehicle data delivery. It uses the stationary node installed in each intersection and on the trajectory of the destination vehicle, which is termed as the target point. This strategy ensures that the packet will arrive earlier at the target point than the destination vehicle. Upon reaching the target point, the destination vehicle receives the packet. However, TSF requires the installation of stationary nodes at all intersections, which is expensive.

Shared trajectory-based data forwarding scheme (STDFS) [13] aims to provide effective vehicle-to-vehicle communication over multihops in VANETs; therefore, STDFS can be used in infrastructure-to-vehicle communication. Shared trajectory information is used to predict encounters between vehicles and to construct a predicted encounter graph. Based on the encounter graph, STDFS optimizes the forwarding sequence to achieve minimal delivery delay given a specific delivery ratio threshold. However, only certain parts of vehicle trajectories can be shared for privacy or other reasons, and obtaining all trajectory information is difficult. The optimization method from the source node to the destination node does not rely on stationary nodes, thereby increasing the complexity of node operation.

Based on the actual construction of the VANET with partial deployment of stationary nodes, we utilize both the stability of the stationary node and the flexibility of the mobile node to achieve optimum delivery performance. The proposed TOAF algorithm selects relay nodes by using trajectory data given by AP and provides nodes with the ability to predict delivery delays. Thus, TOAF provides a reliable and convenient communication strategy for infrastructure-to-vehicle communication.

3. Problem Formulations

This section formulates data forwarding in VANET. This study aims to realize efficient and reliable packet delivery from APs to moving destination vehicles.

3.1. Assumptions. Figure 1 shows that stationary nodes are partially distributed in VANETs, and intersections 2, 5, 7, 8, 10, and 11 do not have stationary nodes. TCC maintains vehicle trajectories without exposing vehicle trajectories to other vehicles. APs are sparsely deployed in VANETs and are interconnected with each other through wired or wireless networks. APs communicate with nodes and provide DSRC devices information regarding these nodes.

APs acquire vehicle trajectories via vehicle-to-infrastructure communication, such as VADD and TBD. TCC

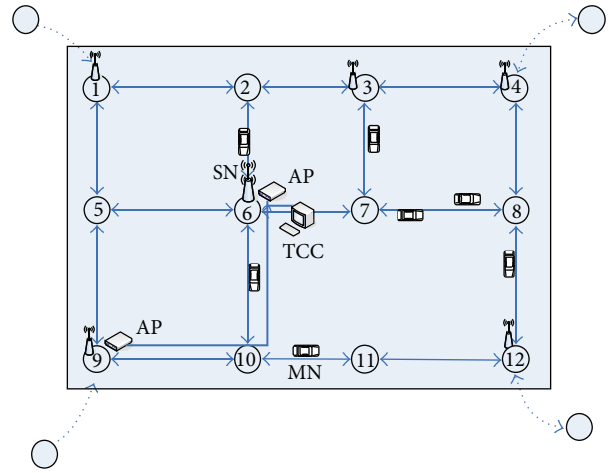


Figure 1: Communication mode of nodes in VANET.

shares the traffic graph of the entire VANET with APs. APs can calculate the optimal area where destination vehicles will receive messages and the transfer sequence $V_n = (v_1, v_2, \ldots, v_n)$, which is selected by APs. Here, V_n is the relay node during data delivery, which includes stationary nodes and mobile nodes from AP to the optimal area. APs select V_n on the basis of vehicle trajectories and predicted traffic statistics. Packets are carried or forwarded by the current relay node in the transfer sequence, or new nodes with shorter delays are added in the sequence until the destination vehicle reaches the optimal area. Delivery ratio relies on stationary nodes, and APs calculate the sequence of stationary node distribution to meet the delivery ratio threshold required by the user.

3.2. Identification of the Optimal Area. Let $f(p)$ be the probability distribution function of the packet delay P, $g(v)$ the probability distribution function of the destination vehicle delay V, TTL the time-to-live (TTL) of the packet, and α the delivery ratio threshold required by the user. In TSF, all the stationary nodes in the intersections are denoted by I, where $i \in I$ is one point in I. The point i should satisfy the following equation to become a target node:

$$P\left[p_i \le V_i\right] = \int_0^{\mathrm{TTL}} \int_0^{v} f(p)\, g(v)\, dp\, dv \ge \alpha. \quad (1)$$

Optimal target point selection depends on the TTL, the packet delay model P, and the vehicle delay model V, which are described in [11]. Different from TSF, the proposed TOAF is based on partial coverage by stationary nodes. Moreover, the two schemes have different delay distributions. We can decide on the location of the optimal area. However, we cannot choose a target stationary node because stationary nodes may not be present in the trajectories of the destination vehicle. Thus, we need other relay nodes to deliver or forward packets to the optimal area, which is based on the trajectories of the destination vehicle.

3.3. Delay Distribution

3.3.1. Vehicle Delay Distribution. The destination vehicle V travels in the region G with Gamma distribution. Let $E(G)$ be the edge in the VANET. For any edge $e_i \in E(G)$, the delay $t \sim \Gamma(\kappa_i, \theta_i), \kappa_i, \theta_i$ is computed by using the average vehicle traveling delay μ_i and the travel-delay variance δ_i^2 by using equations $E[t_i] = \kappa_i\theta_i$ and $\mathrm{Var}[t_i] = \kappa_i\theta_i^2$. Thus, the parameters κ_i, θ_i are computed as follows:

$$\theta_i = \frac{\delta_i^2}{\mu_i}, \qquad \kappa_i = \frac{\mu_i^2}{\delta_i^2}. \tag{2}$$

The vehicle delay distribution in each edge e_i could be computed according to the average delay $E[t_i]$ and variance $\mathrm{Var}[t_i]$ of the vehicle. Assuming that the destination vehicle has moved from the current position to the optimal area, we need to go through N edges in $E(G)$ because the delay distribution of each road is relatively independent from each other. The total vehicle delivery delay is $E[V] = \sum_{i=1}^{N} \mu_i$, and the variance is $\mathrm{Var}[V] = \sum_{i=1}^{N} \delta_i^2$. Thus, the end-to-end delay distribution of the destination vehicle from the current position to the optimal area can be expressed as follows:

$$V \sim \Gamma\left(\kappa_v, \theta_v\right), \qquad E\left[V\right] = \kappa_v\theta_v, \qquad \mathrm{Var}\left[V\right] = \kappa_v\theta_v^2, \quad \text{for } V, \kappa_v, \theta_v > 0. \tag{3}$$

3.3.2. Delay Distribution from AP to the Optimal Area. (I) Delay Distribution between Stationary Nodes. Assume that intersection i is adjacent to intersection j and each intersection has its own stationary node. Let l be the distance between i and j, such that the following two cases exist.

(A) Immediate Forward. In both carry and forward cases, a packet carrier obtains the packet from stationary node i. The packet carrier encounters a vehicle within the DSRC communication range R and forwards the packet to the front node; otherwise, the packet carrier delivers the packet to the communication range of stationary node j. Let the forwarded distance be l_f and the speed of the vehicle v, and the delay for the forwarded distance is only tens of milliseconds, such that we can ignore the delay because it is several orders-of-magnitude less than the carry delay. Set the average rate of the vehicle reaching the intersection i as λ, which obeys the Poisson process. R is the communication range, and v is the vehicle speed. Consider the message forwarding probability $P_f = \beta = 1 - e^{-(\lambda R/v)}$, which denotes that at least one vehicle arrives at the intersection i for the duration R/v; thus, the delay is $((l - R - E[l_f])/v)\beta$.

(B) Wait and Carry. Stationary node i waits for a vehicle and forwards the packet to the vehicle. The vehicle then carries the packet to the communication range of the stationary node i. The waiting probability is $P_w = 1 - \beta = e^{-(\lambda R/v)}$, and the delay is $(1/\lambda + (l - R)/v)(1 - \beta)$.

The node then delays distributions from i to j as $P \sim \Gamma(\kappa_d, \theta_d)$, where

$$\begin{aligned} E\left[d_i\right] &= \frac{l - R - E\left[l_f\right]}{v}\beta + \left(\frac{1}{\lambda} + \frac{l - R}{v}\right)(1 - \beta), \\ \mathrm{Var}\left[d_i\right] &= E\left[d_i^2\right] - \left(E\left[d_i\right]\right)^2 \\ &= \frac{(l - R)^2 - 2\,(l - R)\,E\left[l_f\right] + E\left[l_f^2\right]}{v^2}\beta \\ &\quad + \left(\frac{1}{\lambda} + \frac{l - R}{v}\right)^2(1 - \beta) \\ &\quad - \left(\frac{l - R - E\left[l_f\right]}{v}\beta + \left(\frac{1}{\lambda} + \frac{l - R}{v}\right)(1 - \beta)\right)^2. \end{aligned} \tag{4}$$

(II) Delay Distribution from AP to the Optimal Area. Given that the edges with stationary nodes are independent of each other, the average delay from AP to the optimal area with stationary node coverage is $E[P] = \sum_{i=1}^{N} E[d_i]$ and the variance is $\mathrm{Var}[P] = \sum_{i=1}^{N} \mathrm{Var}[d_i]^2$. By using $E[P] = \kappa_p\theta_p$ and $\mathrm{Var}[P] = \kappa_p\theta_p^2$, we can calculate the distribution function of the data delivery delay between AP and the optimal area as follows:

$$P \sim \Gamma\left(\kappa_p, \theta_p\right), \quad \text{for } P, \kappa_p, \theta_p > 0. \tag{5}$$

(III) Delay Distribution between Intersections without Stationary Nodes. Figure 1 denotes that regardless of intersections 6-7-8–12 or other delivery paths, each intersection is not guaranteed to have at least one stationary node. Thus, the delay of intersections 6–12 needs to be recalculated.

The delay distribution on each edge is consistent with previous descriptions. However, in this study, messages can only be forwarded to another vehicle or carried by the current vehicle in the intersection. Directly calculating the delay between two stationary nodes is difficult. However, by utilizing the vehicle trajectory, we can obtain the trajectory by using three information: report from the vehicle, vehicle owned, or traffic statistics. Suppose a vehicle will travel along a trajectory denoted by a sequence of intersections $j \in N(n)$ between two stationary nodes; then the last intersection of $N(n)$ is N. We can calculate the delay as follows.

Let D be the expected delivery delay (EDD) of the packet, which is located in the source node and transmitted to the intersection N. The probability that a packet is being carried by a vehicle from intersection 1 to intersection j is $P_c = \prod_{h=1}^{j-1} P_{h,h+1}^c$, and $P_{h,h+1}^c = 1 - \prod_{k \in N(h)} P_{hk}$ is the carrying probability from h to $h + 1$, where $N(h)$ is the neighbor intersections of h and P_{hk} is the forwarding probability that the vehicle at intersection h can forward its packets to another vehicle moving toward the neighboring intersection k. The forwarding probability can be obtained

by using traffic statistics. The total carried time of the packet from the source node to intersection j along the trajectory is $C_{1j} = \sum_{k=1}^{j-1} l_{k,k+1}/v$. E_j is the EDD after the packet leaves the current vehicle at j, D_{jk} is the EDD from j to the destination node N when the edge e_{jk} is used as the forwarding edge, and $E_j = \sum_{k\in N(j)} P_{jk}D_{jk}$. Therefore,

$$D = \sum_{j=1}^{N} \left(P_c \times \left(C_{1j} + E_j\right)\right) = \sum_{j=1}^{N} \left(\left(\prod_{h=1}^{j-1} P_{h,h+1}^{c}\right) \times \left(C_{1j} + \sum_{k\in N(j)} P_{jk}D_{jk}\right)\right). \tag{6}$$

For example, from intersection 6 to intersection 12 in Figure 1, the EDD of a vehicle with the trajectories 6-7-8–12, can be calculated as follows:

$$\begin{aligned} D = {} & P_{6,7}D_{6,7} + P_{6,10}D_{6,10} + P_{6,7}^{C} \\ & \times \left(C_{6,7} + P_{7,6}D_{7,6} + P_{7,8}D_{7,8} + P_{7,3}D_{7,3}\right) \\ & + P_{6,7}^{C}P_{7,8}^{C}\left(C_{6,8} + P_{8,4}D_{8,4} + P_{8,7}D_{8,7} + P_{8,12}D_{8,12}\right) \\ & + P_{6,7}^{C}P_{7,8}^{C}P_{8,12}^{C}\left(C_{6,12} + P_{12,8}D_{12,8} + P_{12,11}D_{12,11}\right). \end{aligned} \tag{7}$$

However, if no vehicle contains the trajectories 6-7-8–12, the EDD also can be calculated by the prediction from the traffic statistics given.

We could estimate the delay of the packet along the edges without stationary nodes. Figure 2 shows that paths T_1 and T_2 can reach the target point, node 4. Moreover, path T_1 could not rely on the stationary node in intersections 10 and 5. The delay of the edges of 9-10-11 and 12–5-4 should be re-estimated. When the edges of 8-7-6 and 6-5-4 in path T_2 are the same as T_1, we can calculate the EDD of the edges without stationary nodes as follows:

$$D_{T1} = D_{9,11} + D_{12,4}, \qquad D_{T2} = D_{8,6} + D_{6,4}. \tag{8}$$

3.4. Infrastructure-to-Vehicle Communication Strategy. Given that D_{T_i} is the minimum EDD of the packet transmitted along the certain path T_i in edges without stationary nodes, we can calculate the delay of each path by using (6), which is based on the vehicle trajectories or traffic statistics. Taking for example, D_{T_1} in Figure 2, AP obtains the vehicle distribution of intersections 9–11 according to the average delay of intersections 8-9 and the vehicle trajectories. AP then uses traffic statistics if no vehicle report is available in its trajectory and then calculates the delay of intersections 9–11 and intersections 12–4. AP could set up a new TTL for the packet $\text{TTL}_{\text{new}} = \text{TTL} - D_{T_i}$. The position of the target point, which is our optimal area, can be obtained using the following equation:

$$P_1 = \int_0^{\text{TTL}_{\text{new}}} \int_0^{v} f(p)\, g(v)\, dp\, dv \geq \alpha. \tag{9}$$

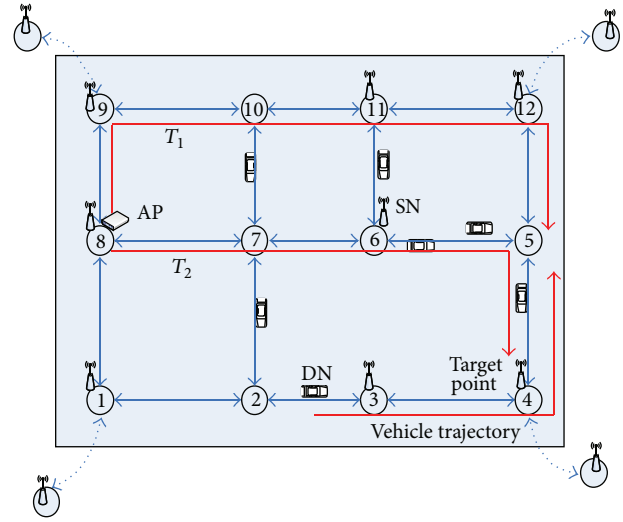


Figure 2: Multiple paths in VANET with partial coverage by stationary nodes.

The optimal area is the area where the target point forwards the message to the destination vehicle v. AP selects the target point in the greater range in VANET when no stationary nodes are available on the trajectories of the destination vehicle. Otherwise, AP finds a vehicle traveling the opposite route of the destination vehicle and uses this vehicle as the target vehicle. Thus, the target vehicle receives the packet before encountering the destination vehicle. Let the traveling delay of the destination vehicle be $V \sim \Gamma(\kappa_v, \theta_v)$, and vehicles $v_1, v_2, \ldots, v_m$ have their respective traveling delay distributions of $V_j \sim \Gamma(\kappa_{vj}, \theta_{vj})$, $j = 1, 2, \ldots, m$. We first determine v_j, which is a vehicle that could encounter the destination vehicle. The stationary nodes in the trajectory of v_j are satisfied by the following equation:

$$\begin{aligned} &(1)\ P_1 = \int_0^{\text{TTL}} \int_0^{v} g(v)\, g\left(v_j\right) dp\, dv_j \geq \alpha, \\ &(2)\ P_2 = \int_0^{\text{TTL}_{\text{new}}} \int_0^{v} f(p)\, g\left(v_j\right) dp\, dv_j \geq \alpha. \end{aligned} \tag{10}$$

The delivery process is divided into two steps: first, AP delivers the packet to vehicle v_j through the target point discussed above; second, v_j forwards the message to the destination vehicle v. The area where v_j forwards the message to the destination vehicle v is the optimal area.

4. TOAF Algorithm

The AP in our TOAF algorithm could obtain vehicle trajectories and predict the node distribution on the basis of traffic statistics through TCC. The AP first determines the optimal area for delivery according to the partial coverage of stationary nodes in intersections and then creates the transfer sequence $V_n = (v_1, v_2, \ldots, v_n)$ by predicting all mobile nodes that reported their trajectories. AP predicts the delay of each stationary node $S_n = (s_1, s_2, \ldots, s_n)$, in which s_n is the position

of the optimal area. The node v_i in V_n delivers the packet according to V_n. When v_i, which is the node carrying the packet, encounters a new node not in the sequence, the new node calculates the EDD to the next stationary node s_i in S_n by using (6) based on its own trajectory and traffic statistics. The new node is added to the sequence if the calculated EDD is less than v_i, provided that the delivery ended when the packet is forwarded to the optimal area S_n.

The delivery from AP to the optimal area is divided into two cases, which is based on whether stationary nodes exist within the TTL in the trajectory of the destination vehicle.

4.1. Coverage by Stationary Nodes in the Trajectory of the Destination Vehicle in TTL. If the predicted vehicle of AP encounters stationary nodes in its trajectory, select a stationary node as the optimal area. The algorithm strategy is as follows:

(1) determine the optimal area S_n in the trajectory of the destination vehicle by using (9),

(2) AP selects the delivery sequence V_n and sends the packet to the optimal area through the sequence. When the node v_i, which carries the packet, encounters a new node v_{new}, the new node estimates the EDD = D_{new} to the next stationary node according to traffic statistics and its own trajectory by using (6). If D_{new} has a shorter delay than the node v_i in the sequence, add v_{new} to V_n until the packet reaches the optimal area s_n. If the destination vehicle arrives at the optimal area when the packet is forwarded to s_n, the destination vehicle gets the packet from s_n, thus completing the delivery process. Node s_n forwards the packet along the reverse trajectory of the destination vehicle when this node has the packet earlier than asked. TOAF adds a new node with less delay to the next stationary node in sequence; therefore the EDD will be lower than predicted. Delivery probability is guaranteed by the stationary nodes because TOAF creates a sequence under the delivery probability α. TOAF does not maintain a backup for each node, thus making sure that the efficiency of frequency resources is maximized.

4.2. No Stationary Node in the Trajectory of the Destination Vehicle in TTL. If AP predicts that selecting a stationary node in the trajectory of the destination vehicle in TTL is impossible, then AP selects a target vehicle v_j as relay node from vehicles that could encounter the destination vehicle. AP determines the sequence V_n, from AP to v_j, as well as sequence V_m from v_j to the optimal area of the destination vehicle.

(1) Determine the target vehicle v_j according to (10).

(2) Deliver the packet from AP to the target vehicle v_j, as discussed in Section 4.1.

(3) Deliver the packet from target vehicle v_j to the destination vehicle. v_j delivers the packet to the destination vehicle as sequence V_m and constantly adds a node that has shorter delay from the current position to the optimal area of the destination vehicle. TOAF ensures the timeliness and reliability of delivery by constantly revising the sequence on the basis of the actual traffic.

5. Performance Evaluation

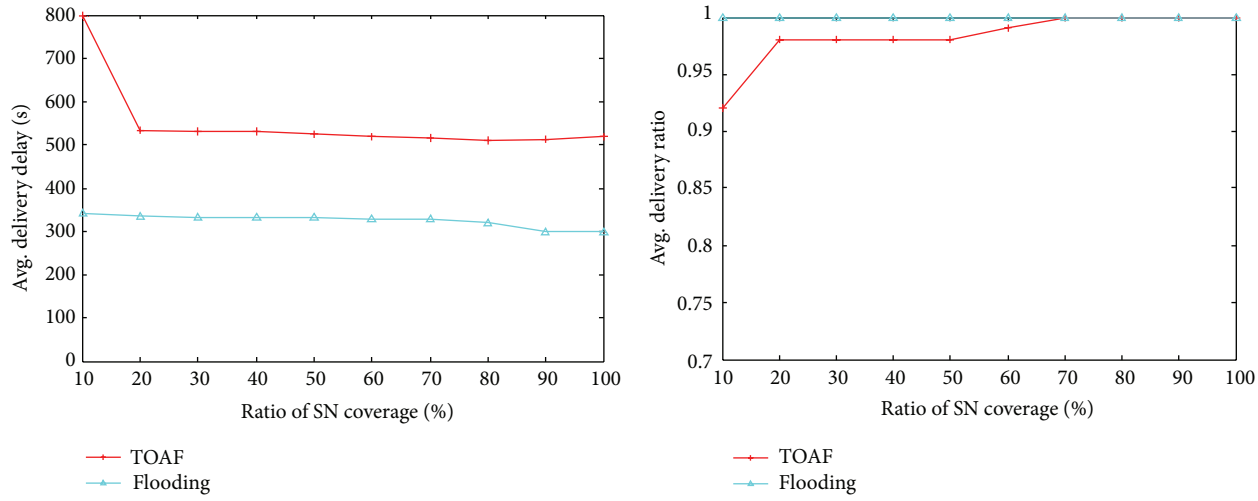
We evaluate the performance of TOAF in this section. The model is based on the description given in [11]; we have built a simulator based on the scheduler provided by SMPL [19] in C with the following settings. A road network with 49 intersections is used in the simulation, and one Access Point is deployed in the center of the network. Each vehicle's movement pattern is determined by a Hybrid Mobility Model of City Section Mobility Model [20] and Manhattan Mobility model [21]. The vehicles are randomly placed at one intersection as a start position among the intersections on the road network, randomly select another intersection as an end position, and wait for a random waiting time (i.e., uniformly distributed from 0 to 10 seconds) at intersections in order to allow the impact of stop sign or traffic signal.

The simulation configuration is shown in Table 1, which has 49 intersections in the range of 8.25 km × 9 km. The communication range is 200 m, and vehicle speed follows the normal distribution of $N(\mu_v, \sigma_v)$. The maximum and minimum speeds are, respectively, $\mu_v + 3\sigma_v$ and $\mu_v - 3\sigma_v$. The default μ_v is 40 MPH, the default σ_v is 5 MPH, and vehicles can change their speed at each road section. The vehicle travel path length is $l \sim N(\mu_l, \sigma_l)$, where $\mu_l = d_{i,j}$ km is the shortest path distance from start position i to end position j in VANET, and $\sigma_l = 3$ km determines a random detour distance. We assume that APs predict the node distribution on the basis of trajectories reported from vehicles and traffic statistics. Packets are dynamically generated randomly from APs to any selected vehicle. The total number of generated packets is 2,000, and the simulation is continued until all of these packets are either delivered or dropped due to TTL expiration. We set the TTL to 2000 s and the delivery probability bound α to 0.9. Considering the randomness of stationary node coverage, we compared TOAF in different stationary node coverage ratios with flooding algorithm (without considering communication conflict and storage limit) and compared with the TSF algorithm and flooding algorithm under different situations in 20% coverage ratio.

5.1. Delivery in Different Coverage Ratios of Stationary Nodes. TOAF mainly predicts the delivery sequence by AP. The stationary nodes in sequence deliver packets with 100% probability. Thus, EDD and delivery ratio are easy to control in stationary nodes. AP creates a reasonable sequence according to the actual situation of the VANET. The current node adjusts the sequence in accordance with the actual situation of the VANET, thus improving the delivery performance under different coverage ratios by stationary nodes. Figure 3 shows that the delivery delay and the delivery ratio have a slight decline with decreased distribution of stationary nodes. Nevertheless, even in the case of 20% coverage, the delivery delay and the delivery ratio are close to full coverage and

Table 1: Simulation configuration.

Parameter	Description
Road network	The number of intersections is 49 The area of the road map is 8.25 km × 9 km
Communication range	$R = 200$ m
Number of vehicles (N)	The number of vehicles moving within the road network. The default value of N is 250
Time-to-live (TTL)	The expiration time of a packet. The default TTL is the vehicle trajectory's lifetime and it is the vehicle's travel time for the trajectory, that is, 2,086 seconds
Vehicle speed (v)	$v \sim N(\mu_v, \sigma_v)$, where $\mu_v = \{20, 25, \ldots, 60\}$; $\sigma_v = \{1, 2, \ldots, 10\}$, the maximum and minimum speeds are, respectively, $\mu_v + 3\sigma_v$ and $\mu_v - 3\sigma_v$. The default μ_v is 40 MPH and the default σ_v is 5 MPH
Vehicle travel path length (l)	$l \sim N(\mu_l, \sigma_l)$, where $\mu_l = d_{i,j}$ km is the shortest path distance from start position i to end position j in VANET, and $\sigma_l = 3$ km determines a random detour distance



(a) Delivery delay versus coverage rates by stationary nodes

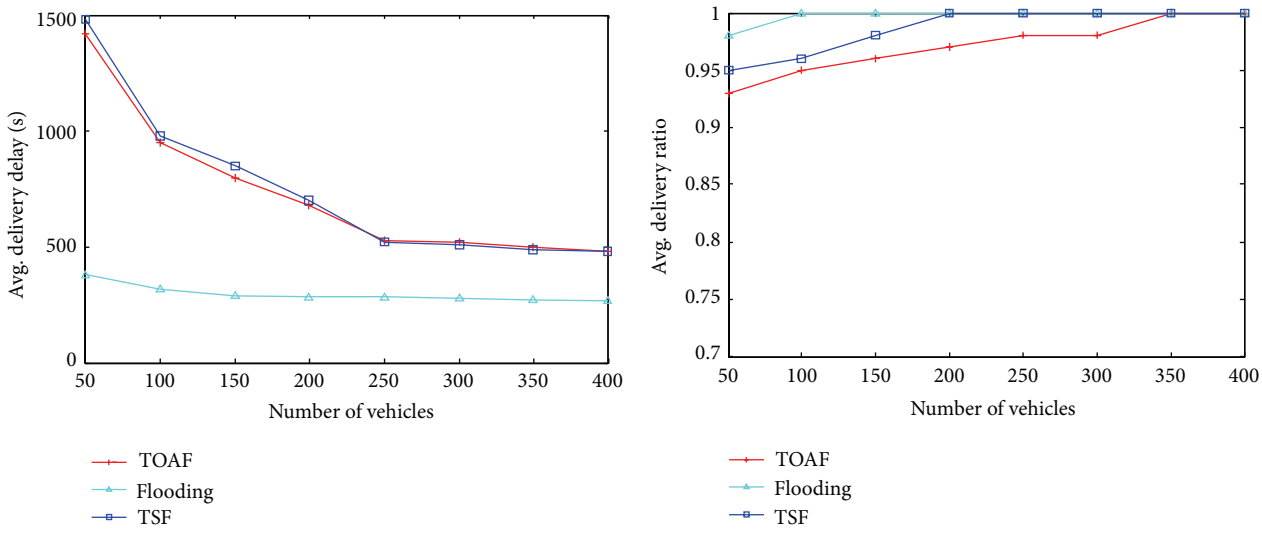
(b) Delivery ratio versus coverage rates by stationary nodes

Figure 3: Performance in different coverage ratios of stationary nodes.

are close to flooding algorithm. The flooding algorithm is a theoretical algorithm subjected to hardware performance and traffic congestion. AP selects sequences on the basis of the optimal area in the trajectory of the destination vehicle and adjusts the nodes during delivery to satisfy the effectiveness and reliability requirements of the infrastructure-to-vehicle communication.

5.2. Influence of Vehicle Number N. Traffic prediction in TOAF may be influenced by the trajectories of moving vehicles. We consider 50 vehicles to 400 vehicles in this study. Figure 4 shows that increasing traffic improves delivery ratio and delay performance. Delivery ratio and delay performance leveled off when more than 150 vehicles were used. The delivery ratio was over 90% when 50 vehicles were used. The delay is slightly lower than the TSF when the number is less than 200, because the sequence is based on the traffic at the beginning of delivery, rather than based on stationary nodes waiting for the arrival of the next vehicle at each intersection. Thus, TOAF can adapt to actual circumstances of VANET by utilizing both stationary nodes and mobile nodes. The flooding delay is significantly lower when a small number of vehicles are involved, which shows that flooding still has a strong competitive advantage in the case of small traffic.

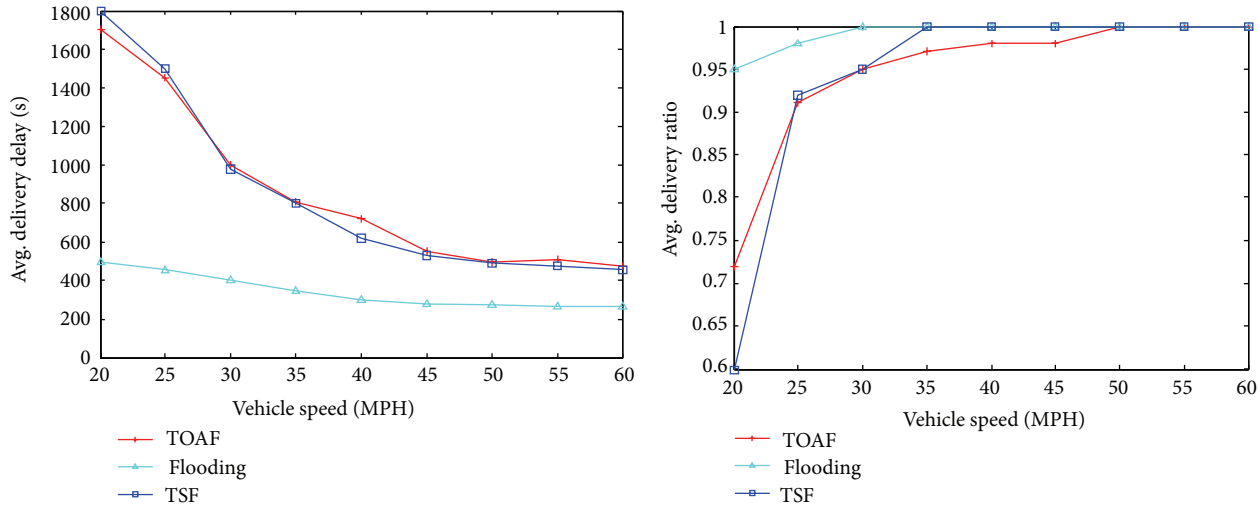
5.3. Influence of Vehicle Speed μ_v. Figure 5 shows that for flooding, TSF, and TOAF with 20% coverage, high-speed vehicles result in low delivery delay because high-speed vehicles yield high vehicle arrival rates at each road segment. The delay in TOAF is less than that of TSF when low-speed vehicles are used because TOAF could add new nodes during the process and TOAF has more chances to decrease the delay. At all vehicle speeds, the performance of TOAF is still close to that of TSF. As shown in Figure 5, when the mean speed is very low, the delivery ratio of TOAF is better than that of TSF. Sequence selection in TOAF ensures that the delivery ratio and performance improves with vehicle speed. Thus TOAF can be used in different circumstances in the VANET.



(a) Delivery delay versus vehicle number

(b) Delivery ratio versus vehicle number

Figure 4: Influence of vehicle number N.



(a) Delivery delay versus vehicle speed

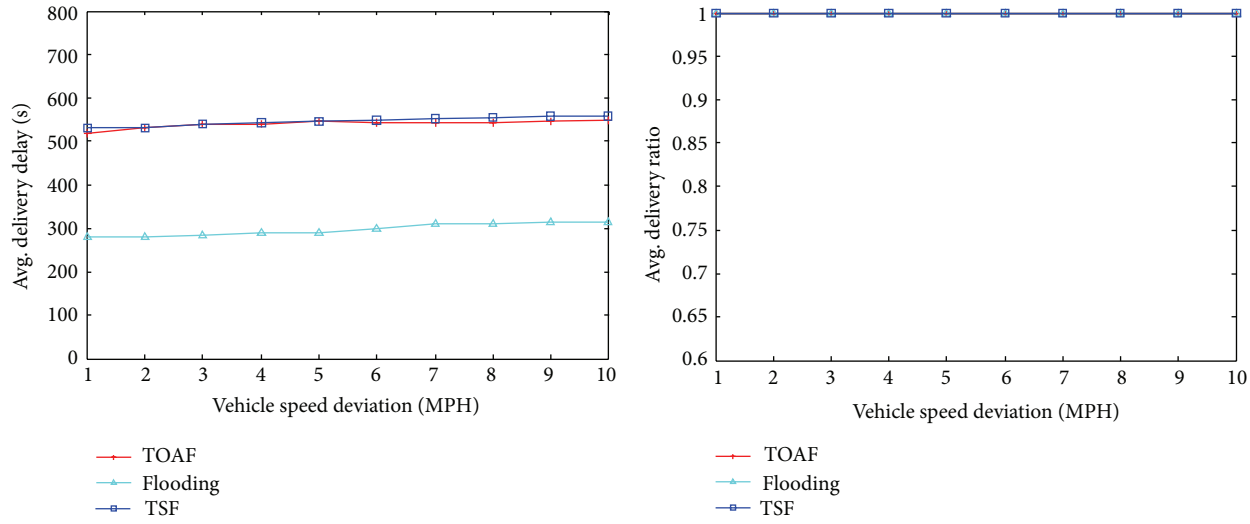
(b) Delivery ratio versus vehicle speed

Figure 5: Influence of vehicle speed μ_v.

5.4. Influence of Vehicle Speed Deviation σ_v. We used vehicle speed deviation to reflect the traffic condition. As shown in Figure 6, average delay increased in TOAF with greater speed deviation. TOAF has lower average delay than TSF because the former adds new nodes under actual circumstance to decrease the delay. The packet delivery ratio has no obvious decrease with increasing speed deviation, and the delivery ratio performance of TOAF is nearly the same as that of TSF and flooding algorithm. These data show that utilizing a small number of stationary nodes can ensure stable delivery ratios. The accuracy and effectiveness of packet forwarding by TOAF will improve when stationary and mobile nodes are both utilized.

6. Conclusion

TOAF is an algorithm based on the actual construction of the VANET with partial coverage of stationary nodes. TOAF continuously adjusts forwarding node sequences during delivery on the basis of actual VANET by predicting the optimal area to the destination vehicle and by selecting a nodes sequence from the AP to the optimal area. Combining the stability of the stationary node with the flexibility of the mobile node, TOAF reduces the randomness of the delivery process and improves the delivery ratio. TOAF also reduces the delay jitters and improves the reliability of infrastructure-to-vehicle communication. The simulation



(a) Delivery delay versus vehicle speed deviation

(b) Delivery ratio versus vehicle speed deviation

Figure 6: Influence of vehicle speed deviation σ_v.

results further validate the effectiveness of TOAF in practice. For our future work, we will explore infrastructure-to-vehicle data forwarding by using the stationary node installed on roadsides, but not at the intersections (e.g., brand of bus station).

Acknowledgments

This work is supported by the National Science and Technology Major Project of China (2011ZX03005-002), the State Key Development Program for Basic Research of China (2011CB302902), China NSF (60933011, 11102124), the Program for New Century Excellent Talents in University, Ministry of Education of China (NCET-10-0604), and the Science and Technology Support Projects Foundation of Sichuan Province of China (2010GZ0170).

References

[1] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in sparse vehicular ad hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1538–1556, 2007.

[2] D. Jiang, Q. Chen, and L. Delgrossi, "Optimal data rate selection for vehicle safety communications," in *Proceedings of the 5th ACM International Workshop on VehiculAr Inter-NETworking (VANET '08)*, pp. 30–38, September 2008.

[3] A. Skordylis and N. Trigoni, "Delay-bounded routing in vehicular ad-hoc networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 341–350, Hong Kong, May 2008.

[4] H. Labiod, N. Ababneh, and M. García de la Fuente, "An efficient scalable trajectory based forwarding scheme for VANETs," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA '10)*, pp. 600–606, April 2010.

[5] Y. Ding, C. Wang, and L. Xiao, "A static-node assisted adaptive routing protocol in vehicular networks," in *Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks (VANET '07)*, pp. 59–68, September 2007.

[6] D. A. Roozemond, "Using intelligent agents for dynamic Urban traffic control systems," in *Proceedings of European Transport Conference (PTRC '99)*, Cambridge, Mass, USA, September 1999.

[7] Q. J. Kong, Z. Li, Y. Chen, and Y. Liu, "An approach to Urban traffic state estimation by fusing multisource information," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 499–511, 2009.

[8] Philadelphia Department of Transportation, "Traffic Control Center," http://philadelphia.pahighways.com/philadelphia-atcc.html.

[9] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.

[10] J. Jaehoon, G. Shuo, G. Yu, H. Tian, and D. Du, "TBD: trajectory-based data forwarding for light-traffic vehicular networks," in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS '09)*, pp. 231–238, Montreal, Canada, June 2009.

[11] J. Jeong, S. Guo, Y. Gu, T. He, and D. H. C. Du, "TSF: trajectory-based statistical forwarding for infrastructure-to-vehicle data delivery in vehicular networks," in *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS '10)*, pp. 557–566, Genova, Italy, June 2010.

[12] ETSI, "DSRC_Standardization," http://www.etsi.org/WebSite/Technologies/DSRC.aspx.

[13] F. Xu, S. Guo, J. Jeong et al., "Utilizing shared vehicle trajectories for data forwarding in vehicular networks," in *Proceedings of the the 30th IEEE International Conference on Computer Communications (IEEE INFOCOM '11)*, pp. 441–445, April 2011.

[14] L. Chen, Z. Li, S. Jiang, and C. Feng, "MGF: mobile gateway based forwarding for infrastructure-to-vehicle data delivery in vehicular ad hoc networks," *Chinese Journal of Computers*, no. 3, pp. 454–463, 2012.

[15] M. Mabiala, A. Busson, and V. Vèque, "Inside VANET: hybrid network dimensioning and routing protocol comparison," in *Proceedings of IEEE 65th Vehicular Technology Conference (VTC '07)*, pp. 227–232, April 2007.

[16] H. Yomogita, "Mobile GPS Accelerates Chip Development," http://techon.nikkeibp.co.jp/article/HONSHI/20070424/131605/.

[17] D. Niculescu and B. Nath, "Trajectory based forwarding and its applications," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 260–272, September 2003.

[18] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '98)*, pp. 76–84, Dallas, Tex, USA, October 1998.

[19] M. MacDougall, *Simulating Computer Systems: Techniques and Tools*, MIT Press, Cambridge, Mass, USA, 1987.

[20] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.

[21] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: a framework to systematically analyze the Impact of Mobility on Performance of RouTing protocols for Ad hoc Networks.," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 825–835, San Franciso, Calif, USA, March 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 648695, 15 pages
http://dx.doi.org/10.1155/2013/648695

Research Article

Data Processing and Algorithm Analysis of Vehicle Path Planning Based on Wireless Sensor Network

Wenyuan Tao and Mingqin Chen

School of Software Engineering, Tianjin University, Tianjin 300072, China

Correspondence should be addressed to Wenyuan Tao; taowenyuan@tju.edu.cn

Received 11 January 2013; Accepted 9 March 2013

Academic Editor: Yan Zhang

Optimizing the path planning to reduce the time and cost is an essential consideration in modern society, and existing research has mostly concentrated on static path planning and real-time data information in vehicle navigational applications. Using dynamic path planning to adjust and update the path information in time is a challenging approach to reduce road congestion and traffic accidents. In this paper, we present a data analysis algorithm that determines an efficient dynamic path for vehicle repair-scrap sites and navigates more flexibly to avoid obstacles, where the key idea is to design the sensor wireless network that helps to obtain data from different devices. Firstly, the data processing scheme for real-time data with regional cluster and node division can be obtained from different sensor devices through the wireless senor network. Secondly, the search space and the relevant road information are restricted to a strongly connected graph. The most important strategy for an optimal solution to find the shortest path is the search method. Finally, to validate the performance of our design and algorithm, we have conducted a simulation based on necessary traffic variables. The performance simulation results show that real-time dynamic path planning can be significantly optimized using our data processing scheme.

1. Introduction

The acceleration of an automobile digitalization has led to the era of networked cars. Nowadays, China has become the largest automobile market in the world with an annual increase of 40% in car sales. There is no doubt that cars have brought convenience to our life; however, faulty cars could cause great troubles.

In order to make effective use of resources, governments at all levels in China have built sufficient repair-scrap sites to solve the problems for faulty cars. Receiving real-time information on the state of vehicles through wireless sensor network (WSN) [1, 2] and informing car owners of the closest repair-scrap sites are effective means to solve the problem of faulty cars and reduce potential traffic problems.

The key to solving this problem lies in the quantity and location of repair-scrap sites. To ensure the lowest overall costs, the average distance to the closest repair-scrap sites should be minimized. The collection and analysis of data and the issue of vehicle routing are crucial to solve this problem.

At present, automobiles mainly rely on vehicle navigation systems to receive satellite information and show the current position, driving direction, and distance from destinations on the electronic map. They choose the best driving route within the known road network range based on the shortest path principal. Currently, mature vehicle navigation systems in the market are mostly based on static path planning. When traffic accidents and road congestions occur, static path planning is unable to change routes in time. Thus, the vehicle dynamic path planning becomes a hotspot in research. With the rapid development of WSN, path search data is no longer static. Instead, data are collected, processed, and transferred through a network formed by a large number of sensor nodes distributed in a region in a self-organized manner. The sensor nodes have the characters of low-cost, low-power dissipation and the functions of perception, data processing, storage, and wireless communication [3]. With the change from static

data to dynamic data, real-time information of current traffic condition can be available. The accuracy of recommended paths increases, does so the pressure on processing data; and thus, the requirement for the algorithm of data processing is more stringent.

In the issue of path optimization, the computing time increases in proportion to the increase of nodes especially in large networks. In the path optimization of large urban networks, the number and weight of nodes and edges increase dramatically; thus, the needed calculating time and real-time renewal quantity are large and can greatly affect the calculating efficiency of path optimum algorithms.

Based on the historical traffic information, combined with the current traffic information, the vehicle dynamic path planning is able to predict future traffic flow. It can adjust and update the best driving path in time to reduce traffic accidents and road congestions effectively. In recent years, the traffic status prediction becomes increasingly important to the research of vehicle navigation.

According to the understanding of specific reality, local path planning can be divided into two types: one with completely unknown environmental information and the other one with partly unknown environmental information. With the advent of the Internet, traffic information, vehicle information, and path information can be collected through the wireless sensor network. Environmental information can usually be quickly obtained so as to make global positioning which facilitates site selection.

In this paper, we propose an optimal network design model that decides the site location among vehicle repair-scrap sites. The cost of the model is minimized, proportional to the distance. The most important step for site positioning is to determine the distance.

We assume WSN, through which a large number of original data can be obtained. Preprocessing the original data according to specific demands, we can obtain data in a standard format (without abnormal data) upon which data analysis and path matching can be conducted so as to find the best path. These methods can be applications of wireless sensor networks in home area networks [4] and smart grids [5].

Depending on the existing environment and communication technologies, our efficient path planning method avoids obstacles through rectangular decomposition [6], according to the graph theory. The path planning algorithm consists of two levels: global path planning and local path planning. As to the global path planning layer, we have collected faulty sites, the location of the candidate repair centers, and the candidate scrap fields based on wireless senor network and the planning of the government.

In accordance with the scale of candidate sites, the global regional environment will be divided into several equal units by the grid method and rectangular decomposition. Local path planning is the planning of the path between two sites. According to the start site and the end site of the local area, we can choose the optimal path around obstacles in order to obtain the shortest one. We abstract the network into a graph and then find the shortest path via real-time traffic network and traffic information.

2. Related Work

The necessity of solving vehicle routing problem convinces us to analyze carefully according to given schemes. The vehicle path information is collected through wireless sensor network; then after analyzing the data, we can find the shortest path to arrive at the destination. Thus, the previous work related to this paper consists of three main aspects: data collection through wireless sensor network, data processing by graph theory, and algorithm analysis for the shortest path.

Much work has been done on wireless sensor networks. Akyildiz et al. [7] provide a review of factors that influence the design of sensor networks. It is known that, wireless sensor networks use battery-operated computing and sensing devices. Ye et al. [8] propose S-MAC, a medium access control (MAC) protocol designed for wireless sensor networks. Mainwaring et al. [9] do the in-depth study of applying wireless sensor networks to real-world habitat monitoring and other researches on wireless sensor networks [10]. Al-Karaki and Kamal [11] work on routing techniques in wireless sensor networks and data aggregation in wireless sensor networks [12]. According to these works, wireless sensor networks should care more about how to process real-time data for vehicle navigation.

On graph theory, Bryant [13] presents a new data structure to represent the Boolean functions and an associated set of manipulation algorithms. Tarjan [14] proposes an improved version of an algorithm to find the strongly connected components of a directed graph, and an algorithm to find the disconnected components of an indirect graph is presented. Some methods have been developed using graph theory [15], graph structure [16], and graph search [17–20]. Graph theory being regarded as an effective analysis tool is applied to optimization of road networks, which simplifies the problem.

In order to provide efficient vehicle path planning, a dynamic path planning method which is suitable for domestic vehicle navigation applications based on researching the dynamic road network models including arithmetic and the real-time traffic information was presented [21]. An effective transport system using global information vehicle information and communication system (VICS) and local information and communication system (IVC) was introduced [22]. The VICS, as one of the intelligent transport systems, has been developed to reduce the traffic jam; it can give real-time global information to drivers to help them find the shortest path, although the shortest path has been studied for a long time. Floyd [23] and Johnson [24] have worked on the shortest path, and there are some other related studies [25].

It can be concluded that there are two key points in the work of this paper: exploiting the vehicle navigation applications to collect real-time data information and using proper routing algorithms to minimize the total cost. As is noted above, most of these solutions are unaware of the problem that source data is so "weak" that there is only one kind of data source from GPS. Thus, in this paper we design the sensor wireless network that can obtain data from different devices such as vehicle GPS, digital camera, speed

sensor, and direction sensor. As their routing algorithms cannot solve the problem of real-time path finding properly, we design a new routing algorithm based on the shortest path to deal with the real-time path finding problem.

3. Problem Definition

A minimum total cost for a repair-scrap network is needed to be built. The costs to be considered in designing repair-scrap network include construction cost and transportation cost. The following parameters and variables will be used in the proposed model:

$i \in I$ is a faulty site;

$j \in J$ is a candidate repair center;

$k \in K$ is a candidate scrap field;

$m \in M$ is the capacity of a repair center;

$h \in H$ is the capacity of a scrap field;

a_i is the number of faulty cars at faulty site i;

ρ_i is the proportion of faulty cars at faulty site i that is needed to be sent to repair centers;

E is the proportion of faulty cars that is needed to be scrapped after going through repair centers;

d_{ij} is the distance of transporting a faulty car to repair center j from faulty site i;

d_{ik} is the distance of transporting a faulty car to scrap field k from faulty site i;

d_{jk} is the distance of transporting a faulty car to scrap field k from repair center j;

C^m is the cost of building a repair center of capacity m;

C^h is the cost of building a scrap field of capacity h;

Θ the proportional coefficient.

The definitions of decision variables are as follows:

$$X_j^m = \begin{cases} 1 & \text{Build a repair center of capacity } m \\ & \text{if in candidate site } j \\ 0 & \text{OR,} \end{cases}$$

$$Y_k^h = \begin{cases} 1 & \text{Build a scrap field of capacity } h \\ & \text{if in candidate site } k \\ 0 & \text{OR,} \end{cases}$$

$$Z_{ij} = \begin{cases} 1 & \text{If faulty cars in faulty site } i \\ & \text{are transported to repair center } j \\ 0 & \text{OR,} \end{cases} \quad (1)$$

$$M_{ik} = \begin{cases} 1 & \text{If faulty cars in faulty site} \\ & i \text{ are transported to scrap field } k \\ 0 & \text{OR,} \end{cases}$$

N_{jk} is the number of faulty cars being transported to scrap field k from repair center j.

Considering the cost of building repair centers and scrap fields and the cost of transportation, we build the following optimization model according to the features of a candidate location within the vehicle network.

The target function is

$$\min C = \sum_m \sum_j C^m X_j^m + \sum_h \sum_k C^h Y_k^h + \sum_i \sum_j a_i \rho_i \theta Z_{ij} d_{ij} + \sum_i \sum_k a_i \theta (1 - \rho_i) M_{ik} d_{ik} + \sum_j \sum_k \theta N_{jk} d_{jk}. \quad (2)$$

Formula (2) represents the minimum total cost, including the cost of building repair centers and scrap fields as well as the cost of transportation.

The following conditions should be satisfied for formula (2):

$$\sum_j Z_{ij} = 1 \quad \forall\, i \in J, \quad (3)$$

where Z_{ij} represents whether faulty car i is transported to repair center j. Formula (3) ensures that all faulty cars in all faulty sites can only be dealt with by one repair center:

$$\sum_i a_i \rho_i Z_{ij} \le \sum_m m X_j^m, \quad (4)$$

where $\sum_m m X_j^m$ represents the total capacity of repair center j, and $\sum_i a_i \rho_i Z_{ij}$ represents the sum of all the faulty cars in repair center j. Formula (4) ensures that the number of faulty cars transported to a repair center cannot exceed its capacity:

$$\sum_i a_i (1 - \rho_i) M_{ik} + \sum_j \varepsilon T_j N_{jk} \le \sum_h m Y_k^h \quad \forall k \in K, \quad (5)$$

where $\sum_h m Y_k^h$ represents the capacity of scrap fields k. $\sum_i a_i(1 - \rho_i) M_{ik}$ represents the total number of scraped cars to scrap fields k from all faulty sites, and $\sum_j \varepsilon T_j N_{jk}$ represents the total number of scrapped cars to scrap fields k from all repair centers. Formula (5) represents the number of faulty cars transported to scrap fields as well as scrapped cars which cannot exceed its capacity:

$$\sum_i M_{ik} = 1 \quad \forall k \in K, \quad (6)$$

where M_{ik} represents whether faulty car i is transported to scrap field k. Formula (6) ensures that all faulty cars in all faulty sites can only be dealt with by one scrap field:

$$\varepsilon \sum_i a_i \rho_i Z_{ij} = \sum_k N_{jk} \quad \forall j \in J, \quad (7)$$

where $\sum_k N_{jk}$ represents the total number of crapped cars from repair center to scrap field j, and $\varepsilon \sum_i a_i \rho_i Z_{ij}$ represents the total number of cars from all repair centers to scrap field j. Formula (7) represents the flowing equilibrium relationship between faulty cars into repair centers and scrapped cars out of scrap fields:

$$\sum_m X_j^m \le 1 \quad \forall j \le J, \quad (8)$$

$$\sum_h Y_k^h \le 1 \quad \forall k \in K. \quad (9)$$

Formula (8) represents one candidate that site that should be built as a repair center, respectively. Formula (9) represents one candidate site that should be built as a scrap field, respectively:

$$\begin{gathered} X_j^m \in (0,1), \qquad Y_k^h \in (0,1), \\ \forall j \in J, \quad \forall m \in M, \quad \forall h \in H, \quad \forall k \in K, \\ Z_{ij} \in (0,1), \qquad M_{ik} \in (0,1), \\ \forall i \in I, \quad \forall j \in J, \quad \forall k \in K, \\ M_{jk} \geq 0 \quad \forall j \in J, \quad \forall k \in K. \end{gathered} \tag{10}$$

Formula (10) defines the constraints of decision variables X_j^m, Y_k^h, Z_{ij}, and M_{ik}.

This paper focuses mainly on the issue of vehicle path, as well as analyzing path d in mathematical model.

Since the objective of sites location is to find the shortest path between the two given sites in predetermined workspace, we derive a minimum total cost of a repair-scrap network for the given environmental structure that involves a number of faulty sites and a number of obstacles. In this paper, the algorithm procedure is described by rectangular decomposition, the strong connection graph, and site-to-site algorithm.

The costs to be considered include the construction cost and transportation cost, both of which are proportional to the distance. Meanwhile, the degree of crowdedness and the level of unblock for the transport path also affect path selection. According to the subjective intention of a person, you can choose the path if you want to go by the two constraints. This definition shows that the distance and the two constraints are important for the proposed global model.

4. Data Processing Scheme

Needed data are collected through wireless sensor network. Constituted by numerous cheap microsensors nodes deposited in the monitoring field, the mobile sensor network is a self-organized multihop network system. The sensor nodes have the characters of low-cost, low-power dissipation and the functions of perception, data processing, storage, and wireless communication. Data are collected through gathering, processing, and transferring the perceptive object in the area covered by network corporately.

Perceptive objects, sensor nodes, and users are three main elements of WSN. The monitoring area is the possible appearing area of the perceptive objects; perceptive objects are affairs that attract users and vary with the change of scenes. In the shortest path planning for vehicle repair-scrap sites, the perceptive objects are possible spots of faulty cars within the monitoring field. WSN usually includes sensor node, convergent node, and task management node. The convergent node has quite strong data processing and power supply ability. Normally there are one or several convergent nodes in a WSN, and a large number of common sensor nodes are distributed in the sensing field in the random or defined deployment. The sensor node is responsible for collecting information related to affairs that people are interested in, returning data along some path in the way of multihop, and finally transmitting data to task management node by the convergent node through Internet or data network with satellite. Inside sensor network, every node is not only the collector and sender of information, but also the router. Cooperating with each other, each node transmits sense data to the convergent node through multihop routing in the way of relay.

As data collection is the foundation of a research, we design the following data collection framework to obtain specific and accurate roads information.

The platform processing frame diagram (as is shown in Figure 1) consists of data collecting, data preprocessing, data analysis, and path recommendation. By gathering data through sensors, original data are obtained; after preprocessing the specific features of original data, we can get data that can be used directly for the experiment; by using algorithm of high efficiency, network data analysis and path recommendation can be realized.

To guarantee the efficiency of wireless transmission, it is inappropriate to send one datum at one time, as the sampling of the physical quantity of each node in WSN is conducted individually. Meanwhile, because of the continuity of time data, road network is divided into several sections in the design of WSN to collect original data separately. In each collection area three nodes with different functions (data node, convergent node, and task management node) are used instead of using just data node. In this way, the pressure on data transmission can be greatly reduced by simple data fusion.

As is shown in Figure 2, road network is divided into N regions, with each region consisting of 3 different nodes (data node, convergent node, and task management node). After the original data are obtained from these nodes, they will be stored in the corresponding database of each region for future research. The data obtained from sensor nodes cannot be directly used in our experiment; thus, these data should be preprocessed into the data documentation format required by specific experiments in the data procession server of each region according to different experimental requirements. After the preparation of experimental data, we search the road network in the large server according to the algorithm of this paper to recommend the best path.

According to the features of WSN, we fractionize the examination area, or divide it based on the users' choice. Perceptive objects are crossings or paths that vehicles will pass. The examination area is the possible appearing place of perceptive object (from start to target place). Sensor node, convergent node, and task management node in the examination area upload real-time crossing information (such as no-left-turn or no-right-turn crossings, one-way street sections, and time periods in which the traffic of specific type of vehicle is closed) to Internet or satellite network, plan route with proposed algorithms, and then send back perceptive objects through convergent nodes for visual display.

Wireless sensor node and roadside access point are usually included in vehicle-based WSN in which the roadside

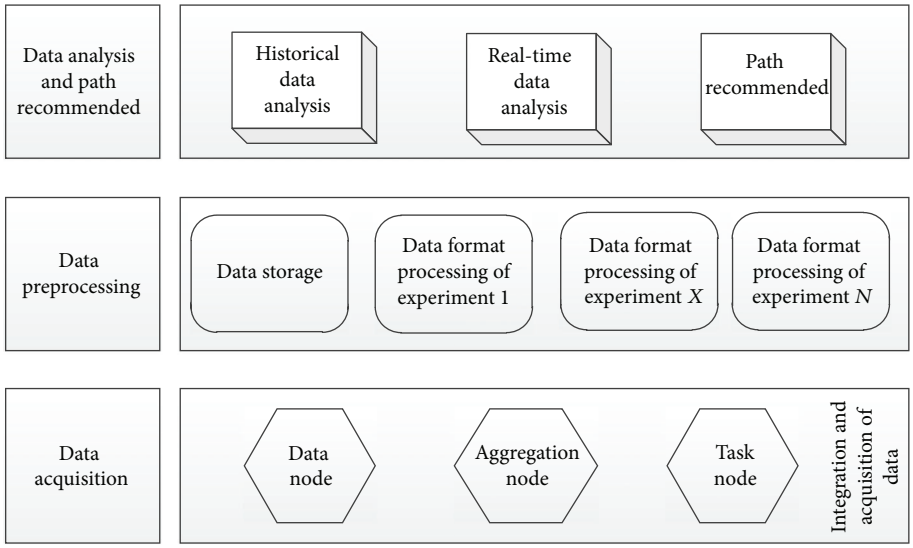


Figure 1: Frame diagram of data acquisition.

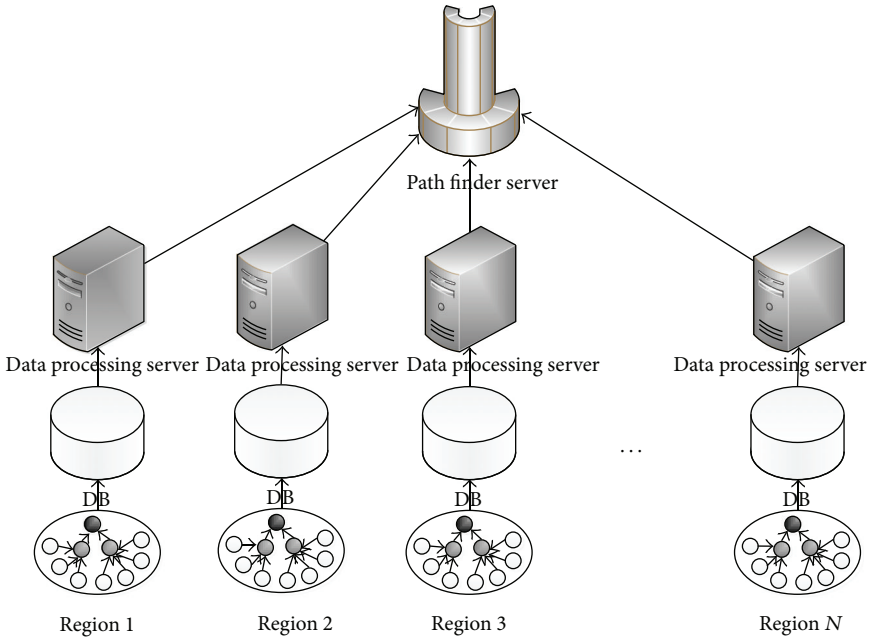


Figure 2: Schematic diagram of data acquisition.

access point plays two roles: convergent node and gateway to realize the interflow between WSN and backbone network. In the planning of the vehicle repair-scrap sites positioning, nodes in the network are vehicles loaded with road condition perception equipment. During the driving process, vehicles can collect samples of congestion and road condition along the way. At the same time, vehicles become the consumers of information, which requires the drivers to understand not only the current road condition but also road condition ahead so as to choose the best path. Roadside access point needed for the communication between related departments and vehicles also requires relaying support from sense data.

In this paper, data information is collected through WSN. Real-time information of traffic condition can be available by installing sensors like radar, camera, GPS navigator, and aerial detection in vehicles to monitor driving conditions, exhaust emission, and real-time road condition timely. And it will also bring convenience to drivers in making driving schedules.

5. Algorithm Analysis

Based on the data collected through WSN, we will design an algorithm for path planning by analyzing the established data model.

The problem to be considered is the shortest path planning for vehicle repair-scrap sites. In this paper, we restrict the geographical environment space to the strong connection

graph. According to the sites and the obstacles, each site can be considered as a point whose coordinate is the site left-edge or right-edge center's coordinate. Once all points are scanned, all sites are covered. The core of the algorithm is to determine the strong connection graph in the environment firstly, mark the node based on the relationship between sites secondly, and find the shortest paths by traversing the nodes involved finally.

The planning of the vehicle repair-scrap sites positioning mainly contains four different types of sites: repair center, scrap field, faulty site, and obstacles (e.g., river, park). According to the relationship among the nodes, we need to determine the shortest path of each faulty site to every repair center and to every scrap field. Based on all the identified shortest paths, we can determine the optimal planning for the sites.

5.1. An Environment Model. Based on the wireless senor network and the government's plan, we obtain a detailed outline of the geographical environment. The given information helps us to determine the site's position in the given environment. Then, we can mark the faulty sites, repair centers and scrap fields, and obstacles with different symbols based on their characteristics in order to establish a map of the environment as is shown in Figure 3.

5.2. A Rectangular Model. In this paper, the known grid map is used to build each site and obstacle into a rectangular model as is shown in Figure 4. All sites and obstacles are divided into various rectangles with different length and width. The definite means are as follows:

(1) find the minimum x value for each site, taking all obstacles into consideration. If more than one grid point has the minimum x value, then find the minimum y value for the grid point and mark it as $M(x_1, y_1)$;

(2) find the maximum x value for each site and obstacle. If more than one grid point has the maximum x value, then find the maximum y value for the grid point and mark it as $N(x_2, y_2)$;

(3) each site or obstacle is represented as a rectangle whose two diagonal end points are defined by the M and N.

5.3. Modeling the Paths around Obstacles. Firstly, we define the concept of the line segments and the nodes for a strong connection graph.

Let R be an $m * n$ uniform grid graph that consists of a set of grid nodes $\{(x, y) \mid x \text{ and } y \text{ are integers such that } 1 \leq x \leq n, 1 \leq y \leq m\}$ and grid edges connecting grid nodes. The length of grid edges connecting adjacent nodes in R is assumed to be 1.

Let $S = \{S_1, S_2, \ldots, S_p\}$ be a set of mutually disjoint sites with boundaries on R. Each rectangle in S is a site.

Let $B = \{B_1, B_2, \ldots, B_q\}$ be a set of mutually disjoint obstacles with boundaries on R. Each rectangle in B is an obstacle.

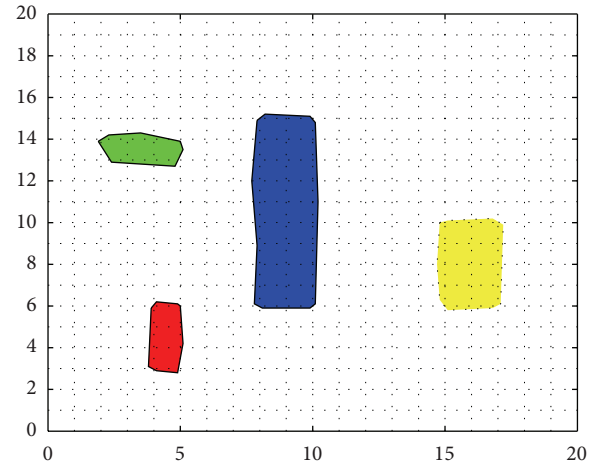


Figure 3: Partially geographical environment.

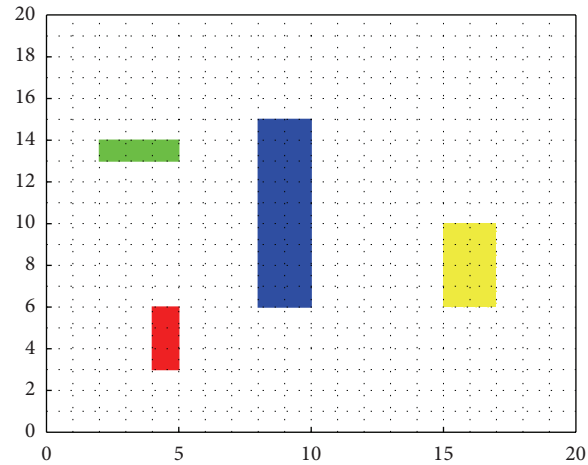


Figure 4: A rectangular model and sites identity.

Let G denote a partial grid of R that consists of grid nodes which are not contained in the interior of any obstacle in B or any site in S and grid edges that are not incident to interior grid nodes of any obstacle in B or any site in S as is shown in Figure 5.

5.3.1. Line Segments. Each obstacle is represented as a rectangle. In order to get the shortest path from the start point to the target point, we need to construct a connection graph G_c. The first step is to determine line segment L, as finding line segments is an important step.

Each site or obstacle rectangle has four vertices that are corresponding to the four vertices located in the upper left corner, the upper right corner, the lower right corner, and the lower left corner of the site. Let $V = \{V_1, V_2, \ldots, V_p\}$ be a vertex set which consists of four vertices of each site and obstacle, start point and target point on R. In this paper, we define that the start point is one of four vertices, and s is closer to the target point t than others. Meanwhile, the target point t is one of four vertices and t is closer to the start point s

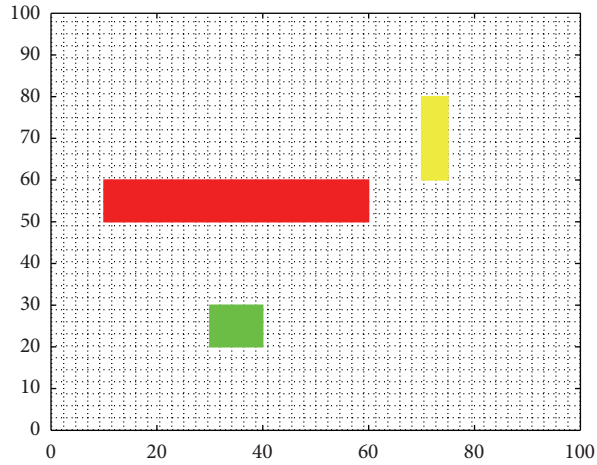


Figure 5: An environment model.

than others we have decided. The line segment L is decided according to the algorithm. Figure 6 is the diagram of line segments.

Step 1. In the environment map R, let $L(R)$ be the set of line segments that form the boundaries of R.

Step 2. We find any vertex $V_i(x, y)$ from vertex set V and draw the line segment in parallel with the y-axis. According to the x-axis coordinate, the sites in the space are sorted. If x is between the abscissa of the sites, and the y-axis parallel line through the point A intersects with the site, then the site is marked. And if x is not between the abscissa of the sites, let the site be marked as 0.

Step 3. Let the line segment be in parallel with y-axis. If it is marked as 0, the line segment from $[x, 0]$ to $[x, 100]$ will be completed through A and is parallel to the y-axis. If it is marked, we need to identify the segment of the upper endpoint and the lower endpoint. Let ordinate $y1$ of the upper endpoint $(x, y1)$ be the minimum value of the y-coordinate by upward line parallel to the y-axis and through the point A when the first encounters a site. Let ordinate $y2$ of the lower endpoint $(x, y2)$ be the maximum value of the y-coordinate by downward line parallel to the y-axis and through the point A when the first encounters a site.

Step 4. We find any vertex $V_i(x, y)$ from vertex set V, and draw the line segment in parallel with the x-axis. According to the y-axis coordinate, the sites in the space are sorted. If y is between the ordinate of the sites, the x-axis parallel line through the point A intersects with the site, and the site is marked. If y is not between the ordinate of the sites, let the site be marked as 0.

Step 5. Let the line segment be in parallel with x-axis. If it is marked as 0, the line segment from $[0, y]$ to $[100, y]$ can be completed through A and parallel to the x-axis. If it is marked, we need to identify the segment of the left endpoint and the right endpoint. Let ordinate $x1$ of the left endpoint $(x1, y)$ be the maximum value of the x-coordinate by left line parallel to the x-axis and through the point A when the first encounters a site. Let ordinate $x2$ of the right endpoint $(x2, y)$ be the minimum value of the x-coordinate by right line parallel to the x-axis and through the point A when the first encounters a site.

Step 6. If the vertex set V is null, the line segment search is completed. The line segment set is as follows $L = L(R)+L(X)+L(Y)$; Otherwise, return to Step 2.

Based on the above algorithm, we can decide the final line segments. For example, the line segments are shown in Figure 7.

5.3.2. Connection Graph. After finding the set of edges, we need to search the intersection point between the edges. Let these intersection points, start point, and target point be composed of a set of nodes. The pseudocode of generate connection graph is shown in Algorithm 1.

At first, it is necessary to find the relationship between these nodes in the set, for example, the intersection point which is closed to the next intersection point in the direction of X-axis and Y-axis. Through these nodes and the relations between these nodes, we abstract the whole environment into a strong connected graph as is shown in Figure 8.

5.3.3. The Definition of the Shortest Path. In the paper, the start point s and the target point t are faint. We define the Manhattan distance $M(s,t)$ between s and t. Let Q be any obstacle-avoiding path from s to t. The detour number $d(Q)$ is defined as the total number of nodes on Q that are directed away from t. The length of Q is $M(s,t) + 2d(Q)$. Q is a shortest path if and only if $d(Q)$ is minimized among all paths connecting s and t.

G is a strong connection graph for s and t. According to the coordinates of the corner points of R and obstacles in B, and a given pair of the start and target points, G can be used to find the optimal path. Let s, t be the start and target point, respectively, and v, u be any point in the shortest path, respectively. A direction assigned to an edge (u, v) and the direction is from u to v. We define the detour length of $(u \rightarrow v)$ with respect to target node t denoted by $du\,(u \rightarrow v)$ as shown in Figure 9. Let l be the line passing through t and perpendicular to $u \rightarrow v$.

$$du\ (u \longrightarrow v) = \begin{cases} 0 & \text{if } u \text{ and } v \text{ are on the same side of } l \\ & \text{and } u \text{ is further from } l \text{ than } v; \\ \text{the length of } u \longrightarrow v & \text{if } u \text{ and } v \text{ are on the same side of } l \\ & \text{and } u \text{ is closer to } l \text{ than } v; \\ \text{the length of } w \longrightarrow v & \text{if } l \text{ intersects } u \longrightarrow v \text{ at } w. \end{cases} \tag{11}$$

The detour length of a node u with respect to a source node s and a target node t, denoted by $\sigma(u)$, is the sum of the detour lengths of all directed edges in any directed shortest

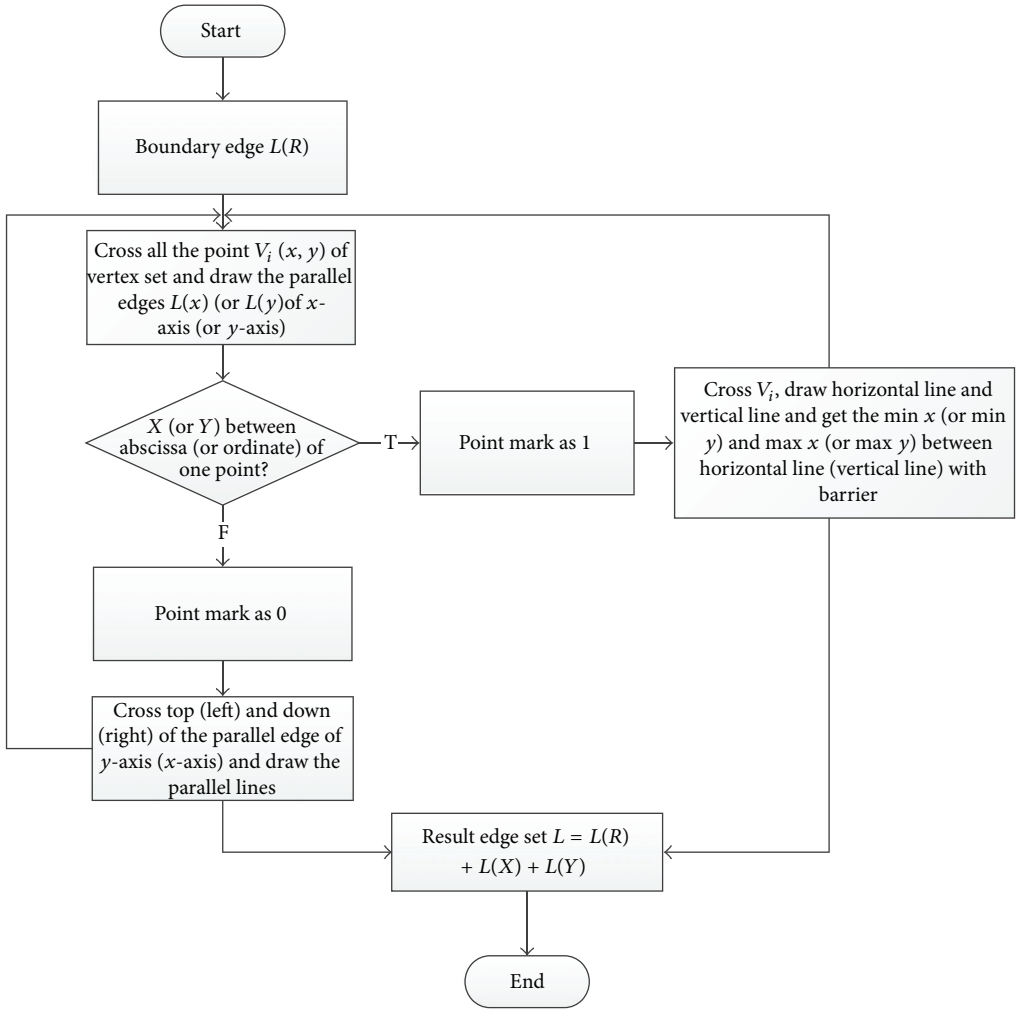


Figure 6: The flow diagram of line segments search.

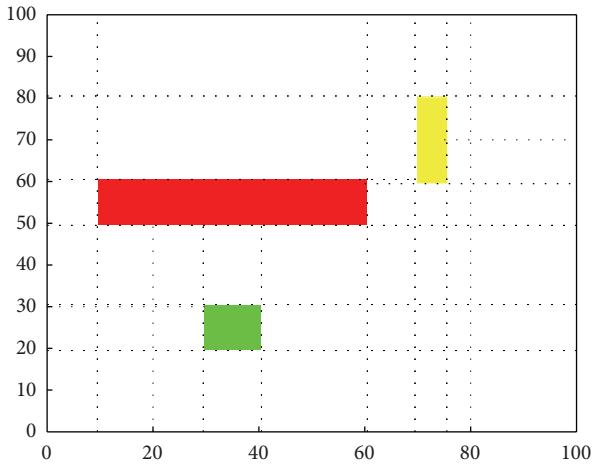


Figure 7: Search the line segments.

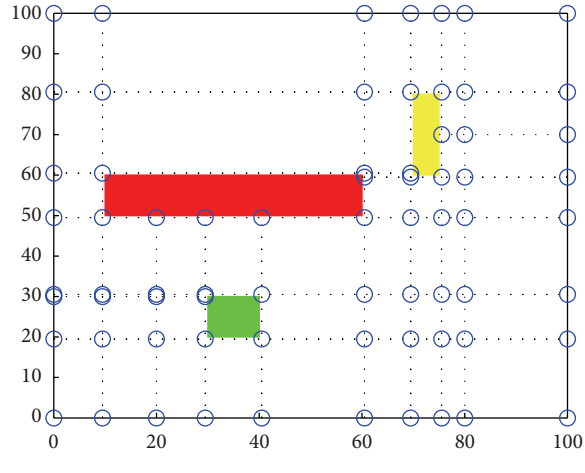


Figure 8: The strong connected graph.

path from s to u in G. Let P be a shortest path from s to t in G. The length of P is equal to $M(s,t) + 2\sigma(t)$.

5.3.4. The Shortest Path Algorithm. After having a clear definition of the shortest path, we can begin to search the

```
lineRec X add edge of figure
lineRec Y add edge of figure
lineRec X add DrawSegment(StartPoint) . Xsegment
lineRec X add DrawSegment(TargetPoint) . Xsegment
lineRec Y add DrawSegment(StartPoint) . Ysegment
lineRec Y add DrawSegment(TargetPoint) . Ysegment
for all barrier in BarrierPosition
    for four conners of barrier
        lineRec X add DrawSegment(conner) . Xsegment
        lineRec Y add DrawSegment(conner) . Ysegment
    end
end
for all segment Xsegment in lineRec X
    for all segment Ysegment in lineRec Y
        crossPointRec add polyxpoly(Xsegment, Ysegment)
    end
end
sortrows crossPointRec //sort the Point, and get all Points finished

//the important function—DrawSegment
function DrawSegment
    sort BarrierPosition by X
    get minY and maxY between x = X · x and all BarrierPosition
    lineSegment X = [x, x, minY, maxY]

    sort BarrierPosition by Y
    get minX and maxX between y = Y · y and all BarrierPosition
    lineSegment Y = [minX, maxX, y, y]
    return lineSegment X and lineSegment Y
end

// The initialization of graph
for all Point point x in crossPointRec
    for four direction of point x
        get the nearest Point and draw line
    end
end
```

Algorithm 1: The initialization of the graph.

shortest path from the start point to the target point with the space. In this section, the shortest path algorithm of site-to-site will be introduced pseudocode is shown in Algorithm 2.

From the start point s, the algorithm generates graph G by points and edges, the points in the graph G are divided into two point subsets named as *VISITED* and *CANDIDATES*. Assume DU is the shortest path from the start point s to any arbitrary point (set u as the point name) in the connected graph G.

In the space, the way shown in Figure 10 to find out the shortest path is as follows.

Step 1. First, assume that the subset $VISITED = \emptyset$ and the set of points defined as *CANDIDATES* that wait to be accessed are empty, which means that all points have not been accessed by default. Then, the DU value of all points is initialized to be infinite, and the parent point is empty by default.

Step 2. Add point s into the subset of *VISITED* and claim that point s has been accessed. Then, choose the first waiting point as the current point, which belongs to the subset of CANDIDATES and is nearest to point s.

Step 3. Access the four "Next Points" that are near to the current point in turn and calculate the DU value between s and each one of the four points. If the sum of the value of current point and the value calculated from the current point to Next Point is below the value of "Next Point," update the value of "Next Point" as well as the parent point; otherwise, nothing is done.

Step 4. Set the first point in the subset of *CANDIDATES* as Next Point, update the subset of *VISITED*, add the Next Point to the subset of *VISITED*, mark the Next Point as accessed, delete the Next Point in the subset of *CANDIDATES*, and add the unaccessed points near the Next Point to the subset of *CANDIDATES*.

Step 5. Search in the *CANDIDATES* to continue finding the "Next Point" waiting to be accessed skip to Step 3 to continue. Therefore, this loop will not end until the current point reaches the last node in the *CANDIDATES* and the DU value of each node has been calculated. The next step is to choose a path from the start point s to the target point t.

```
init VISITED set, CANDIDATES set, Point i(0-length)
add NowPoint into CANDIDATES set
while CANDIDATES set not empty
    add NowPoint into VISITED set
    del NowPoint from CANDIDATES set
    NowPoint.visited = true
    for four directions of NowPoint
        NextPoint = NowPoint.direction
        if NextPoint.visited = false
            add NextPoint into CANDIDATES set
            if DU(NowPoint) + dis(NowPoint,NextPoint) < DU(NextPoint)
                DU(NextPoint) = DU(NowPoint) + dis(NowPoint,NextPoint)
                Parent(NextPoint) = NowPoint
            end
        end
    end
    if CANDIDATES set is empty
        break
    end
    set first point of CANDIDATES set as NowPoint
End
// Inverted Sequence
set TargetPoint as NowPoint
NextPoint = Parent(NowPoint)
while NextPoint not null
    Draw(NowPoint,NextPoint)
    NowPoint = NextPoint
    NextPoint = Parent(NowPoint)
end
```

Algorithm 2: The shortest path algorithm.

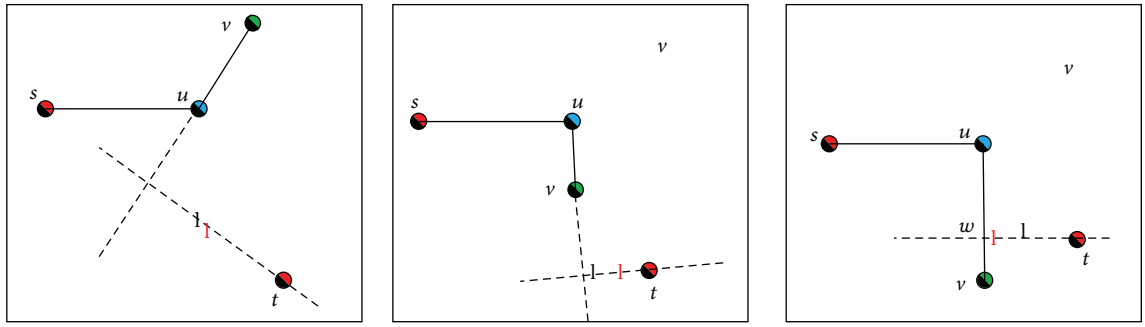


Figure 9: The definition of edge.

Step 6. Setting the target point as the current point, we can find the parent point of the target point (denoted as the "Next Point") in order to connect the current point with the "Next Point."

Step 7. The "Next Point" is denoted as the current point and then skips to Step 6 to continue execution. The cycle will not end until the "Next Point" meets the start point.

The simple example is shown in Figures 11, 12, and 13.

5.3.5. Congestion Weight Variables. In the above algorithmic model, we take DU, the distance between the start point s to the target point t, into account when choosing the shortest path. Then the shortest path between points is decided when the DU comes to its minimum.

As regarded to the practical urban traffic, the road congestion and time-phrased differences of one-way or two-way streets should be estimated. We express congestion as W (weight), the shortest distance between points as DU, and the degree of priority as a%, which leads to the choice of the

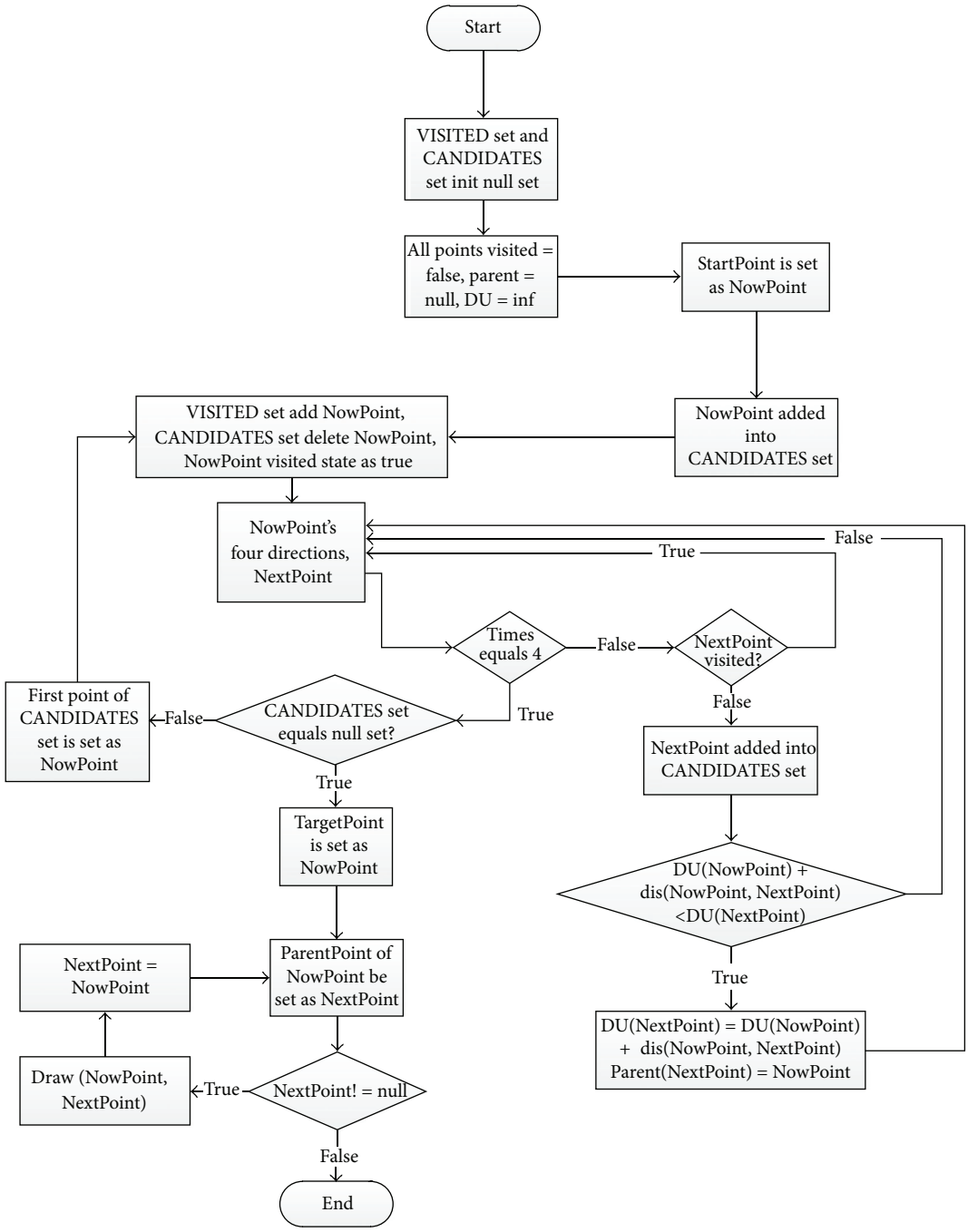


Figure 10: The flow diagram of the shortest path algorithm.

shortest path as follows: $L = a\% \times \mathrm{DU} + (1 - a\%)W$. DU and W are equal magnitudes.

If $a = 50$, it means that the congestion weight is as important as distance, and they have the same degree of priority. (1) If the two distances are the same, we can tell from L that the best passing path is when the congestion weights are at its minimum (Figure 14). (2) If the two congestion weight are the same, we can reach the best passing path from L when the distance DU is at its minimum. (3) If both the distance DU and the congestion weight are different, then through the calculation of L, the smallest minimum L is the best path.

According to Figure 14, the distances from the start point s to the target point t are the same through A, B, or C. Thus, the path can be decided just by comparing the congestion weight W of these three paths, $W_A = 128$, $W_B = 120$, and $W_C = 136$, so the best travel path is path A, the red one.

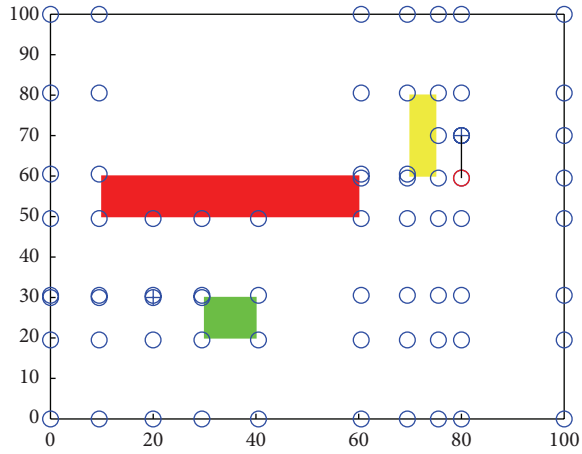


Figure 11: From the start point to search "Next Point."

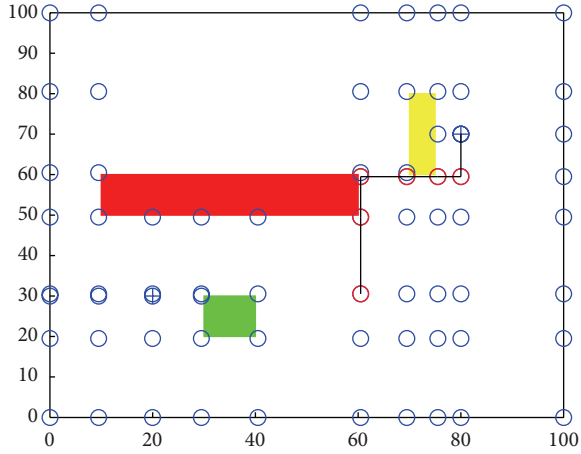


Figure 12: The shortest path.

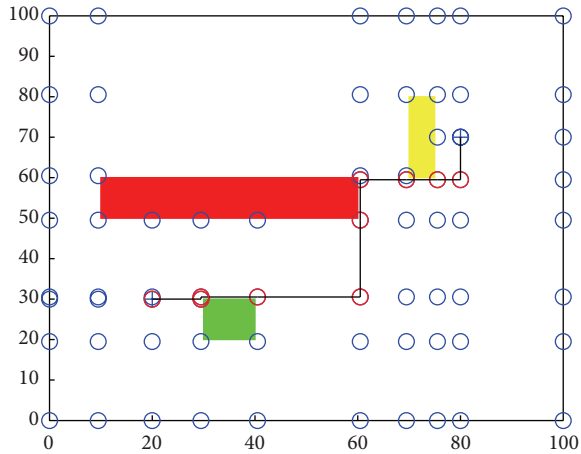


Figure 13: Search the shortest path.

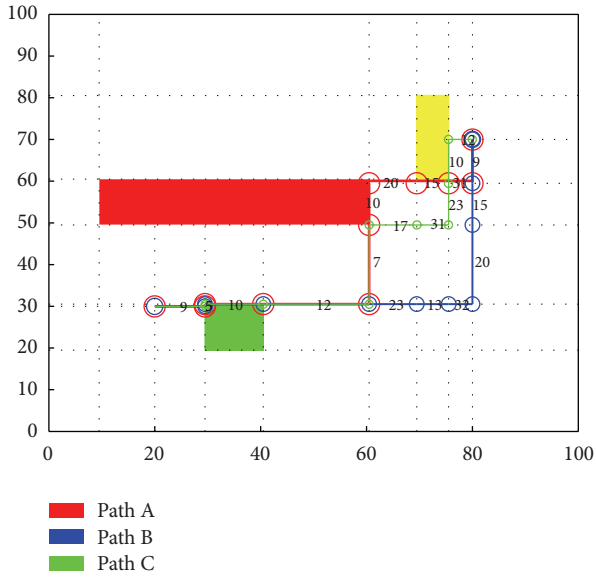


Figure 14: Best path chosen when congestion weight is as important as distance.

If $a > 50$, the distance DU is more important than the congestion weight. Then, under that condition the driver does not take time as the first priority and is not willing to burn oil for further trip; then, he can chose to wait and stick to the shorter path, the smallest minimum of L.

Figure 15 shows the highest priority of distance DU between the start point s to the target point t through path A, B, and C. Therefore, the best path can be decided by comparing the distance DU of these three paths and then calculating the minimum L. In the given example, the distance is the smallest one, so the red path A is the best travel path.

If $a < 50$, it refers that congestion weight W is more important than distance DU, which means that the driver takes time as the first priority and is willing to steer clearly of the congestion road to avoid time consumption; then, the smallest minimum L is the best path.

According to Figure 16, the highest priority is the weight W from the start point s to the target point through path A, B, and C. As a result, comparing the weight W of these three paths and then calculating the smallest minimum L will lead to the best path. In this given example, $W_A = 128$, $W_B = 120$, and $W_C = 136$, the blue path B is the best one.

5.3.6. Variables of One-Way or Two-Way Road. There are many one-way roads and two-way roads in urban traffic. One-way road means that vehicles travel from entrance on one side of the road and can only exit from the other side of the road, and usually there is a sign illustrating no entrance to exit; whereas two-way road refers to the road that you can enter and exit from both sides of the road freely, usually it has two lanes, four lanes, or six lanes. Some two-way roads are timed to remit the traffic pressure. In bustling parts of the city or locations that are easily congested, one-way roads and two-way roads are often restricted by time period.

As is presented in Figure 17, the red path is the shortest path in this environmental space with the assumption that

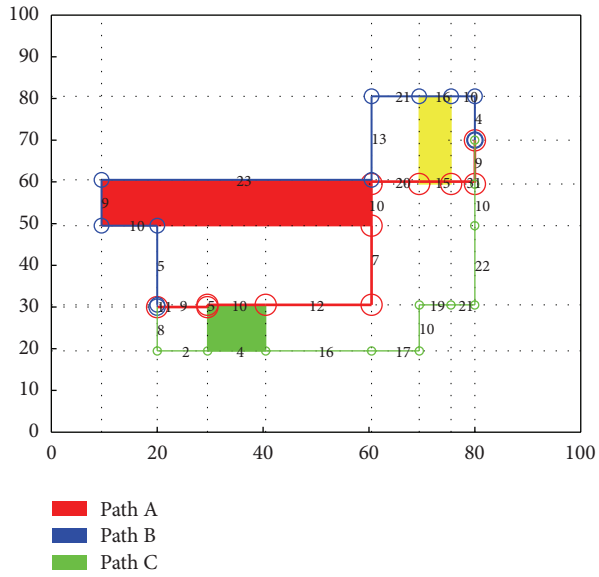


Figure 15: Best path choosing when distance is more important.

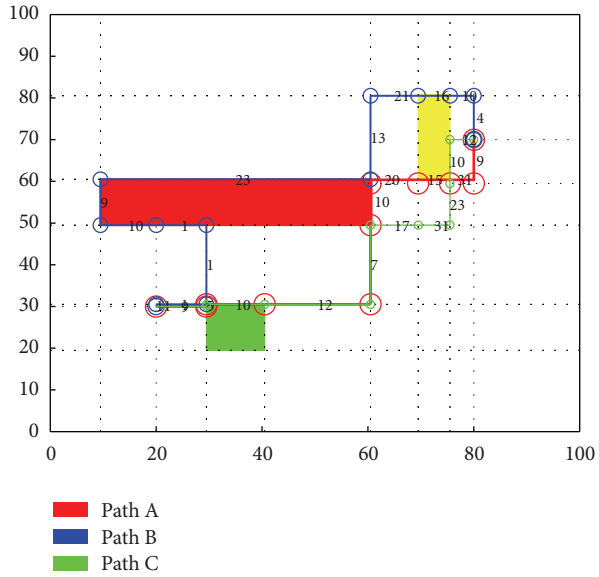


Figure 16: Best path chosen when congestion weight is more important.

all paths are passable. However, in reality, many paths are changing between one-way road and two-way road by times. For instance, from 7:00 am to 9:00 am, and from 5:00 pm to 7:00 pm roads in front of hospitals (yellow building) are one-way roads, restricting motor vehicles except buses to travel from east to west. As a result, during these two periods, the blue path is the best path for motor passengers.

6. A Simulation Study

In order to validate the arrangement of WSN, the necessity of preprocessing, and the correctness of path search algorithm in this paper the following experiment is designed.

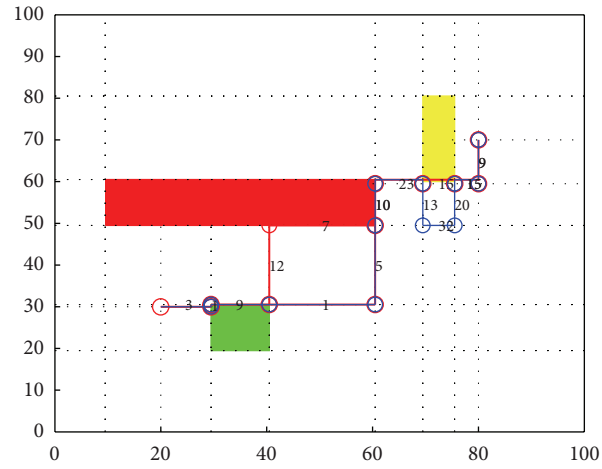


Figure 17: Best path choosing in one-way road and two-way road.

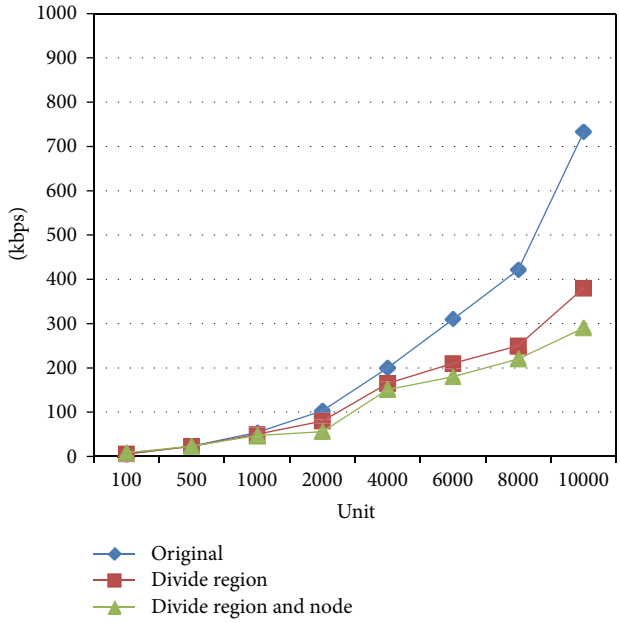


Figure 18: Comparison between three WSN.

Figure 18 compares the use of bandwidths among three WSN arrangements (original, divide region, and divide region and node) that vary with the number of sensors.

From the experimental results, we can see that the bandwidth usages of the WSN arrangements are almost the same when the number of sensors is small. The bandwidth of original WSN arrangement is rather small; however, it increases sharply with the increase of sensors. The bandwidth of divide region WSN arrangement increases slowly, while for divide region and node arrangement, it increases at the slowest speed. Thus, we can see that the WSN designed in this paper can be well adapted to the transportation field using a large number of sensors.

Figure 19 shows a comparison diagram of path-searching time by using original data and preprocessed data. The

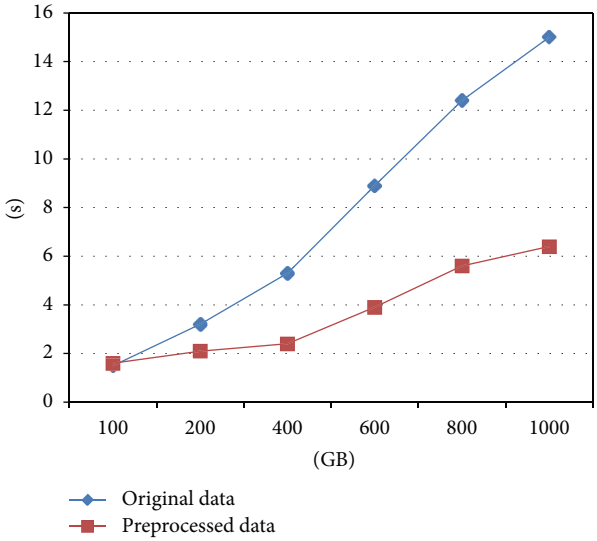


Figure 19: Comparison of path-searching times when using the original data versus preprocessed data.

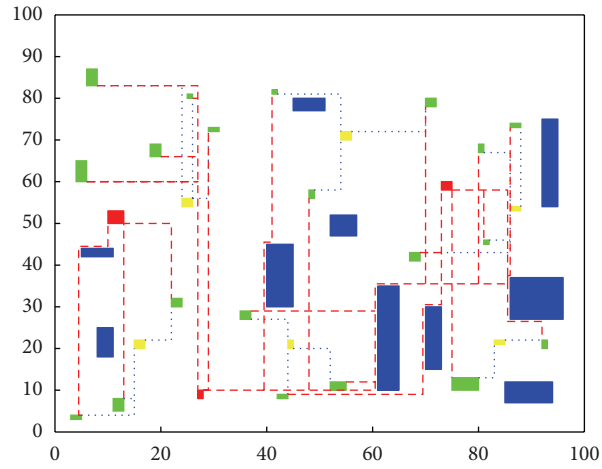


Figure 20: Result of vehicle routing problem.

horizontal axis represents the size of data, which ranges from 200 GB to 1000 GB. The vertical axis represents time spending, which is measured in seconds.

It is obvious that using original data to search paths takes longer time than using preprocessed data. Path-searching time of original data is more than 10 seconds, which seriously affects users' experience. The preprocessed data takes less than 6 seconds in path finding which can satisfy users' demand. So preprocessed data is proved to be a right measure since path finding efficiency as well as users' experience is greatly improved in this way.

Based on divide region and node WSN arrangement, and the preprocessing of original data, we use the algorithm to find path.

Through the practical environment of traffic network and real-time traffic, we abstract the environmental space into graphic. Then, by using the algorithm of the shortest path between points in the graph, we generate an example to test the validity of this algorithm. In this example, repair centers, scrap fields, faulty sites, and obstacles are generated from sensor devices. 20 places of faulty sites and their coordinates, quantity as well as the proportion of the failed vehicles going to repair centers or scrap fields are also generated from the sensor devices. We now assume that there are 6 repair centers and 3 scrap fields with the given coordinates, and the capacities of repair centers and scrap fields are different. Based on the above algorithm for the shortest path between points, we use MATLAB 7.0 to complete some simulated experiments. Since the algorithm is random in some way, we run the example ten times and come up with the shortest path between points that avoid obstacles, as is shown in Figure 20.

Figure 20 shows that if the urban traffic network was taken into account and the path already settled, vehicles have to follow the rules of the path. There are many paths between two points. The start point and the target point have been defined by WSN, for example, the green node and red node. According to Section 5, the algorithm cares only for the shortest path between the two points rather than the start point or target point. We use the shortest path algorithm between points in accordance with transport expanse, length of path, the capacity of the points, and the degree of congestion of the roads to get the best transport path, which is illustrated in the graph. The result of the calculation presents that, on the basis of the urban traffic network, we can analyze and generate the best path information according to the information of point capacity and traffic condition gathered through the wireless sensor network. Then, drivers can choose a travel path from these best path information. If the algorithm accuracy of the solution is ensured, it will assist drivers to a great extent. Meanwhile, it is also applicable and easy to popularize.

7. Conclusion

Finding the best path that takes the shortest time is the key to a path recommendation system. This paper has proposed an algorithm based on WSN to solve the problem.

The data used in this paper are self-defined and collected through WSN. Since there are a large number of sensors in WSN, the paper proposes the following design to reduce the pressure on network bandwidth: divide the road network into several regions, while each of them has an independent server for data preprocessing to reduce pressure on the path search server. Meanwhile, the nodes in each region are divided into three types: data, convergence, and task management nodes. Data collected by adjacent nodes are sent to convergent nodes, and we just need to transmit data of convergent nodes in the network. Task management nodes are used to distribute different sensor nodes to obtain data of different types.

The original data are preprocessed into the standard format needed in the experiments. Using the proposed "site-to-site" graph algorithm and combining historical traffic information and real-time traffic information, the best path search can be conducted timely to ensure an optimal route.

Combined with historical as well as current traffic information, the algorithm can be recommended as the best

path that existing recommendation algorithms cannot reach. Much work remains to be done for further improvement, such as to optimize the algorithm efficiency, to reduce the calculation interval, to experiment with more varieties of data, and to verify the correctness of this algorithm more comprehensively.

References

[1] Z. F. Mao, G. F. Nan, and M. Q. Li, "A dynamic pricing scheme for congestion game in wireless machine-to-machine networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 840391, 9 pages, 2012.

[2] G. F. Nan, G. X. Shi, Z. F. Mao, and M. Q. Li, "CDSWS: coverage-guaranteed distributed sleep/wake scheduling for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, pp. 1–14, 2012.

[3] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 1655–1663, Anchorage, Alaska, USA, April 2001.

[4] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home M2M networks: architectures, standards, and QoS improvement," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 44–52, 2011.

[5] Y. Zhang, R. Yu, M. Nekovee, Y. Liu, S. Xie, and S. Gjessing, "Cognitive machine-to-machine communications: visions and potentials for the smart grid," *IEEE Network Magazine*, vol. 26, no. 3, pp. 6–13, 2012.

[6] C. Tian, Y. Liu, S. Feng, and S. Zhu, "Complete coverage of known space—rectangular decomposition," *Chinese Journal of Mechanical Engineering*, vol. 40, no. 10, pp. 56–61, 2004.

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

[8] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the IEEE Infocom*, pp. 1567–1576, June 2002.

[9] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 88–97, September 2002.

[10] J. Heidemann, F. Silva, and C. Intanagonwiwat, "Building efficient wireless sensor networks with low-level naming," *Operating Systems Review*, vol. 35, no. 5, pp. 146–159, 2001.

[11] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[12] L. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the International Conference on Distributed Computing Systems Workshop (ICDCS '02)*, pp. 575–578, 2002.

[13] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Transactions on Computers*, vol. 35, no. 8, pp. 677–691, 1986.

[14] R. E. Tarjan, "Depth-first search and linear graph algorithms," *Siam Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.

[15] P. Seymour, A. Schrijver, and R. Diestel, "Graph theory," *Oberwolfach Reports*, pp. 135–183, 2005.

[16] A. Broder, R. Kumar, F. Maghoul et al., "Graph structure in the Web," *Computer Networks*, vol. 33, no. 1–6, pp. 309–320, 2000.

[17] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified NP-complete graph problems," *Theoretical Computer Science*, vol. 1, no. 3, pp. 237–267, 1976.

[18] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, "Algorithms for drawing graphs: an annotated bibliography," *Computational Geometry*, vol. 4, no. 5, pp. 235–282, 1994.

[19] J. E. Hopcroft and R. E. Tarjan, "Dividing a graph into triconnected components," *Siam Journal on Computing*, vol. 2, no. 3, pp. 135–158, 1973.

[20] E. Gansner, E. Koutsofios, S. North, and K. P. Vo, "Technique for drawing directed graphs," *IEEE Transactions on Software Engineering*, vol. 19, no. 3, pp. 214–230, 1993.

[21] H. Sun, C. Zhai, and X. Zhan, "Dynamic path planning techniques based on real time traffic information," *Control&Automation*, vol. 23, no. 3–8, pp. 177–178, 2007.

[22] T. Fukuda, K. Takefuji, Y. Ikemoto, and Y. Hasegawa, "Dynamical path-planning for vehicles based on global traffic information and communication," in *Proceedings of the International Conference on Intelligent Transportation (ITSC '02)*, pp. 538–543, 2002.

[23] R. W. Floyd, "Algorithm 97: shortest path," *Communications of the ACM*, vol. 5, no. 6, p. 345, 1962.

[24] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *Journal of the ACM*, vol. 24, no. 1, pp. 1–13, 1977.

[25] A. Schorr, "On shortest paths in polyhedral spaces," in *Proceedings of the ACM Symposium on Theory of Computing (STOC '84)*, pp. 144–153, 1984.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 692146, 10 pages
http://dx.doi.org/10.1155/2013/692146

Research Article

Moving Target Oriented Opportunistic Routing Algorithm in Vehicular Networks

Yu Ding,[1] Wendong Wang,[1] Yong Cui,[2] Xiangyang Gong,[1] and Bai Wang[3]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*
[3] *School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Wendong Wang; wdwang@bupt.edu.cn

Received 11 January 2013; Accepted 12 March 2013

Academic Editor: Jianwei Niu

In vehicular networks, the multihop message delivery from information source to moving vehicles presents a challenging task due to many factors, including high mobility, frequent disconnection, and real-time requirement for applications. In this paper, we propose a moving target oriented opportunistic routing algorithm in vehicular networks for message delivery from information source to a moving target vehicle. In order to adapt the constantly changing topology of networks, the forwarding decisions are made locally by each intermediate vehicle based on the trajectory information of the target vehicle. The simulation and real trace experiment show that our design provides an efficient message delivery with a higher success ratio, shorter success time, and lower transmission overhead compared with other reference approaches.

1. Introduction

Vehicular networks are consisted of moving vehicles which are equipped with dedicated short-range communication (DSRC) and location-aware modules. Usually, vehicular network nodes also have processing and storage capabilities, which have enabled vehicular network to form a new intelligent network. In recent years, vehicular networks have been envisioned to be useful in traffic management, road safety, and other information provision applications. Prime examples of such applications include potential traffic hazards/jams alert, shortest path detection, interests/goals sharing with vehicles, and real-time information obtainment.

Apart from the applications or expected benefits, the challenges and solutions are also the attractions of vehicular networks. A large number of researches have been devoted to this subject [1–3]. Recently, vehicular networks research has involved multihop message delivery from moving vehicle to infrastructure. VADD [4] is a vehicle-assisted message delivery protocol which forwards a message to the best path with the lowest estimation message delivery delay. Geopps [5] presents a forwarding protocol which makes forwarding decisions based on the encounters and geographical information in navigation systems of vehicles.

At the same time, it is also an important research area for returning the message to the moving target vehicle. In this case, the ultimate destination for the message stays in motion, which involves significant challenges. The simplest solutions, such as flooding and epidemic [6] (one delay-tolerant network routing), require high costs to deliver the message successfully, since the message is randomly forwarded in the whole network. Also, there are many MANET protocols [7] designed for moving targets. However, the high speed nodes, dynamic topology, map-based movement, and long distance of moving of vehicle networks make these protocols not so suitable. Trajectory-based statistical forwarding [8] (TSF) is proposed in order to deal with the infrastructure-to-vehicle message delivery performed through the computation of a target point based on the destination vehicle's trajectory. This target point represents an optimal rendezvous point of the message and the destination vehicle. However, the optimal point computation is based on several assumptions, such

as the traffic statistics following the Poisson arrival model (which occurs in optimal situation only). Additionally, the TSF shall apply solely to the situations where stationary nodes are installed at each intersection. A routing protocol is presented in [9] to enable infrastructure-to-vehicle message delivery based on the navigation system. It also requires several APs as infrastructures to provide support, and the mechanism of the protocol is not realized. In spite of that, the above researches are not suitable in actual situation, since infrastructures are not equipped in many cases.

In the majority of cases, a characteristic of the infrastructure-to-vehicle research is the decision of the message transmission path that takes place before the transmission actually starts. However, the traffic condition is also constantly changing; as such, a fixed transmission path may lead to a nonoptimum result. Unlike existing research, the forwarding decisions about the next hop in opportunistic routing (as proposed in this paper) are performed exactly when needed. The transmission path is selected according to the real-time traffic conditions in order to deliver the message with a high success ratio, short success time, and low transmission overhead.

In this paper, we propose a moving target oriented opportunistic routing (MORN) algorithm in vehicular networks, which is designed for delivering messages from an information source to the moving target vehicle. This process will depend on the message's potential being opportunistically carried and forwarded among moving vehicles. In the paper, a problem is formulated for the message delivery, and the subsequent carrier selection is modelled, which is the most important issue in the routing. And then a novel opportunistic routing algorithm employing the unicast strategy is proposed to solve the above problem, in which the next hop is decided according to the trajectory of the subsequent carrier candidate and the target vehicle. In the algorithm, both the success ratio and the success time are taken into account for the subsequent carrier decisions. We evaluate the performance of MORN through the opportunistic network environment (ONE) [10] simulator and real world trace experiment [11–13]. Compared to other reference approaches, MORN has a better performance on message delivery in terms of success ratio, average hops, and overhead.

The rest of this paper is organized as follows. Section 2 gives background of the existing research on multihop infrastructure-to-vehicle message delivery in vehicular networks. The problem formulation of MORN is proposed in Section 3. Section 4 describes how to model the subsequent carrier selection. Section 5 describes the details of MORN. Section 6 provides the evaluation and the result of this research. Finally, Section 7 concludes the paper.

2. Related Work

Multihop message delivery through vehicular networks is complicated by the fact that vehicular networks are highly mobile and have the potential to be frequently disconnected. This issue is complicated further when the destination of a message is also in motion, as the network must locate the position of the moving target vehicle and make sure the message is delivered successfully.

The routing protocol for DTN is one solution to this problem. The epidemic [6] involves message delivery to the target vehicle through the messages exchange throughout the whole vehicular network. While resulting in a maximal spreading speed and maximal delivery success ratio, this approach would also result in a maximal waste of network resources. This process would require the participation of almost all vehicles in the networks, which is impractical. Some kinds of DTN routing [14, 15] deliver the message by estimating the "likelihood" of each vehicle being able to meet the target vehicle, as based on node encounter history. However, the success ratio is not considered satisfactory when the estimation is dependent on few nodes. Conversely, it is also considered a great waste for the computing and storage capability of each vehicle node when the estimation is based on a large number of vehicle nodes.

Among the vehicular networks routing protocols, geographic routing is known to be scalable with respect to the size of network, and as such is a good candidate for intervehicle communications. The geographic routing takes advantage of road map knowledge, calculating the best path between source and destination [16]. GeoDTN + Na [17] is a hybrid Geo-DTN routing solution which incorporates the strength of DTN forwarding in geographic routing to mitigate the impact of intermittent connectivity. Some geographic routing involves a computationally expensive Dijkstra algorithm which leads to each neighbor for each forwarding decision [18]. The best path in geographic routing is the one having the shortest route, the lowest cost, shortest latency, or otherwise optimal attributes of all candidates [19]. A global route is required in the geographic protocols, but it also presents a significant cost with respect to time and network resources. Additionally, there is no guarantee that the computing path is the best one since the traffic conditions are constantly changing. Even more significantly, geographic routings require the coordinates of the target to be known before the message forwarding starts, something which is impossible in vehicular networks when the target vehicle is moving. Some advantaged geographic routing systems assume the coordinates of the target are known at any given time [20, 21]. However, it is still invalid to deliver the message in vehicular networks if the target moves a substantial distance from its known position.

Another solution for the delivery of a message to a moving target involves the calculation of the route before the transmission takes place. This calculation is based on the speed, location, movement, or trajectories of vehicle nodes. A multihop routing approach of a vehicular network in an urban area is presented in [22]. The optimal route from source to destination is selected according to the smallest expected disconnection degree (EDD), which is calculated from the given information on a vehicle's speed, trajectory, and location. The algorithm forwards the message to the vehicle with the smallest EDD value (from route to destination) of all vehicles in the whole vehicular network. The shortest trajectory from source to destination is calculated from the roadway geometry along with the location and movement. There are still two similar algorithms [23, 24] in

Table 1: Notations used in this paper.

v_i	A vehicle node in the network
$p_{v_i}(t)$	The position of v_i at time t
p_{v_i}	The path of vehicle v_i
$x_{v_i}(t), y_{v_i}(t)$	The latitude and longitude of $v_i(t)$
TTL	The validity of a message
t_{now}; time current	The current time of the network
Transmit range	The distance limit of inter vehicle communications
λ	The weight coefficients of success ratio and delay

this process. Although these algorithms result in a relatively good performance, the overhead and massive calculation for the whole network makes the approach unrealistic.

Some advanced research in this area has resulted in improvements to these types of approaches by finding an optimal point on the project path of the target vehicle, which stores the message and waits for the target vehicle to arrive. TSF [8] is designed to find one optimal target point which the vehicle is expected to intersect. This point is also designated as the position where the message is delivered to the destination, and is determined by the distribution of message delay and vehicle delay. When the message is delivered to the target point, a stationary node stores it and waits for the target vehicle. The target point is selected to minimize the delivery delay and satisfy the required message delivery success ratio. Obviously, in this case TSF will work only in the optimal situation that the traffic statistics follow the Poisson arrival models. In [9], a routing protocol which is similar to TSF is presented. Also, there are some data delivery schemes utilizing vehicles' trajectory. STDFS is proposed in [25], which predicts the encounters between vehicles by utilizing shared vehicle trajectories, and an encounter graph is then constructed to aid packet forwarding. STDFS needs access points' help and a precalculation for predicting encounters, which is impractical in vehicular networks.

3. Problem Formulation

In this section, a scenario is described, and then the problem formulation and several related definitions about MORN are provided.

In order to illustrate the problem clearly, let us consider a scenario where a moving vehicle wishes some customized driving information, such as real-time traffic information, as obtained from information source. In this scenario, there are two potential message delivery processes: (1) the vehicle sends the message request to information source, or (2) the message received from information source is forwarded to the moving vehicle reversely. The first process has been studied in the existing research; however, the second process has not been researched in depth as it presents more of a challenge. Here, MORN is proposed for the second process.

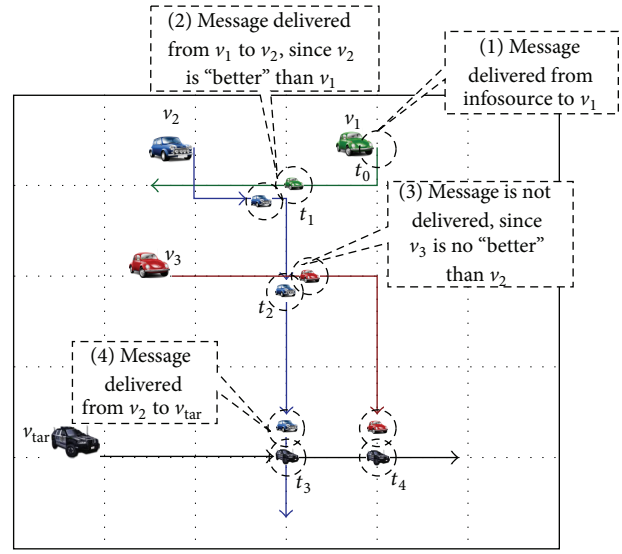


Figure 1: MORN delivering message: $v_1 \rightarrow v_2 \rightarrow v_{\text{tar}}$.

There are two assumptions to be made about the vehicles of the vehicular network: (a) vehicles are equipped with short-range wireless communicate devices (e.g., DSRC device) and GPS navigators with preset destination. With these resources, vehicles are able to disclose their own physical location, destination, and trajectory. Meanwhile, the vehicles can be easily synchronized on the same clock by GPS, and (b) the message needing to be delivered contains the target vehicle's trajectory, the time to live (TTL) of the message, and the message itself. Each vehicle carrying the message can obtain the target vehicle's trajectory and the TTL of the message.

The goal of MORN is to deliver message from information source to a moving target vehicle, as dependent on the ability of the message to be opportunistically carried and forwarded across moving vehicles. The maximum success delivery is the primary consideration, while the minimum success time is also taken into consideration.

Consider a traffic network which is modeled as a directed graph $G(J, E)$. J represents the set of junctions on the traffic map, and E represents the edge set of roads connecting the junctions. On the traffic graph G, V, and P represent two important factors. V is the vehicle set, while $v_i \in V$ represents a vehicle. P is the set of paths representing the tracks of individual vehicles, and $p_{v_i} \in P$ is the path of vehicle v_i. In this paper, we define IS as information source, which is the source of the message delivery, and v_{tar} as the target vehicle (being the destination). The position of vehicle v_i at time t is defined as $p_{v_i}(t)$, which can be represented by geographic information system (GIS) position $(x_{v_i}(t), y_{v_i}(t))$ in the map, where x and y are the latitude and longitude of the position. Table 1 captures the notations used in this paper.

A special case of MORN is shown in Figure 1. In this case, the variables v_1, v_2, and v_3 represent three (nontarget) vehicles, and v_{tar} represents the target vehicle. Each arrow represents the trajectory of a vehicle, while the icons on the trajectory represent the position of the vehicle at a time

point. When the target vehicle wants to receive any kind of message from information source, it issues a request. The request contains the target vehicle's future trajectory as well. The trajectory can be obtained from the navigation system or some routing estimation system. When information source receives the request, the requested message needs to be transferred to the target vehicle as allowed by the opportunistic forwarding capability of the vehicular network. The process by which the message is sent back to the target vehicle is the key point of MORN. The details of the scenario are presented in Figure 1.

First, vehicle v_1 receives the message from information source at time t_0. The trajectory of target vehicle v_{tar} is contained in the message. In the whole process of carrying the message, v_1 periodically broadcasts the trajectory of v_{tar}.

Secondly, vehicle v_1 meets vehicle v_2 at time t_1. After some calculation and comparison, v_1 finds v_2 to be "better" in terms of delivering the message to v_{tar}, since v_2 has the ability to deliver the message in a closer place to target vehicle than v_1. As such, the message is forwarded to v_2, and the message carried by v_1 is discarded.

Next, the same scenario takes place at time t_2, where v_2 meets vehicle v_3. Both v_2 and v_3 have the ability to deliver the message to v_{tar}. As could been seen from Figure 1, however, it is obvious that v_2 has the ability to forward the message to v_{tar} at time t_3, while v_3 can deliver the message at time t_4. Since the message delivery time for v_3 is not "better" in comparison to v_2, the message forward does not take place between v_2 and v_3.

At last, v_2 forwards the message to v_{tar} at time t_3, and the target vehicle has received the message.

4. Subsequent Carrier Selection

Since the MORN is based on the idea of opportunistically unicast routing a message to a moving target vehicle, the most important issue involves the selection of a succession of carriers that are most likely to carry the message to the moving target vehicle. This opportunistic routing will involve exploiting information from the navigation system (NS) in each vehicle. The selection of the subsequent carrier for a vehicle should be considered from 2 items as follows for an efficient message transmission.

(1) The nearness of the candidate and the target vehicle's trajectory.

(2) The possible time of delivery.

More detailed description is as follows.

Definition 1 (trajectory nearest distance $d_{\min}(v_i, v_j)$). The trajectory nearest distance of two vehicles v_i, v_j at any time $0 \le t \le \text{TTL}$ is defined as $d_{\min}(v_i, v_j) = \min_{0\le t\le \text{TTL}} d_{v_i,v_j}(t)$.

Note. The distance of two cars at time t can be calculated as

$$d_{v_i,v_j}(t) = \text{dist}\left(v_i(t), v_j(t)\right) = \sqrt{\left(x_{v_i}(t) - x_{v_j}(t)\right)^2 + \left(y_{v_i}(t) - y_{v_j}(t)\right)^2}. \tag{1}$$

Definition 2 (nearness of trajectory $d^{\star}_{\min}(v_i, v_j)$). The nearness of the two vehicles' trajectory is defined as

$$d^{\star}_{\min}\left(v_i, v_j\right) = \min_{0\le t\le \text{TTL}} \frac{d_{v_i,v_j}(t)}{d_{\text{IS},v_{\text{tar}}}(0)} = \frac{d_{\min}\left(v_i, v_j\right)}{d_{\text{IS},v_{\text{tar}}}(0)} \tag{2}$$

which represents the nearest degree of two vehicles at any available time.

In this way, we can denote the nearness of the candidate and the target vehicle's trajectory as $d^{\star}(v_i)$

$$d^{\star}(v_i) = d^{\star}_{\min}(v_i, v_{\text{tar}}) = \min_{0\le t\le \text{TTL}} \frac{d_{v_i,v_{\text{tar}}}(t)}{d_{\text{IS},v_{\text{tar}}}(0)} = \frac{d_{\min}(v_i, v_{\text{tar}})}{d_{\text{IS},v_{\text{tar}}}(0)}. \tag{3}$$

Definition 3 (minimum time for nearest distance $t_{\min}(v_i, v_j)$). The $t_{\min}(v_i, v_j)$ is defined as the earliest time when two vehicles v_i, v_j intersect at the nearest distance $d_{\min}(v_i, v_j)$.

Definition 4 (possible success time $t^{\star}_{\min}(v_i, v_j)$). The possible success time is defined as $t^{\star}_{\min}(v_i, v_j) = t_{\min}(v_i, v_j)/\text{TTL}$, which represents the earliest degree of two vehicles at the nearest distance.

Hence, the possible success time of the candidate vehicle is

$$t^{\star}(v_i) = t^{\star}_{\min}(v_i, v_{\text{tar}}) = \frac{t_{\min}(v_i, v_{\text{tar}})}{\text{TTL}}. \tag{4}$$

Definition 5 (relative nearness $\sigma(v_i, v_j)$). Consider that

$$\sigma\left(v_i, v_j\right) = \lambda * \left(d^{\star}(v_i) - d^{\star}\left(v_j\right)\right) + (1-\lambda) * \left(t^{\star}(v_i) - t^{\star}\left(v_j\right)\right) \quad \text{for } 0 \le \lambda \le 1 \tag{5}$$

is called relative nearness.

Relative Nearness is a value for comparison of the two vehicles' (v_i and v_j) nearness of trajectory, $d^{\star}(v_i)$ and $d^{\star}(v_j)$. It is also used for comparing the two vehicles' possible success times, $t^{\star}(v_i)$ and $t^{\star}(v_j)$. Different values of λ are used to construct three typical cases: when λ equals 1, and relative nearness considers only the nearness of two vehicles' trajectory transmission, while the success time is only considered when λ equals to 0, and the nearness of trajectory and the success time are both taken into account with λ being other values.

The decision of message validity is also one of the important issues for MORN. If a valid message is misjudged, the message may be not received by target vehicle as expected. On the contrary, if an invalid message is misjudged, a redundant message will be transferred throughout vehicular networks. In MORN, we define the message validity as follows.

(1) The message validity has a close connection with the following track of the vehicle which carries the message. If the following track is far off the trajectory of the target vehicle, the message becomes invalid as its potential possibility to be received by the target vehicle becomes small. The distance validity of the message can be calculated as:

$$M_d(v_i) = \min_{t_{\text{now}} \le t \le \text{TTL}} \frac{d_{v_i, v_{\text{tar}}}(t)}{d_{\text{IS},v_{\text{tar}}}(t_{\text{now}})} = \frac{d_{\min}(v_i, v_{\text{tar}})}{d_{\text{IS},v_{\text{tar}}}(t_{\text{now}})}. \quad (6)$$

(2) The travel time of the message is another item related to message validity. We compare the travel time and the TTL of a message in order to judge if the message is outdated. In order to compute the time validity of the message, we defined the time validity as follows:

$$M_t(v_i) = \begin{cases} \frac{t_{\text{now}}}{\text{TTL}} & \text{if } 0 \le t_{\text{now}} \le \text{TTL}, \\ 1 & \text{if } t_{\text{now}} \ge \text{TTL}. \end{cases} \quad (7)$$

Definition 6 (message validity $M^*(v_i)$). Consider that

$$M^*(v_i) = (1 - M_d(v_i)) * (1 - M_t(v_i)) \quad (8)$$

is defined as the message validity of vehicle v_i at time t_{now}.

5. Moving Target Oriented Opportunistic Routing Design

In this section, we propose the framework of MORN, the opportunistic routing system which delivers message from information source to a moving target vehicle using the trajectory information available in vehicle navigation systems. Following the MORN framework, the key algorithm is explained in detail.

5.1. Framework Design. The main purpose of MORN is to keep looking for vehicles that can potentially deliver the message closer and earlier to the moving target vehicle. It requires the subsequent carrier to be closer to the target vehicle at time t than the present carrier, which means that the trajectory nearest distance of the new candidate is smaller than that of the present carrier. In another instance, the subsequent carrier candidate having the trajectory nearest distance at an earlier time t can also be chosen as the subsequent carrier. The framework can be explained as follows.

(1) A vehicle carrying the message periodically broadcasts the trajectory of the target vehicle.

(2) One-hop neighboring vehicles calculate the nearness of trajectory (3) and possible success time (4) of themselves according to the trajectory of the target vehicle.

(3) The present carrier make decisions, either keeping the message or forwarding it to a one-hop neighbor based on comparing the relative nearness (5) of the neighbor and itself.

(4) The present carrier monitors the message validity (8). If the message is invalid, the present carrier will discard the message.

(5) This process is repeated until the target vehicle receives the message or until the message is invalid.

5.2. Algorithm for Subsequent Carrier Selection. In this subsection, key algorithm of computing the nearness of the v_i and v_{tar}'s trajectory ($d^*(v_i)$) and possible success time of v_i ($t^*(v_i)$) is described. On this basis, the procedure of choosing the subsequent carrier is proposed.

5.2.1. Relative Nearness Computing. To choose the subsequent carrier, we need to calculate the $d^*(v_i)$ and $t^*(v_i)$ of each vehicle v_i, as shown in Algorithm 1. The distance limitation of vehicle-to-vehicle communications is represented as transmit range, and the current time of the network is represented as time current. Initially, the $d_{\min}(v_i, v_{\text{tar}})$ is defined as the current distance of v_i and v_{tar}, while the $t_{\min}(v_i, v_{\text{tar}})$ is the current time (Line 1-2). Secondly, if the current $d_{\min}(v_i, v_{\text{tar}})$ is smaller than transmit range, which means that the v_{tar} can receive the message from v_i directly, and there is no need to find a smaller $d_{\min}(v_i, v_{\text{tar}})$ (Line 3–5). If the current $d_{\min}(v_i, v_{\text{tar}})$ is not small enough, the smallest distance of v_i and v_{tar} will be calculated at each valid time. The process is ended when any distance of time t is smaller than transmit range, or when each distance is computed (Line 6–14). At the end, d^* and t^* are calculated with $d_{\min}(v_i, v_{\text{tar}})$ and $t_{\min}(v_i, v_{\text{tar}})$ (Line 15-16).

5.2.2. Subsequent Carrier Selection. The procedure of choosing the subsequent carrier is shown in Algorithm 2, in which the present carrier is v_i. Initially, v_i calculates the $d^*(v_i)$ (nearness of the v_i and v_{tar}'s trajectory) and $t^*(v_i)$ (possible success time of v_i) (Line 1). Then a decision is made to either chose the neighbor, v_j, as the subsequent carrier, or to keep the message with the current carrier (Line 2–11). This includes the following steps. (i) Make the $d^*(v_j)$ and $t^*(v_j)$ of each vehicle v_j meet during the trip of v_i and then find the optimum v_j^* with the optimum value of $d^*(v_j)$ and $t^*(v_j)$ (Line 3–5). (ii) Compute the $\sigma(v_i, v_j^*)$ according to the definition of relative nearness. If the $\sigma(v_i, v_j^*)$ is larger than a certain threshold, v_j^* is chosen as the subsequent carrier, and the message is transferred from v_i to v_j^*; otherwise, nothing happens (Line 6–10).

Now, we analyze the computation complexity of MORN. In a vehicular network, the number of vehicles met during the whole transmission process is $n = |V_{\text{met}}|$, and the number

Input
Vehicle node v_i;
Output
The nearness of v_i and v_{tar}'s trajectory, $d^*(v_i)$;
The possible success time of v_i, $t^*(v_i)$;
(1) $d_{\min}(v_i, v_{\text{tar}}) = d_{v_i,v_{\text{tar}}}(TimeCurrent)$;
(2) $t_{\min}(v_i, v_{\text{tar}}) = TimeCurrent$;
(3) **if** $d_{\min}(v_i, v_{\text{tar}}) < TransmitRange$ **then**
(4) $t = \text{TTL}$;
(5) end if
(6) **for** each $t \in TimeCurrent \le t < \text{TTL}$ **do**
(7) **if** $d_{v_i,v_{\text{tar}}}(t) < d_{\min}(v_i, v_{\text{tar}})$ **then**
(8) $d_{\min}(v_i, v_{\text{tar}}) = d_{v_i,v_{\text{tar}}}(t)$;
(9) $t_{\min}(v_i, v_{\text{tar}}) = t$;
(10) **if** $d_{\min}(v_i, v_{\text{tar}}) < TransmitRange$ **then**
(11) Break;
(12) **end if**
(13) **end if**
(14) **end for**
(15) $d^*(v_i) = \dfrac{d_{\min}(v_i, v_{\text{tar}})}{d_{\text{IS},v_{\text{tar}}}(0)}$;
(16) $t^*(v_i) = \dfrac{t_{\min}(v_i, v_{\text{tar}})}{\text{TTL}}$;

Algorithm 1: Relative nearness computing.

Input
Current message carrier, vehicle node v_i;
Output
Subsequent message carrier, vehicle node v_j
(1) Read the $d^*(v_i)$ and $t^*(v_i)$;
(2) **while** $\neg((TargetReceiveMsg)$ or $(t \ge \text{TTL}))$ **do**
(3) Get each neighbor v_j of v_i;
(4) Get $d^*(v_j)$ and $t^*(v_j)$ from v_j;
(5) Find the optimum v_j^*;
(6) **if** $\sigma(v_i, v_j^*) > c$ **then**
(7) Transfer Message from v_i to v_j^*;
(8) Delete Message on v_i;
(9) Break;
(10) **end if**
(11) **end while**

Algorithm 2: Subsequent carrier selection for v_i.

of update times is $m = |t|$. For each encounter, a relative nearness is computed having the complexity of $O(m)$ and is required for each of the comparing vehicles. In the whole process of message transmission, an n number of encounters happen. As a result, the overall computation complexity of MORN is $O(mn)$.

6. Performance Evaluation

In this section, we evaluate the performance of MORN through extensive simulations using the ONE [10] simulator. Since DTN routing is one of the possible solutions for message delivery to moving target without the help of stationary nodes, we have compared the performance of MORN with two alternate DTN routing mechanisms—FirstContact [26] and PRoPHET [14]. In the case of the former alternate mechanism, FirstContact, a meeting of two (or more) vehicle nodes, triggers transmission of the message from one node to another as it has time. Then, it removes the message from the first node after it has been transferred. As a result, only one copy of every message is retained in the network—similar to MORN. PRoPHET, the latter alternate mechanism, performs variants of flooding. It estimates the "likelihood" of each vehicle node's ability to deliver a message to the target vehicle based on node encounter history.

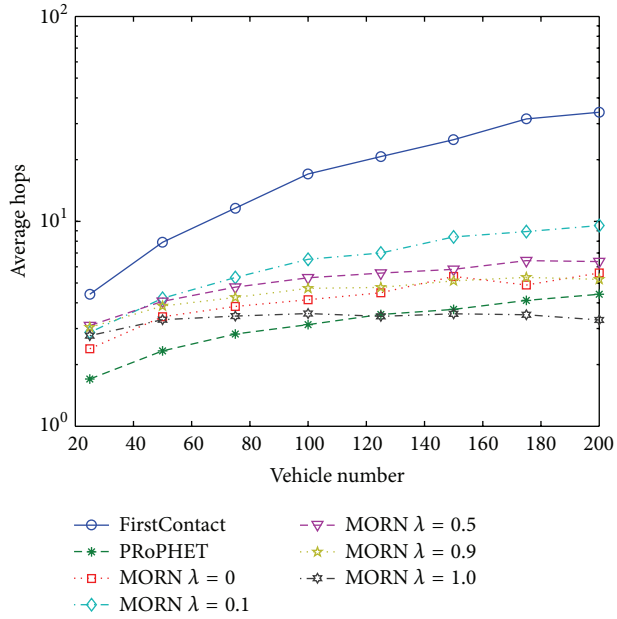


Figure 2: The average hops comparison for different densities.

We conducted 101 rounds of simulations with different random seeds for each vehicle number N. The scenario chosen for simulation was the road map of Helsinki, Finland. Each vehicle's movement pattern is determined by *ShortestPathMapBasedMovement* model. In this model, vehicles take the shortest path on the road of the map exactly. For each simulation, a vehicle node was selected randomly as the target vehicle, and the simulation was repeated 10 times with different target vehicles for each random seed. Only a transmission message was considered in the simulation, and the loss of transmission during the communication procedure was not taken into consideration. The weight coefficients of the success ratio and the success time is represented as λ, and the weight of success ratio and the success time of MORN's consideration varied with the value of λ. Referring to some previous works [4, 27], we set the values for related simulations parameters. The simulations parameters are listed in Table 2.

When $\lambda = 0$, only success time is taken into consideration. This results in a very low success ratio and a short success time. It is a special case with stochastic character for MORN. Therefore, the result of MORN with $\lambda = 0$ is not explicable compared with other cases.

In Figure 2, we have plotted the average hops of algorithms with the vehicles number N increases. For different values of λ of MORN, the average hops decrease as the λ increases. It is obviously that the success ratio has a higher weight in MORN as λ increases. The choice for subsequent carrier shows more characteristics of success ratio. That means that some possible forwardings which may gain shorter success time are canceled. Hence, the forwarding number is dwindling when λ increases. From Figure 2, we can see that FirstContact has a larger number of average hops whenever the density is low or high, and the gap between

Table 2: Simulation parameters.

Parameter	Value
Size of network area	$4500 * 3400\ m^2$
Simulation time	500 s
Transmit range	100 m
Transmit speed	2 Mbps
Vehicle number N	25; 50; 75; 100; 125; 150; 175; 200
Average node speed	15~80 MPH
Message size	7 kB
λ	0; 0.1; 0.5; 0.9; 1

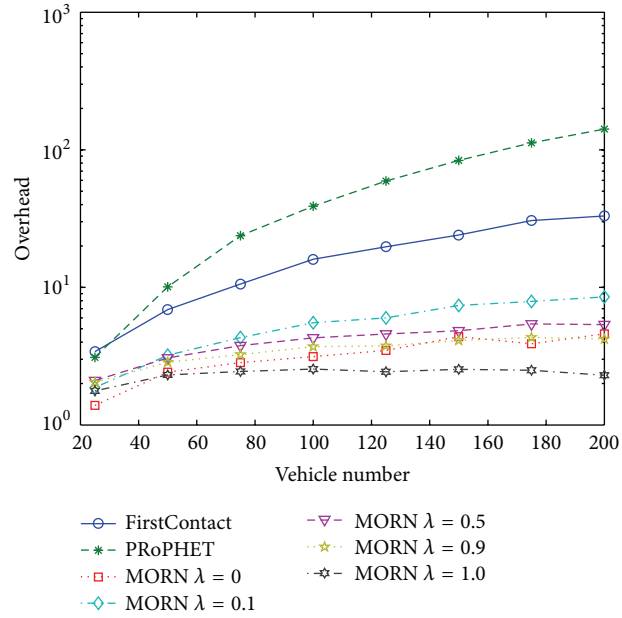


Figure 3: The overhead comparison for different densities.

FirstContact and the other two algorithms is even greater when the density is high. This is primarily caused by the stochastic of the message forwarding in the whole network. At a high vehicle density, the average hops are high as a result of the high vehicle number met during the message trip. The PRoPHET's average hops are similar to the MORN's. That means that MORN and PRoPHET have a similar forwarding number.

Figure 3 demonstrates the transmission overhead (number of message replicas) for different densities. Since both the FirstContact and MORN employ unicast strategy, the overhead of FirstContact and MORN has the same characteristic with the average hops. While the PRoPHET employs multicast strategy, the overhead of PRoPHET is obviously higher than the others. From Figure 3, it is obvious that MORN has the lowest overhead of the three, and the gap between MORN and the others increases as the density increases.

The average success time (ST) comparison is shown in Figure 4. It is obvious that the smaller the λ of MORN, the shorter the average success time. This verifies the effectiveness

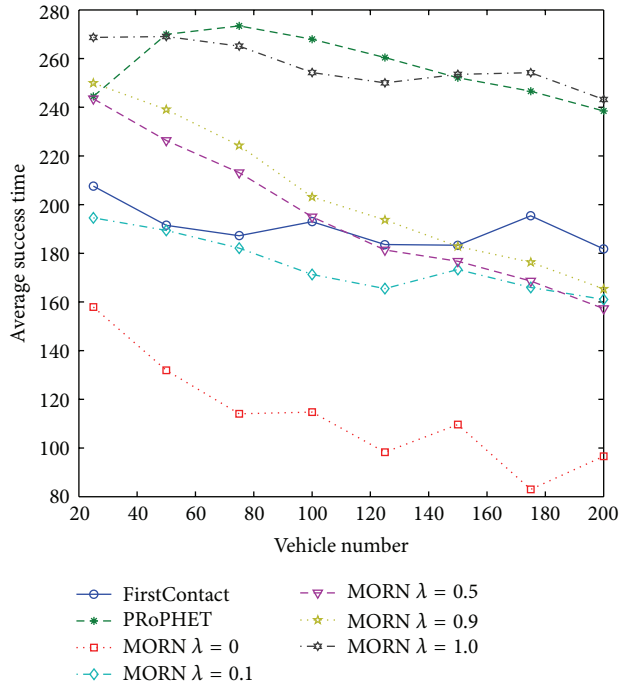


Figure 4: The average success time comparison for different densities.

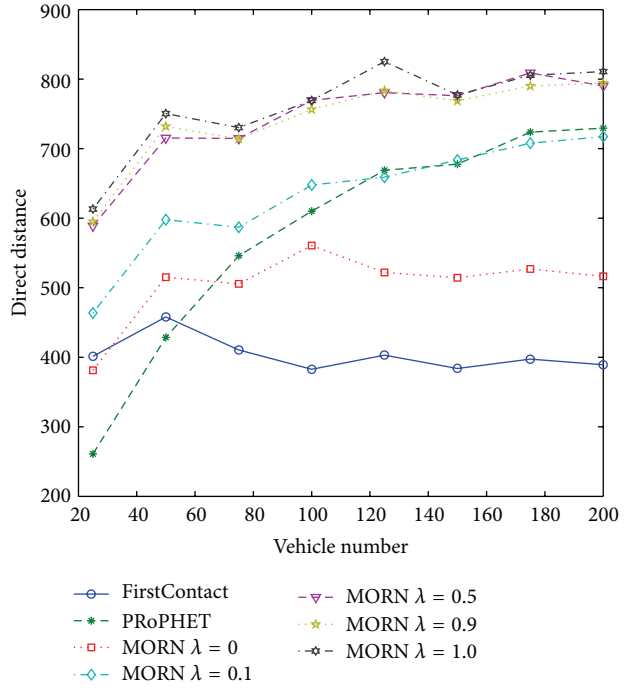


Figure 5: The direct distance comparison for different densities.

of the possible success time ($t^*(v_i)$) in the definition of relative nearness. FirstContact has a relatively lower success time with a lower success ratio (see Figure 7), which means that the FirstContact can only deliver the message successfully when the target vehicle has a relative near distance to the information source in most cases. MORN with $\lambda = 1$ and PRoPHET have the similar average success time. However, MORN has a much lower success time than PRoPHET with other λ values. The result shows that the opportunistic forwarding in MORN has a better performance in success time compared with the forwarding based on node encounter history in PRoPHET.

Figure 5 shows the impact of vehicle number on the direct transmission distance as calculated by FirstContact, PRoPHET, and MORN. From Figure 5, we can see that FirstContact has the lowest direct transmission distance, while MORN has the highest direct distance. The lowest direct transmission distance of FirstContact again demonstrates that FirstContact can only deliver the message successfully when the target vehicle has a relative near distance to the information source in most cases, while the multicast strategy helps PRoPHET to obtain a lower direct distance obviously. For MORN, the direct distance increases as the success ratio has a higher weight due to the λ increases, which means that MORN selects a forwarding strategy which concerns more on success ratio instead of direct transmission distance. Figure 6 shows the vehicle number met during the message trip for different densities. The vehicle number met during the message trip increases as the density of the vehicular network increases.

Figure 7 shows the success ratio (SR) of various algorithms for different densities. The SR of FirstContact is lower

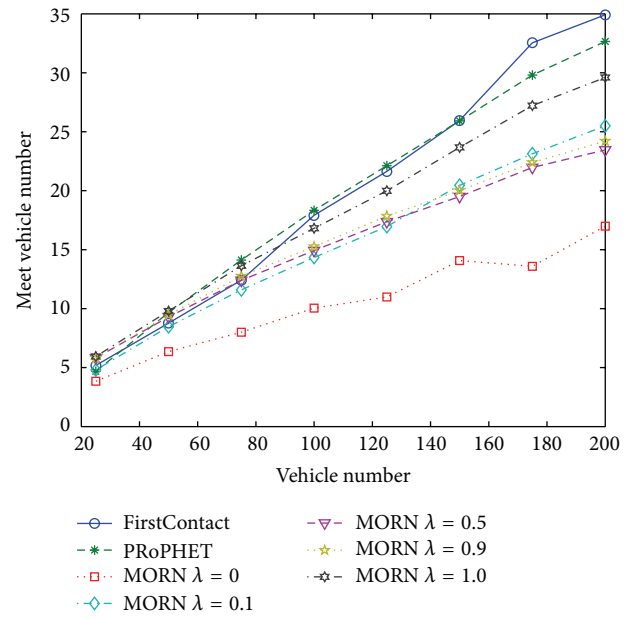


Figure 6: The vehicle number met during the message trip for different densities.

than the other routing algorithms and has no obvious change as determined by the vehicle number. PRoPHET performs better than FirstContact, especially when the vehicle number is larger, meaning that it performs well with greater density of vehicles. For MORN, the SR increases as the λ increases. It is obvious that MORN shows the best performance of all, especially when $\lambda \geq 0.5$.

Table 3: Simulation results.

	Routing type	SR (%)	Hops	Overhead	ST (s)
U_1 with 10 vehicles	FirstContact	0.6	6	7.67	57.8
	PRoPHET	0.8	1	3.75	60.65
	MORN $\lambda = 0.5$	0.8	1	0	62.42
U_1 with 15 vehicles	FirstContact	0.71	5	7.25	31.2
	PRoPHET	0.85	4	10.3	42.6
	MORN $\lambda = 0.5$	0.85	2	1	30.25
U_2 with 20 vehicles	FirstContact	0.5	7	17.4	56.14
	PRoPHET	0.9	3	13	99.36
	MORN $\lambda = 0.5$	1	1	0	70.26
U_2 with 30 vehicles	FirstContact	0.8	5	6.75	28.8
	PRoPHET	1.0	2	6.6	29.08
	MORN $\lambda = 0.5$	1.0	1	0	28.2

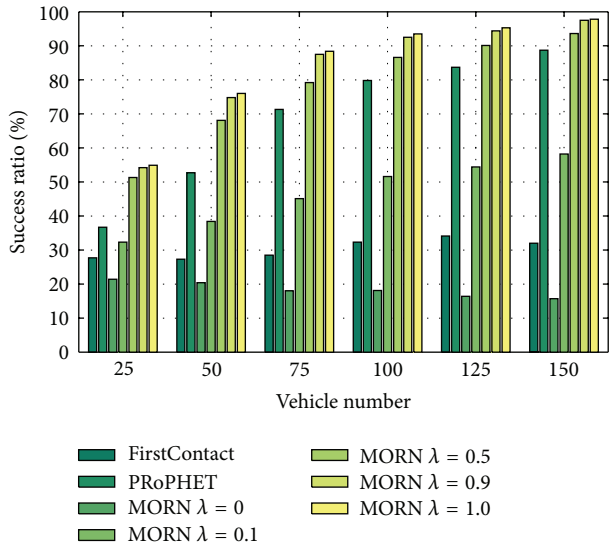


Figure 7: The success ratio comparison for different densities.

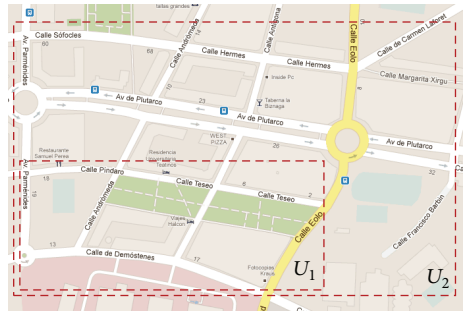


Figure 8: Map of Málaga, Spain.

The performance of the three algorithms is due to the fact that FirstContact delivers the message without optional choice, while both PRoPHET and MORN can deliver the message based on some kind of estimation mechanism. Another reason for the underperformance of FirstContact is that it has only one copy of each message existing in the network, similar to MORN, which results in a "random walk" search for target vehicle. Hence, the SR is not varying too much.

The three algorithms performance evaluations are also carried out by exchanging packets on six different vehicular network instances from real area of Málaga, Spain (Figure 8). Two different sizes of the same metropolitan area and two traffic densities are defined as [11–13]. The road traffic is generated by SUMO [28], in which the vehicles move following the realistic mobility and traffic rules. At the same time, five sources exist for messages generation. The results in Table 3 show that MORN also has a better performance in this situations.

7. Conclusion

In this paper, we present and discuss MORN, an opportunistic routing unicast algorithm in vehicular networks. The main idea of MORN is to identify a "better" message carrier with the potential to deliver a message closer and earlier to the moving target vehicle. And the transmission procedure of MORN is implemented completely through the message carrying and forwarding across vehicles, without any help of infrastructures. In MORN, there is no global route from source to destination that must be created and maintained, and the forwarding decision is made on a per-hop basis. Even the path of the message is not determined before the transmission starts. The evaluation results show that, when compared to the existing algorithms, MORN has a good performance in various vehicle densities in terms of success ratio, average hops, overhead, and success time. Most notably when the vehicle density is high, MORN had an impressive performance.

Acknowledgments

This work was supported in part by the National Basic Research Program of China (973 Program) (Grant no. 2009-CB320504) and the National Natural Science Foundation of China (Grants nos. 61271041 and 61202436).

References

[1] R. Verdone, "Multihop R-ALOHA for intervehicle communications at millimeter waves," *IEEE Transactions on Vehicular Technology*, vol. 46, no. 4, pp. 992–1005, 1997.

[2] J. LeBrun, C. N. Chuah, D. Ghosal, and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *Proceedings of the 61st Vehicular Technology Conference (VTC '05)*, pp. 2289–2293, Dallas, Tex, USA, June 2005.

[3] J. Zhao, T. Arnold, Y. Zhang, and G. Cao, "Extending drive-thru data access by vehicle-to-vehicle relay," in *Proceedings of the 5th ACM International Workshop on VehiculAr Inter-NETworking (VANET '08)*, pp. 66–75, September 2008.

[4] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.

[5] I. Leontiadis and C. Mascolo, "GeOpps: geographical opportunistic routing for vehicular networks," in *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM '07)*, pp. 1–6, June 2007.

[6] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Tech. Rep., 2000, http://www.cs.duke.edu/~vahdat/ps/epidemic.pdf.

[7] K. Weniger and M. Zitterbart, "Address autoconfiguration in mobile ad hoc networks: current approaches and future directions," *IEEE Network*, vol. 18, no. 4, pp. 6–11, 2004.

[8] J. Jeong, S. Guo, Y. Gu, T. He, and D. H. C. Du, "TSF: trajectory-based statistical forwarding for infrastructure-to-vehicle data delivery in vehicular networks," in *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS '10)*, pp. 557–566, June 2010.

[9] I. Leontiadis, P. Costa, and C. Mascolo, "Extending access point connectivity through opportunistic routing in vehicular networks," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM '10)*, pp. 486–490, San Diego, Calif, USA, March 2010.

[10] A. Keranen, J. Ott, and T. Karkkainen, "The ONE simulator for DTN protocol E-valuation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools '09)*, Rome, Italy, March 2009.

[11] J. Toutouh, J. Garcła-Nieto, and E. Alba, "Intelligent OLSR routing protocol optimization for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1884–1894, 2012.

[12] J. Toutouh and E. Alba, "Optimizing OLSR in VANETS with differential evolution: a comprehensive study," in *Proceedings of the 1st ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet '11)*, pp. 1–8, Miami, Fla, USA, November, 2011.

[13] J. Toutouh and E. Alba, "An efficient routing protocol for green communications in vehicular ad-hoc networks," in *Proceedings of the 13th Annual Conference Companion on Genetic and Evolutionary Computation (GECCO '11)*, N. Krasnogor, Ed., pp. 719–726, ACM, New York, NY, USA.

[14] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *Proceedings of the 1st International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR '04)*, vol. 3126 of *Lecture Notes in Computer Science*, pp. 239–254, 2004.

[15] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: routing for vehicle-based disruption-tolerant networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–11, April 2006.

[16] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 156–161, June 2003.

[17] P. C. Cheng, K. C. Lee, M. Gerla, and J. Härri, "GeoDTN+Nav: geographic DTN routing with navigator prediction for urban vehicular environments," *Mobile Networks and Applications*, vol. 15, no. 1, pp. 61–82, 2010.

[18] J. Tian, L. Han, K. Rothermel, and C. Cseh, "Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks," in *Proceedings of IEEE Intelligent Transportation Systems (ITS '03)*, pp. 1546–1551, Shanghai, China, October 2003.

[19] B. Seet, G. Liu, B. Lee, C. Foh, K. Wong, and K. Lee, "A-STAR: a mobile ad hoc routing strategy for metropolis vehicular communications," in *Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, vol. 3042 of *Lecture Notes in Computer Science*, pp. 989–999, Springer, Berlin, Germany, 2004.

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

[21] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, 2003.

[22] Z. Mo, A. Zhu, K. Makki, and N. Pissinou, "MURU: a multihop routing protocol for urban vehicular ad hoc networks," in *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous '06)*, pp. 1–8, July 2006.

[23] A. Festag, H. F. Fußler, H. Hartenstein, A. Sarma, and R. Schmitz, "FLEETNET: bringing car-to-car communication into the real world," in *Proceedings of 11th World Congress on Intelligent Transportation Systems (ITS '04)*, pp. 1–8, Nagoya, Japan, October 2004.

[24] D. Yu and Y. B. Ko, "FFRDV: fastest-ferry routing in DTN-enabled vehicular ad hoc networks," in *Proceedings of the 11th International Conference on Advanced Communication Technology (ICACT '09)*, pp. 1410–1414, February 2009.

[25] F. Xu, S. Guo, J. Jeong et al., "Utilizing shared vehicle trajectories for data forwarding in vehicular networks," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 441–445, April 2011.

[26] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 145–157, ACM, New York, NY, USA, September 2004.

[27] B. Xu, A. Ouksel, and O. Wolfson, "Opportunistic resource exchange in inter-vehicle ad-hoc networks," in *Proceedings of IEEE International Conference on Mobile Data Management (MDM '04)*, pp. 4–12, usa, January 2004.

[28] D. Krajzewicz, M. Bonert, and P. Wagner, *The Opensource Traffic Simulation Package SUMO*, InRoboCup06, Bremen, Germany, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 813182, 12 pages
http://dx.doi.org/10.1155/2013/813182

Research Article

An Energy-Efficient Motion Strategy for Mobile Sensors in Mixed Wireless Sensor Networks

Zhen-Jiang Zhang,[1] Jun-Song Fu,[1] and Han-Chieh Chao[2]

[1] *Department of Electronic and Information Engineering, Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China*
[2] *Department of Electrical Engineering, National Dong Hwa University, Hualien 26249, Taiwan*

Correspondence should be addressed to Zhen-Jiang Zhang; zhjzhang1@bjtu.edu.cn

Received 7 January 2013; Accepted 7 March 2013

Academic Editor: Jianwei Niu

The mixed wireless sensor networks that are composed of a mixture of mobile and static sensors are the tradeoff between cost and coverage. To provide the required high coverage, the mobile sensors have to move from dense areas to sparse areas. However, where to move and how to move are important issues for mobile sensors. This paper presents a centralized algorithm to assist the movement of mobile sensors. In this algorithm, the management node of the WSN collected the geographical information of all of the static and mobile sensors. Then, the management node executed the algorithm to get the best matches between mobile sensors and coverage holes. Simulation results show the effectiveness of our algorithm, in terms of saving energy and the load balance.

1. Introduction

Wireless sensor networks (WSNs) have been used extensively due to their excellent capability of monitoring real physical environments and collecting data. However, in order to conduct their tasks successfully, it is very important that they be deployed properly, taking into account the limitations of energy support and transmission power.

WSNs cannot be deployed manually in many working environments, such as remote mountainous regions, battlefields, and regions polluted by poisonous gases. An alternative method is scattering the sensors randomly, but this is affected by many uncontrollable factors, and it is difficult to achieve the desired deployment. In early studies, most of the networks consisted of a large number of static nodes, and there were many redundant nodes. Importantly, this approach often led to high cost and uncertainties concerning the coverage.

In the last decade, researchers have focused on mixed networks, which are composed of both static nodes and mobile nodes. Such networks have the advantage of mobility, so they can be moved to appropriate positions to enhance the extent of coverage and reduce the number of nodes. However, there are few algorithms that have been proposed to guide the mobile sensors from their original positions to the desired positions in mixed wireless sensor networks except two classic algorithms which will be introduced in Section 2.

Our problem statement is "given the target area, static sensors that are immobile, and sensors with flexible mobility, design and implement a plan that maximizes the coverage of the sensors, minimizes energy consumption, and results in a balanced energy load." In this paper, we report the results of our design of a new, centralized algorithm for the placement of mobile sensors in a mixed network to achieve the goals mentioned in the problem statement. The algorithm assists in decision making concerning moving the mobile sensors to fill the holes, where coverage was not provided by any sensor. In order to save energy, we attempted to shorten the total distance that all of the mobile sensors moved. In addition, we made sure that no sensors moved an extremely long distance in order to balance the energy load. The implementation of the algorithm was divided into four stages, that is, (1) the management node of the WSN collected the geographical information of all of the static and mobile sensors; (2) the management node executed the algorithm to determine

the best positions to which all of the mobile sensors should move to; (3) the management node issued the resulting locations to the mobile sensors through the network management system (NMS); and (4) the mobile sensors moved to the positions specified by the NMS.

The rest of the paper is organized as follows. Section 2 introduces basic information concerning Delaunay triangulation. The centralized algorithm is presented in Section 3, and its performance is evaluated in Section 4. A summary of the paper is presented in Section 5.

2. Related Work

Many significant achievements have been made in the field of energy-efficient coverage. Due to the characteristics of the nodes that comprise WSNs, there are three types of such networks, that is, (1) static WSNs, in which all the nodes are static; (2) mobile WSNs, in which all the sensors are mobile; and (3) mixed WSNs, in which some of the nodes are static and some are mobile.

The greatest weakness of a static WSN is that there must be significant redundancy among the nodes in order to achieve good coverage. As a result, one of the most important issues to consider in the implementation of a static WSN is energy-efficient coverage (EEC). Many algorithms have been proposed to conserve energy and prolong the network's lifetime [1–5]. Among these algorithms, scheduling methods have been shown to be effective in reducing energy consumption by planning the activities of the devices [6–9].

In mobile WSNs, a fundamental issue is the coverage problem. Many techniques have been developed to deal with this issue, such as coverage pattern-based movement [10–13], virtual force-based movement [14, 15], and grid quorum-based movement [16–19]. The greatest weakness of a mobile WSN is its price, which is significantly greater than the price of a static WSN, because the price of mobile sensors is much greater than the price of static sensors.

A mixed WSN is a tradeoff between a static WSN and a mobile WSN. In most cases, a mixed WSN is the best choice. In a previous work, Voronoi diagrams were used to study methods for estimating the coverage holes in mixed sensor networks [20]. The author proposed a collaborative algorithm (Coven) to determine the location of the mobile sensors for enhancing coverage by estimating the relationship between the area of the coverage hole and the coverage radius of the sensors. However, it is not feasible to apply Voronoi diagrams in WSNs due to their excessive complexity.

Concerning the strategy for moving the sensors, one study [21] proposed a bidding protocol in mixed sensor networks that treated static nodes as bidders or consumers and mobile nodes as service providers. An optimal relay placement for indoor sensor networks was proposed. Mobile nodes have a base price which is the size of coverage holes produced by mobile nodes leaving their positions to cover other holes. Static nodes bid for mobile nodes, and the bid price is the size of the coverage holes detected by static nodes. When the bid price for the mobile nodes is less than the price of the static nodes, the mobile nodes will provide service for the static nodes to cover the holes that have been detected by the static nodes. However, the authors provided no discussion of the energy balances of all of the sensors, so the lifetime of the WSN was short. Additionally, applying the bidding protocols in [21], several sensors may have to move a long distance, which would take more time to complete the construction of the sensor network.

In order to improve the effectiveness of the algorithm in paper [21], paper [22] introduced a proxy-based bidding protocol, which was an improvement over the basic bidding protocol. To reduce distances that the mobile sensors had to move, the proxy-based bidding protocol proposes that mobile sensors perform virtual movements from small holes to large holes and that they only perform physical movements after the final destinations have been identified.

To our knowledge, the paper [21] is the first paper on deploying a mixture of static and mobile sensors to meet the coverage requirement, and [22] is an improvement of [21]. There are few other papers that focus on this problem, and we will compare the algorithm in this paper with the two algorithms.

3. Preliminaries

3.1. Assumptions. In our research, we used the classical Boolean sensing model, in which the sensing area is represented by a circle with a radius. The Boolean sensing model facilitated the analysis and helped us understand the problem. Obviously, we must know the locations of all the sensors, and it had been proved that each node in the wireless sensor network had the ability to establish its location through GPS system or some other form of localization technique [23–27]. In our research, we assumed that at least one of the techniques was available in our WSN.

After executing our algorithm, the management node identified the best positions to which all the mobile nodes should move. But we were unable to plan the path from the current position to the desired destination for the mobile nodes. However, there are also several existing techniques that can be used to plan the path, and we assumed that at least one of them was available.

3.2. Preliminary Technique: Delaunay Triangulation. A Voronoi diagram represents the proximity information about a set of nodes. It provides a geometric construction that detects the coverage holes very effectively, especially in distributed algorithms in which position information for all of the sensors is unknown. However, it is not the best choice for centralized algorithms, so we used Delaunay triangulation instead of a Voronoi diagram.

Delaunay triangulation and the Voronoi diagram are dual constructions. Given a set of nodes, we used a randomized incremental algorithm to triangulate them into many triangles, and the circumcenters of the triangles were the key areas that we had to detect carefully. First, we triangulated all of the static nodes and determined the best position for the mobile sensor to provide the desired coverage. Then, the mobile sensor was not moved to any other place, so,

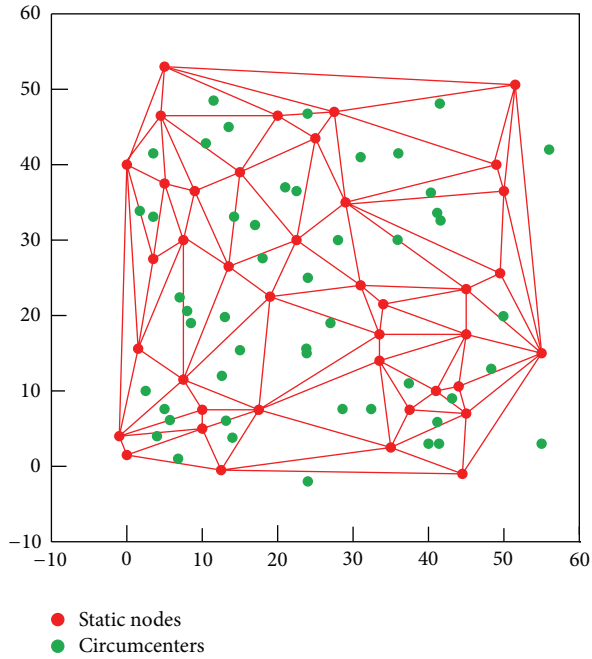


Figure 1: Delaunay triangulation.

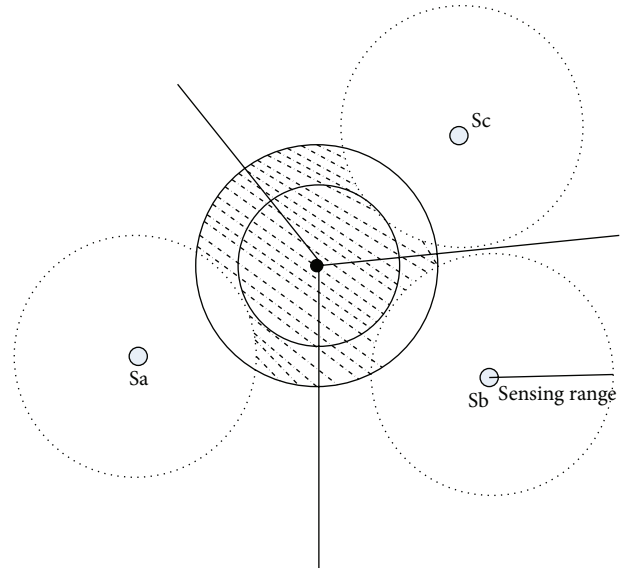


Figure 2: Size of a coverage hole.

in effect, it became a static sensor, and we added it to the set of static nodes. We iterated the process until all of the mobile nodes became static nodes. Then, we determined all of the desired positions for the mobile sensors. The features of Delaunay triangulation guarantee that the distance between the circumcenter of a triangle and its vertices is shorter than the distance between the circumcenter and any other nodes. So, if a the circumcenter of a triangle is not covered by its vertexes, there must be a coverage hole that a mobile sensor should be moved to cover. We will give a detailed introduction in Section 4.

4. Centralized Algorithm

In this section, we present the centralized algorithm, which is divided into four stages. The second stage is the core of the algorithm, and we focused mainly on it. The discussion of the algorithm is presented in four steps, that is, (1) detect the coverage holes, (2) choose the desired positions, (3) use the greedy algorithm to obtain an initial result, and (4) optimize the initial result using the 2-exchange optimization algorithm.

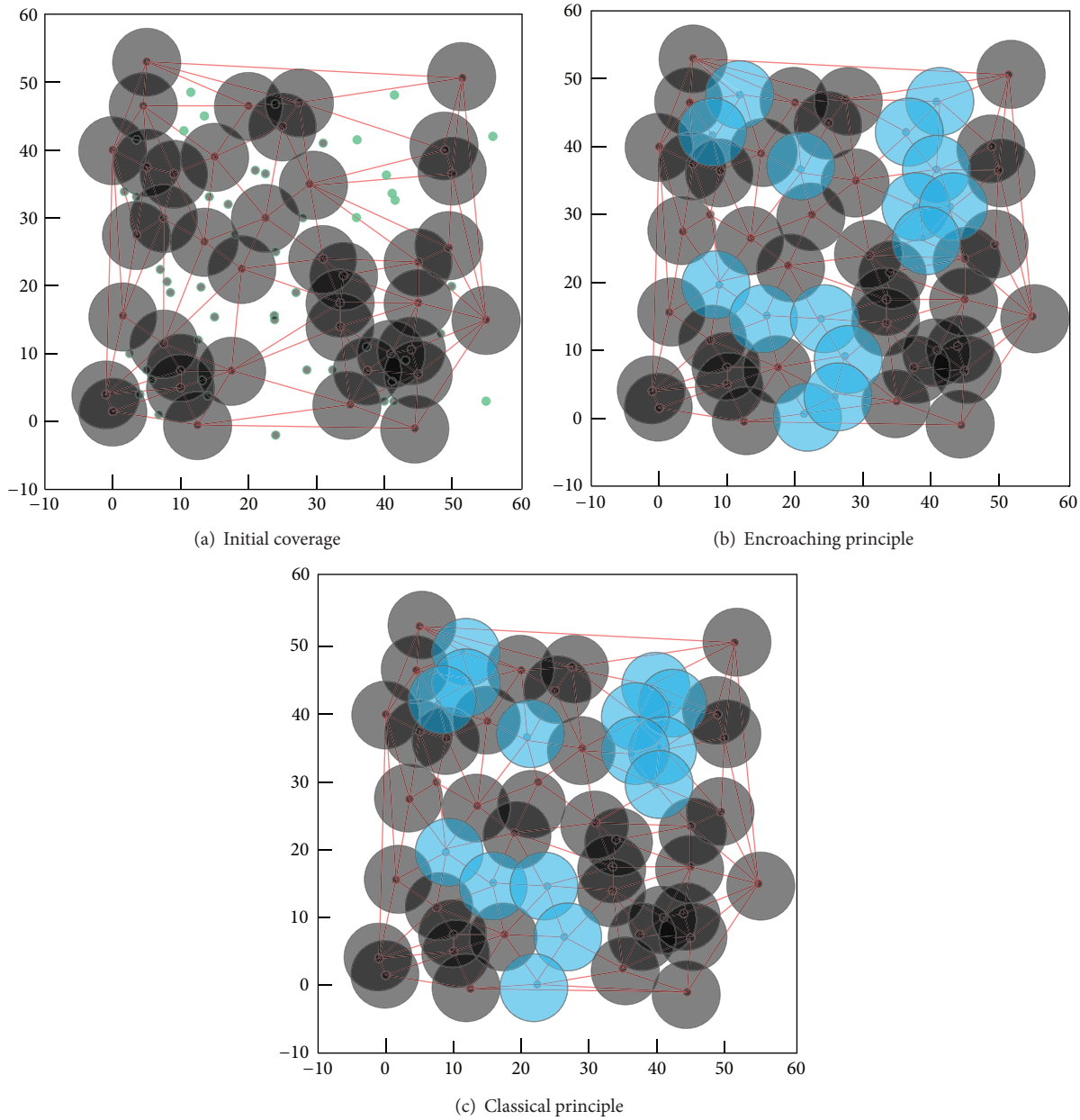
4.1. Detect the Coverage Holes. We used Delaunay triangulation to detect the coverage holes. First, we triangulated the set of static nodes into many triangles and calculated the circumcenter of every triangle, as shown in Figure 1. Second, we detected whether the circumcenter was covered by the three static sensors that formed the triangle. If the circumcenter was covered, it is not a coverage hole. The size of the hole was calculated as

$$s = \pi * (d - R_s)^2, \quad d \geq R_s, \tag{1}$$

where d is the distance between the circumcenter and the vertexes of the triangle and R_s is the sensing range of the sensor, as shown in Figure 2. Because π and R_s are constants for a WSN, we can use d to represent the size of the coverage hole. If $d \leq R_s$, there is no coverage hole; if $d \geq R_s$ and $d \leq 2R_s$, there is a hole, most of which can be covered by one mobile sensor; if $d \geq 2R_s$, there is a big hole that may require more than one mobile node to cover it completely.

4.2. Choose the Desired Positions. Having detected the holes, we must decide which hole is to be covered by a mobile sensor first. To do this, we used the encroaching principle, that is, that the holes adjoining the covered area should be covered preferentially, rather than the bigger holes as has been classically done. The rationale was that if we covered the bigger holes first, there will be locations between the hole and the covered area that will be covered twice. Simulation showed that our approach was better than following the classic approach, as indicated by the improved coverage percentages shown in Figures 3(b) and 3(c), respectively. We calculated the coverage percentage by image processing and determined that the initial coverage percentage was 68.3%, while the coverage percentage of the classic approach was 93.2%, and the coverage percent of the encroaching principle that we used was 95.1%.

Our discussion of the randomized incremental algorithm follows. After we had triangulated the set of static nodes, we were able to detect the coverage holes and identify the place the mobile sensors should cover first. Then, we moved one mobile sensor to that place, and the sensor remained stationary after that. Thus, the mobile sensor became a static sensor, resulting in a change in the triangulation of the static sensors. Before we chose the next hole to be covered, we had to obtain the new triangulation and then decide the hole using the encroaching principle. We iterated the process until the coverage percentage reached a constant value, for example, 95%.



(a) Initial coverage

(b) Encroaching principle

(c) Classical principle

Figure 3: The coverage results by different principles.

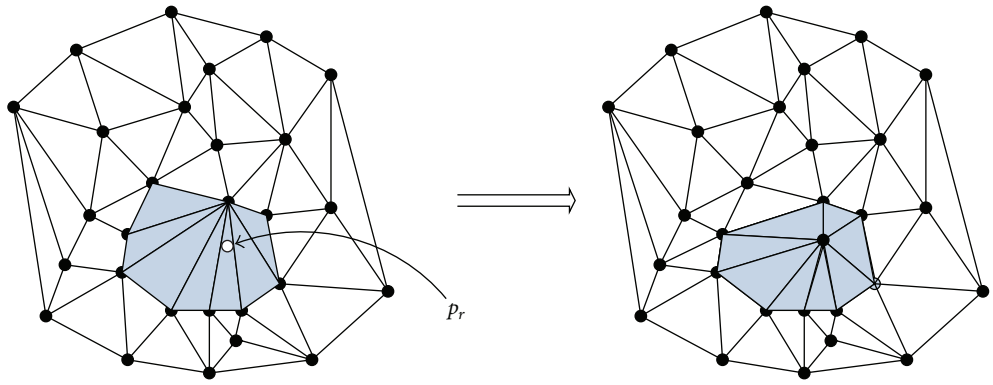
But we did not have to calculate the new triangulation repeatedly. We can calculate the new triangulation based on the previous result, and this is the core idea of the randomized incremental algorithm.

A triangulation is shown in Figure 4. We added a new point, p_r, to the set of nodes and calculated the new triangulation. As shown in Figure 4(b), the new triangulation is only different from the previous triangulation in part. When the number of nodes is large, the difference between the two triangulations is very small, so we just have to calculate a part of the new triangulation. Before we introduce the algorithm, the illegal edge is defined as follows.

In Figure 5(a), there are six angles, that is, $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\}$, and there are six angles, that is, $\{\alpha_1', \alpha_2', \alpha_3', \alpha_4', \alpha_5', \alpha_6'\}$, in Figure 5(b). If $\min(\alpha_i) < \min(\alpha_i')$, $1 \le i \le 6$, then $p_i p_j$ is an illegal edge.

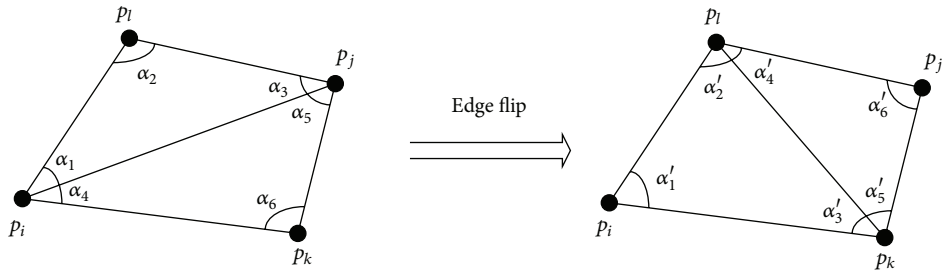
We assumed that T is the triangulation, and the randomized incremental algorithm is described in Algorithm 1.

The function of LegalizeEdge (p_r, $p_i p_j$, T) is describe in Algorithm 2. Now, we have detected the coverage holes and chosen the desired position successfully, which provides the basis for the next step. The numbers of mobile nodes and desired positions are fixed. Once the mobile nodes have moved to the desired positions, the coverage percentage



(a) Previous Result before p_r (b) Result after have added p_r

Figure 4: A triangulation in randomized incremental algorithm.



(a) $p_i p_j$ is illegal edge (b) No illegal edge

Figure 5: Edge flip.

```
(1) do (add p_r to T)
(2)    find the triangle p_i p_j p_k ∈ T that p_r in it
(3)    if (p_r is located in p_i p_j p_k)
(4)       then line p_r and the three vertexes of p_i p_j p_k as shown in Figure 6
(5)       LegalizeEdge (p_r, p_i p_j, T);
(6)       LegalizeEdge (p_r, p_j p_k, T);
(7)       LegalizeEdge (p_r, p_k p_i, T);
(8)    else (* p_r is located on the edge, we assume that the edge is p_i p_j)
(9)       line p_r and p_k, p_l as shown in Figure 7
(10)      LegalizeEdge (p_r, p_i p_j, T);
(11)      LegalizeEdge (p_r, p_j p_k, T);
(12)      LegalizeEdge (p_r, p_k p_j, T);
(13)      LegalizeEdge (p_r, p_j p_i, T);
(14)   return (T);
```

Algorithm 1

```
(1) If (p_i p_j is illegal)
(2)    then assume p_i p_j p_k is the triangle that adjacent to p_i p_j p_r
(3)       replace the edge p_i p_j by p_r p_k
(4)       LegalizeEdge (p_r, p_i p_k, T)
(5)       LegalizeEdge (p_r, p_j p_k, T)
(6) end
```

Algorithm 2

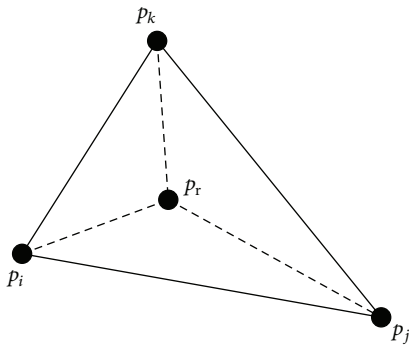


Figure 6: p_r is located in $p_i p_j p_k$.

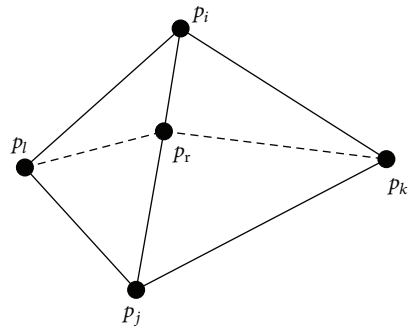


Figure 7: p_r is located on the edge.

obviously will be improved. In order to save energy and balance the energy load, we designed a detailed scheme for determining the positions to which the mobile sensors should move. We can describe the problem in mathematical language as follows.

Given two sets of X and Y:

$$X = \{x_1, x_2, \ldots, x_m\},$$
$$Y = \{y_1, y_2, \ldots, y_m\}, \tag{2}$$

where m is a constant number. There are no common elements in the two sets, that is,

$$X \cap Y = \emptyset. \tag{3}$$

We must match the elements in X and the elements in Y one to one. For example,

$$\{\{x_1, y_1\}, \{x_2, y_2\}, \ldots, \{x_m, y_m\}\} \tag{4}$$

is a one-to-one match. Also, there is a weight for every match $\{x_i, y_j\}$, and we must find a match with the low total weight and without extremely large weight. A heuristic algorithm is a good solution, and we used the greedy algorithm and the 2-exchange optimal algorithm to solve this problem.

In our WSN, the weight of a match is the Euclidean distance between mobiles sensors and the holes that should be covered by the mobile sensors. In order to save energy, mobile nodes should move the shortest distance possible, and, in order to balance the load, the distances should be

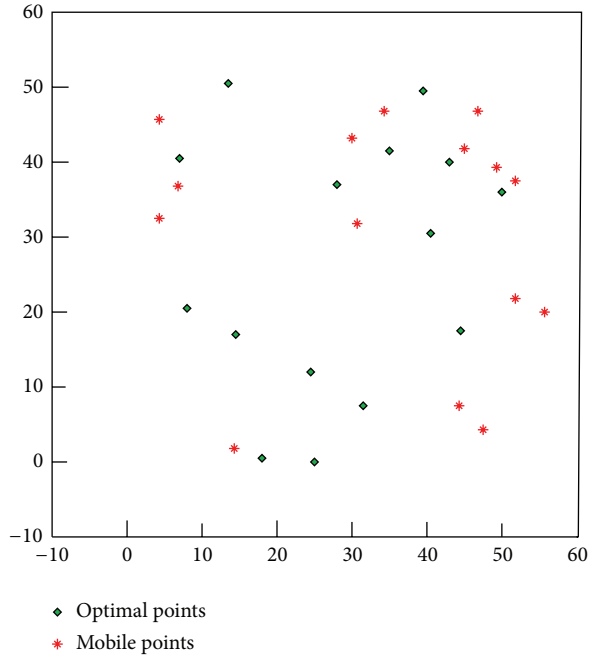


Figure 8: The initial situation.

similar. We assumed two measures for our algorithm, that is, the average distance that the mobile sensors moved and the variance between the distances.

4.3. Obtain an Initial Result by Greedy Algorithm. Now, we have obtained the current positions of the mobile sensors and the positions to which they should move, as shown in Figure 8. For example, the red stars are mobile sensors, and the green points are the coverage holes. We determined the acceptable match in the following two steps, that is, (1) obtain an initial result by greedy algorithm and (2) optimize the result using the 2-exchange optimization algorithm.

Now, we introduce the greedy algorithm in five steps.

Algorithm 1. Greedy algorithm for match

Input. $2n$ points, n mobile sensors, and n coverage holes.

Output. one-to-one match for mobile sensors and holes.

Step 1. We order the holes from 1 to n as shown in Figure 9, where n is the number of the holes.

Step 2. Push the geographical information of the holes to a stack in reverse order and push the information of mobile nodes in another stack randomly, as shown in Figure 10.

Step 3. Choose the mobile node in stack M that is the closest to the hole in the top of stack H and record the match.

Step 4. Pop stack H and delete the mobile node in stack M.

Step 5. If there are no holes in stack H, exit the algorithm, otherwise, return to Step 3.

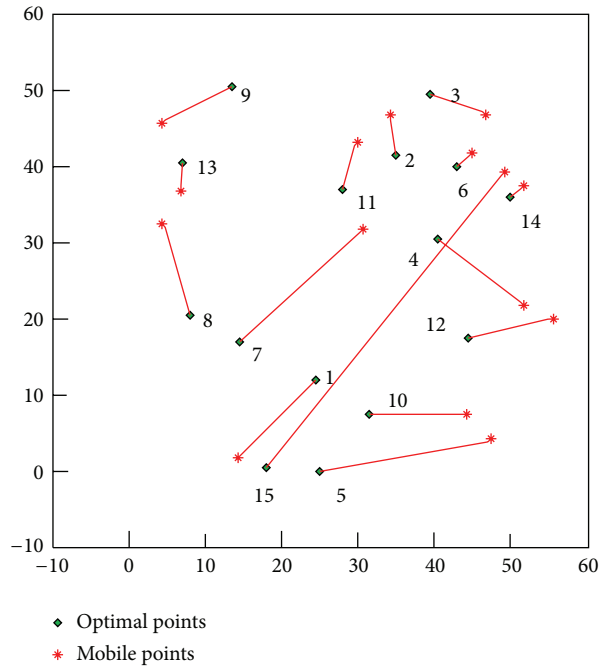


Figure 9: The match result by greedy algorithm.

The results of the algorithm are shown in Figure 9. Every line segment with an arrow indicates that a mobile sensor should move to a coverage hole. It cannot meet our requirements for both average distance and the variance of the distance, and we had to improve the results by using the 2-exchange optimal algorithm.

As shown in Figure 9, when the serial number of the hole is smaller, it is more likely that the length of the line segment with an arrow that points to the hole from the corresponding mobile sensor is shorter. It is an inherent flaw of the greedy algorithm that the results are optimal in local but are not as good in global.

4.4. Optimize the Initial Result by Using the 2-Exchange Optimization Algorithm. The results shown in Figure 9 are not good enough, and we optimized the results by using the 2-exchange optimization algorithm. There is no common point for any two lines because the mobile nodes and coverage holes are a one-toone match. The core idea of the algorithm is to exchange the destination of any two line segments to see whether their total length decreased. If so, we exchanged the destination of the line segment and recorded the result; otherwise, we maintained the present status. The 2-exchange optimization algorithm is a special case of the N-exchange optimal algorithm, and, if N is equal to the number of mobile sensors, the result is the best result.

One problem is the organization of the exchange order to guarantee that no line is ignored or repeated. Because the holes match the lines one to one, the number of a hole can represent the corresponding line, that is to say, the lines are ordered from 1 to n. The optimization algorithm is described in Algorithm 3.

The main part of the algorithm is a loop, and, from the results of the simulation, we found that the margin of improvement in the first loop was the most obvious; the more times we executed the loop, the less improvement we obtained from every execution. Though it is a heuristic algorithm, its complexity cannot be ignored. In fact, if we insist on finding the 2 best results, that is to say, we execute the loop until $L' - L'' = 0$, it is likely that the time required would be unacceptable. In addition, in a WSN, an acceptable result that can be obtained in a simple way is the best result. In fact, the result we obtained was excellent, just 1.04–1.06 times the best match obtained mathematically, and the complexity was reduced significantly.

Figures 10(a) and 10(c) are the initial trace, and Figures 10(b) and 10(d) are the trace after the optimization.

When there were 15 mobile sensors and $\varepsilon = 5$ meters, the loop was executed 2-3 times, and the convergence property was excellent, as shown in Figure 10(c). In order to test the pressure resistance of the algorithm, we assumed that there were 100 mobile sensors and that the loop was executed four times, as shown in Figure 10(d). It is apparent that the complexity increased slowly as the number of mobile sensors increased.

5. Performance Evaluations

5.1. Tradeoff between Cost and Coverage. Before we tested the effectiveness of our algorithm, first, we proved that the mixed WSN was meaningful in real applications. The cost of a WSN is based on the cost of the sensors. We ran simulations for three different compositions of sensors in a WSN and set the coverage to be 90%, 95%, and 98%, respectively. We assumed that the target area was a 50-m × 50-m, square, flat, field and randomly scattered 60 sensors, including static sensors and mobile sensors, in the field. The percentage of mobile sensors varied from 0% to 100%, in 10% increments. The transmission range was set as 20 m, and the sensing range was set as 5 m. We ran 100 simulations for every result and calculated the average result.

There were three cases of network compositions, that is, (1) all sensors were static and random deployment was used; (2) all sensors were mobile, and the VOR protocol was used to deploy the sensors; and (3) for sensor deployment, a percentage of the sensors were mobile, and we used our centralized algorithm. The results are shown in Figure 11.

Because of the randomness of the static sensors, sometimes we could not calculate the accurate minimum number of the sensors to reach a certain coverage. For example, if all the sensors were static and, unfortunately, we placed them all at the same position, we could never cover the target field. In this paper, we scattered the same number sensors 100 times and obtained 100% coverage in each case. For a number of sensors, N, if the average coverage just reached a certain coverage, such as 95% and for $N - 1$, the average coverage cannot reach 95%, then we defined the number N as the minimum number of sensors required to reach a coverage of 95%.

As shown in Figure 11, in the static WSN, the most sensors were required; in the mobile WSN, the least sensors were

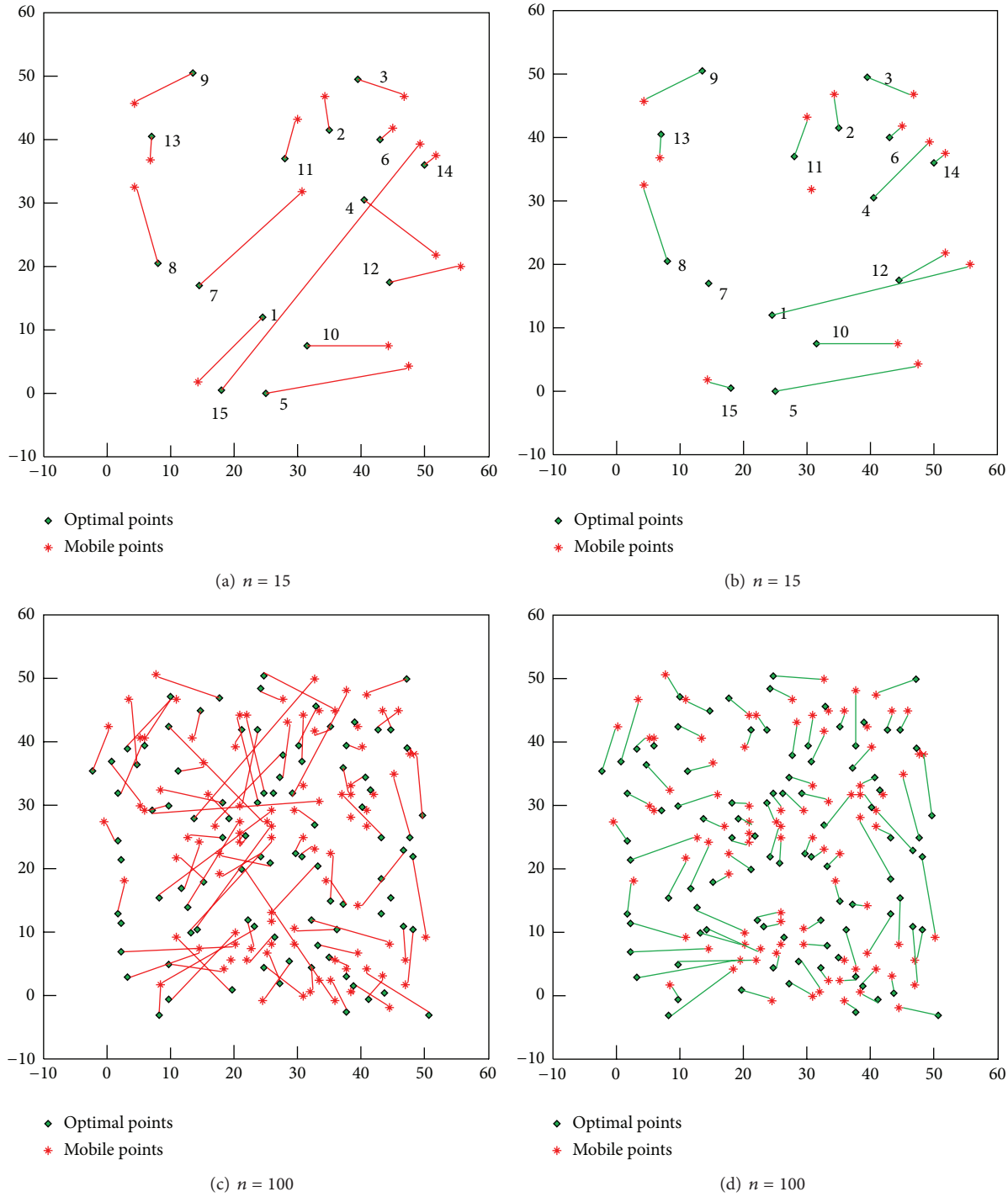


FIGURE 10: (a) and (c) are the initial trace, and (b) and (d) are the trace that have been optimized.

required; in the mixed WSN, the number of sensors required was between the number required for the static WSN and the number required for the mobile WSN. Obviously, the mixed WSN significantly reduced the number of sensors compared to the static WSN. When the percentage of mobile sensors was 30%, the number of sensors in the mixed WSN was about half of the number of sensors in the static WSN, and as the proportion of mobile sensors increased, the number of sensors needed decreased. The number of sensors required in a mobile WSN was less than the numbers required by static or mixed WSNs. But as the number of mobile sensors incraesed, the marginal effect decreased; fore example, when the percentage of mobile sensors increased from 0% to 10%, the number of sensors for 98% coverage decreased from 120 to 85,

Input: n lines with total length of L
Output: n lines with total length of $L'' < L$ and the variance of $L' <$ the variance of L
$L'' = L$;

```
(1)  do {
(2)  for i = 1: n − 1
(3)       for j = i + 1: n
(4)            exchange the destination of line i and line j;
(5)            if (the total length of the two lines decreased)
(6)               record the result;
(7)            else
(8)               maintain the present status;
(9)       end
(10) end
(11) L' = L'';
(12) L'' = the total length of n new lines;
(13)       } while (L' − L'' > ε);
```

where ε is a threshold, it represent the precision of the result we got.

Algorithm 3: 2-exchange optimization algorithm.

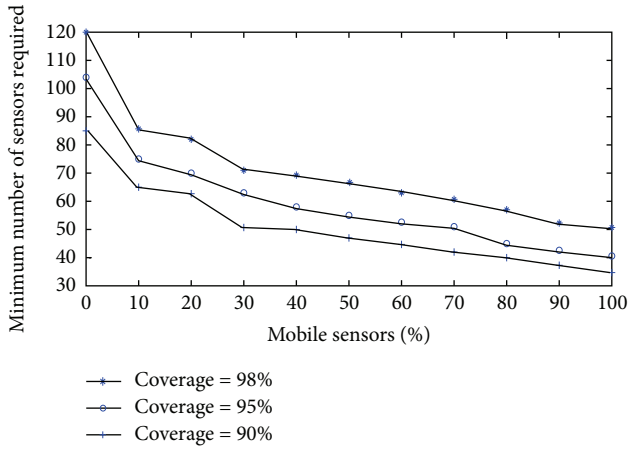


Figure 11: The minimum number of sensors required to reach a coverage percent in different mobile percentage.

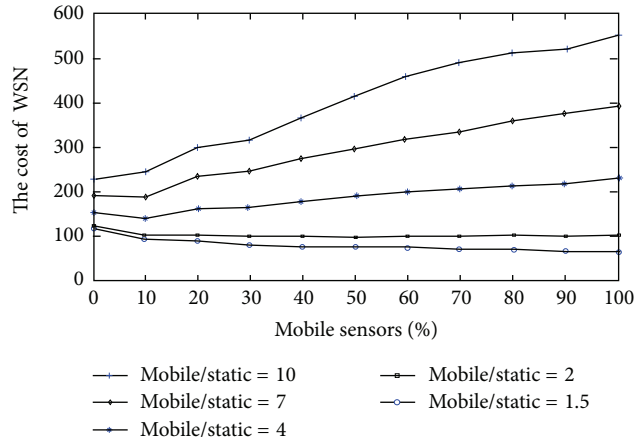


Figure 12: The cost of sensors to reach 95% coverage.

but when the percentage incraesed from 50% to 60%, the effect was not as significant as before. In addition, the price of mobile sensors is higher than the price of static sendors; thus, determing which WSN is the cheapest is related to the ratio of the price of mobile sensors to the price of static sensors. In conclusion, none of the three types of WSN is always the best in any situation.

Figure 12 shows the costs of these three types of WSN to reach 95% coverage. Based on different price ratios of mobile sensors to static sensors, the overall cost of a WSN can be somewhat different. When the ratio is low (e.g., ≤1.5), increasing the percentage of mobile sensors can reduce the overall cost, and the mobile WSN is the best choice. When the ratio is higher (e.g., 2 ≤ the ratio ≤ 7), the mixed WSN has the lowest cost, and the percentage of mobile sensors is not very high (e.g., 10% ≤ the percentage of mobile sensors ≤ 50%). But if the mobile sensor is much more expensive (e.g., the ratio > 7), the static WSN is the best choice.

If the ratio between the price of the mobile sensor and the price of the static sensor is 2–7, the mixed WSN is a good choice. Under this circumstance, our centralized algorithm is useful, and the result is a balance between the cost and the coverage.

5.2. Coverage. The bidding protocol and the proxy-based bidding protocol are two classic protocols in redeploying mobile sensors in a mixed WSN, and we compared our results to the results of these protocols. From Section 5.1, we know that, most of the time, the percentage of mobile sensors cannot be greater than 50%, so we only considered these situations.

As shown in Figure 13, the mixed WSN can improve the coverage significantly; if the WSN is static, the coverage

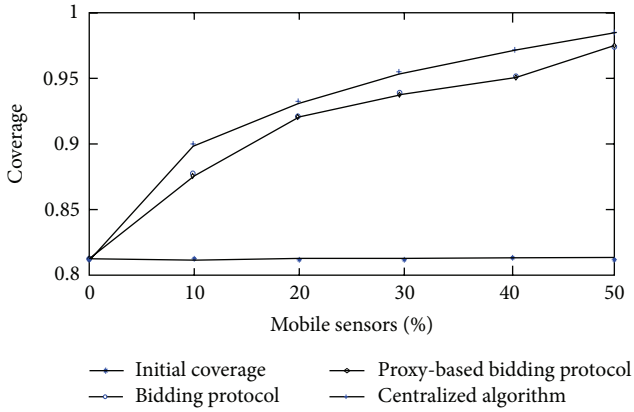


Figure 13: Coverage by different algorithms.

percentage is about 80% and the mixed WSN is about 90 to 98%. In terms of coverage, there is no preference between the bidding protocol and the proxy-based bidding protocol. In our centralized algorithm, the coverage was higher than it was in the bidding protocols. In fact, the bidding protocols provided the same coverage as the greedy algorithm, and our encroaching principle was better than the greedy algorithm.

5.3. Energy Efficiency. Our algorithm is centralized, and it is executed in the management node. After the algorithm has terminated, the management node will issue the result to the mobile nodes only once, and the mobile nodes will move to the proper position to improve the coverage. We should notice that the management node is not strictly energy limited, and there are few communications between the nodes in the WSN. So, the total energy consumption is just associated with mechanical movement, and we used the average distance that the sensors were moved to determine the energy consumption.

Figure 14 shows the average distance that the sensors were moved in the three protocols. It is apparent that the average distance of basic bidding protocol is the longest and that the result of the centralized algorithm is the best. Obviously, the reason is that the nodes in the distributed algorithm just have part of the geographical information of all the nodes and the management node has all the geography information of all of sensors. Because it has more information, the centralized algorithm produced good results, and it is a global optimization. Conversely, the results of the distributed algorithm were not very good because it had less information, and the results comprised only a partial optimization. Compared with basic-bidding protocol, the results of the proxy-based protocol were better.

In the distributed algorithm, the total energy consumption is comprised of two parts, that is, mechanical movement and communication. It is evident from Figure 14 that the total energy consumption of the centralized algorithm is less than just the mechanical movement in the distributed algorithm, so it is considerably less than the sum of the two parts.

Thus, our protocol is much more energy efficient than the two distributed protocols, that is, the basic bidding protocol and the proxy-based bidding protocol.

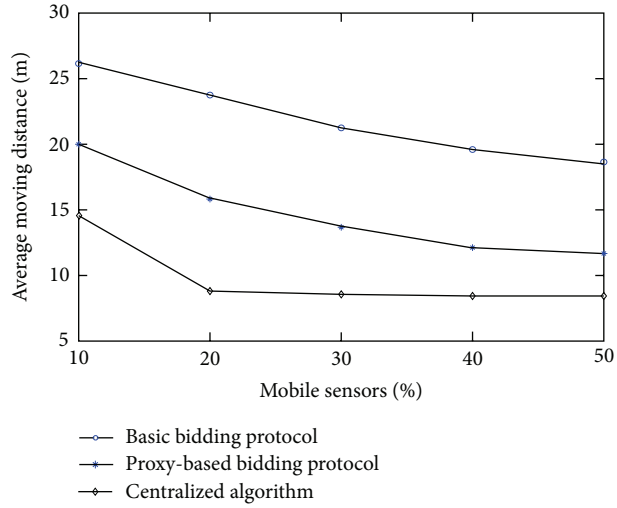


Figure 14: Average moving distance.

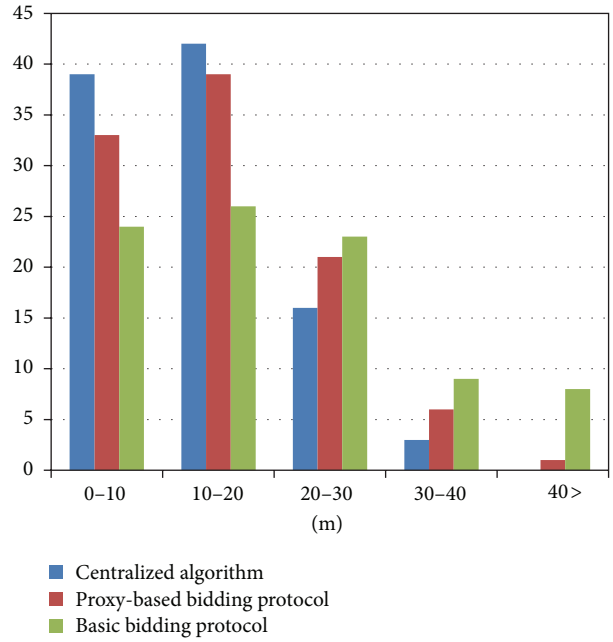


Figure 15: Statistic of the moving distance for the three protocols.

5.4. Load Balance. To show the results more clearly, we increased the number of mobile sensors to 100 and generated 100 coverage holes randomly. Then, we matched them using the centralized algorithm and gathered statistics related to the distances that the sensors were moved. We used the variance of the 100 distances to measure the load balance. The smaller the variance, the more balanced the load is.

As shown in Figure 15 for the centralized algorithm, most of the distances that the mobile sensors were moved were between zero and 30 m, and the distances were more concentrated than they were in the other two protocols. The variance of the distances for the centralized algorithm was 13.38, which was smaller than 78.25, the variance of the proxy-based bidding protocol, and 127.34, the variance of the basic

bidding protocol. The centralized algorithm performed very well for load balancing, and it prolonged the lifetime of the WSN.

6. Conclusions and Future Work

In this paper, we proposed a novel, centralized algorithm to deploy a mixture of mobile and static sensors to construct sensor networks to provide the required uniform sensing service in harsh environments. We used Delaunay triangulation rather than a Voronoi diagram to detect the coverage holes, because it was easier and provided equivalent results. Our performance evaluation showed that for the same conditions, the centralized algorithm was better than the distributed algorithm in coverage percentage, energy-efficiency, and load balancing. Also, we found that it can prolong the lifetime of the WSN significantly because of the energy efficiency and load balance. The reason of the centralized algorithm being better than the distributed algorithm was that it has the overall topology of the WSN, which, in theory, allows us to obtain the global optimum solution.

As future work, we will improve the algorithm in two ways, that is, (1) in order to improve the precision of the coverage, we plan to introduce the probabilistic disk model which is more accurate instead of the classical Boolean sensing model and (2) to further improve the performance of our matching algorithm which is the core of this paper; we plan to introduce some models in Graph Theory and Operations Research Theory.

Acknowledgments

This research is supported by National Natural Science Foundation of China under Grant 61071076, the Academic Discipline and Postgraduate Education Project of Beijing Municipal Commission of Education, National High-tech Research and Development Plans (863 Program) under Grant 2011AA010104-2.

References

[1] J. Heo, J. Hong, and Y. Cho, "EARQ: energy aware routing for real-time and reliable communication in wireless industrial sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 1, pp. 3–11, 2009.

[2] Q. Li, L. Cui, B. Zhang, and Z. Fan, "A low energy intelligent clustering protocol for wireless sensor networks," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '10)*, pp. 1675–1682, March 2010.

[3] A. Luntovskyy, V. Vasyutynskyy, and K. Kabitzsch, "Propagation modeling and placement algorithms for wireless sensor networks," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '10)*, pp. 3493–3497, Bari, Italy, July 2010.

[4] S. Nouh, R. A. Abbass, D. A. E. Seoud et al., "Effect of node distributions on lifetime of Wireless Sensor Networks," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '10)*, pp. 434–439, Bari, Italy, July 2010.

[5] Y. Z. Zhao, T. N. Nguyen, M. Ma, and C. Miao, "An energy-efficient MAC protocol with adaptive scheduling for wireless sensor networks," in *Proceedings of the 7th IEEE International Conference on Industrial Informatics (INDIN '09)*, pp. 446–451, June 2009.

[6] Y. Lin, X. Hu, and J. Zhang, "An ant-colony-system-based activity scheduling method for the lifetime maximization of heterogeneous wireless sensor networks," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pp. 23–30, July 2010.

[7] J. Chen, J. Li, S. He, Y. Sun, and H. H. Chen, "Energy-efficient coverage based on probabilistic sensing model in wireless sensor networks," *IEEE Communications Letters*, vol. 14, no. 9, pp. 833–835, 2010.

[8] J. Jia, J. Chen, G. Chang, and Z. Tan, "Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm," *Computers & Mathematics with Applications*, vol. 57, no. 11-12, pp. 1756–1766, 2009.

[9] Y. Lin, X. M. Hu, J. Zhang, O. Liu, and H. L. Liu, "Optimal node scheduling for the lifetime maximization of two-tier wireless sensor networks," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1–8, July 2010.

[10] P. C. Wang, T. W. Hou, and R. H. Yan, "Maintaining coverage by progressive crystal-lattice permutation in mobile wireless sensor networks," in *Proceedings of the 2nd IEEE International Conference on Systems and Networks Communications (ICSNC '06)*, pp. 1–6, Tahiti, Polynesia, November 2006.

[11] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 35, no. 1, pp. 78–92, 2005.

[12] Y. C. Wang, C. C. Hu, and Y. C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Transactions on Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.

[13] Y. C. Wang and Y. C. Tseng, "Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1280–1294, 2008.

[14] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 1, pp. 61–91, 2004.

[15] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Symposium on Distributed Autonomous Robotic System (DARS '02)*, pp. 1–10, 2002.

[16] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, "Mobility limited flip-based sensor networks deployment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 2, pp. 199–211, 2007.

[17] S. Chellappan, W. Gu, X. Bai, D. Xuan, B. Ma, and K. Zhang, "Deploying wireless sensor networks under limited mobility constraints," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1142–1157, 2007.

[18] Z. Jiang, J. Wu, R. Kline, and J. Krantz, "Mobility control for complete coverage in wireless sensor networks," in *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pp. 291–296, Beijing, China, June 2008.

[19] S. Yang, J. Wu, and F. Dai, "Localized movement-assisted sensor deployment in wireless sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Sysetems (MASS '06)*, pp. 753–758, Vancouver, Canada, October 2006.

[20] A. Ghosh, "Estimating coverage holes and enhancing coverage in mixed sensor networks," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 68–76, November 2004.

[21] G. Wang, G. Cao, and T. F. La Porta, "A Bidding protocol for deploying mobile sensors," in *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP '03)*, November 2003.

[22] G. Wang, G. Cao, P. Berman, and T. F. La Porta, "Bidding protocols for deploying mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 515–528, 2007.

[23] G. Dommety and R. Jain, "Potential networking applications of global positioning systems (GPS)," Tech. Rep. TR-24, Department of Computer Science and Engineering, Ohio State University, 1996.

[24] E. D. Kaplan, *Understanding GPS: Principles and Applications*, Artech House, 1996.

[25] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 81–95, September 2003.

[26] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 45–57, September 2004.

[27] K. F. Ssu, C. H. Ou, and H. C. Jiau, "Localization with mobile anchor points in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 1187–1197, 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 913497, 12 pages
http://dx.doi.org/10.1155/2013/913497

Research Article

A Self-Adaptive Regression-Based Multivariate Data Compression Scheme with Error Bound in Wireless Sensor Networks

Jianming Zhang,[1] Kun Yang,[2] Lingyun Xiang,[1] Yuansheng Luo,[1] Bing Xiong,[1] and Qiang Tang[1]

[1] *School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China*
[2] *School of Computer Science and Electronic Engineering, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK*

Correspondence should be addressed to Jianming Zhang; jmzhang@csust.edu.cn

Received 6 January 2013; Accepted 24 February 2013

Academic Editor: Yan Zhang

Wireless sensor networks (WSNs) have limited energy and transmission capacity, so data compression techniques have extensive applications. A sensor node with multiple sensing units is called a multimodal or multivariate node. For multivariate stream on a sensor node, some data streams are elected as the base functions according to the correlation coefficient matrix, and the other streams from the same node can be expressed in relation to one of these base functions using linear regression. By designing an incremental algorithm for computing regression coefficients, a multivariate data compression scheme based on self-adaptive regression with infinite norm error bound for WSNs is proposed. According to error bounds and compression incomes, the self-adaption means that the proposed algorithms make decisions automatically to transmit raw data or regression coefficients, and to select the number of data involved in regression. The algorithms in the scheme can simultaneously explore the temporal and multivariate correlations among the sensory data. Theoretically and experimentally, it is concluded that the proposed algorithms can effectively exploit the correlations on the same sensor node and achieve significant reduction in data transmission. Furthermore, the algorithms perform consistently well even when multivariate stream data correlations are less obvious or non-stationary.

1. Introduction

Wireless sensor networks (WSNs) can monitor, sense, and collect the data of various environments or monitored objects in an area. And the data are eventually sent to the target users [1]. WSNs have wide range of potential applications including home area and smart grid [2]. Each sensor node in WSN has limited battery power supply, and the used batteries are hard to recharge and replace. It is infeasible that a large number of nodes directly transmit the collected raw data to the base station or sink due to limited bandwidth and battery capacity. Wireless communication consumes most of the power. The literature [3] shows that the power consumed by transmitting the 1-bit data over a 100-meter distance can support to execute 3000 CPU instructions. The literature [4] also points out that the power consumption of the data transmission is much higher than that of data processing. Transmitting a 1-bit data via radio medium is at least 480 times the power consumption used for executing one "addition" operation. Approximately, 70% of the total power is consumed in data transmission. How to effectively reduce the amount of data within the WSN in order to extend the network lifetime is an important issue. Sensor energy can be saved via mechanisms at different layers of the WSN protocol stack [5–8], such as energy efficient routing [9] and battery saving media access control [10], where there are plenty of existing work. L. Zhang and Y. Zhang [11] presented an energy-efficient packet forwarding protocol in wireless sensor networks, considering channel awareness and geographic information. The perspective of this paper is from the application layer by introducing effective data compression.

Sensor nodes use the data sensing units to collect raw data and then transmit the data processed through the data processing units to reduce the amount of data to be

transmitted. Generally, there are two types of processing methods [12]: data aggregation and data approximation. Aggregate functions process the sampled data using some forms of simple statistics such as maximum, minimum, and average. It is an effective mean to reduce the volume of data. However, it just provides simple coarse statistical information while potentially hiding some interesting local varieties of the data. Data approximation can be regarded as a model-based data processing method. When data feeds exhibit a large degree of redundancy, approximation is a less intrusive form of data reduction in which the underlying data feed is replaced by an approximate signal tailored to the application needs. It is only required to transmit the parameters of the distributed model built for the sensor data collected from WSN. Thereby, the amount of transmitted data is greatly reduced, resulting in network energy being saved, and thus the network lifetime is being prolonged. In terms of the way they are applied, there are four major categories for data approximation: probabilistic model [13, 14], time series analysis model [15–17], data mining model [18], and data compression model [19–24].

The paper is based on the following three facts which also show the application scenario. (1) *Multivariate correlation exists.* With the development of the wireless communication and microelectronics technology, wireless sensor node equipped with several sensing units has become popular. Such node can simultaneously monitor several types of data, such as sound intensity, acceleration, temperature, humidity, light intensity, and video, in which certain correlation universally exists. In this paper, the data collected by different sensing units in the same sensor node are referred to as multivariate data or multiattribute data, and the correlation among these different data type is called multivariate correlation or multiattribute correlation. (2) *Spatial correlation does not exist.* Most distributed compression algorithms are based on the assumption that nodes have spatial correlations with other nodes. This introduces large implementation cost. In real life, people often hope to use as small number of sensor nodes to monitor as wide an area as possible in order to reduce investment cost. As a result, these nodes are placed far away from each other, leading to no or little of unstable spatial correlation. In this case, it is a better choice that the algorithm is designed to run independently on each node. (3) *Error is bounded*. Influencing by noise, node failure, unreliable wireless communication, power constraints, and other factors, it exists certain error in acquisition, processing, and transmission processes of sensor data. The sensor data has a certain degree of uncertainty. If the user does not need very accurate results, by sacrificing the data precision, he can achieve the purpose of reducing the data amount of communication. Our study is devoted to minimize the volume of the transmitted data in the error bound predefined by the user.

The major contribution of the paper lies in the following aspects. This paper designs the algorithms considering the temporal and multivariate correlations of the data and also taking into account the case when the multivariate correlation is unstable, but without considering the spatial correlation of the data. Our algorithms run in each sensor node collecting a number of data streams. This paper proposes a self-adaptive regression-based multivariate data compression algorithm with error bound (denoted as AR-MWCEB) and implements it in C. According to the predetermined error bounds, AR-MWCEB makes decision automatically to transmit raw data or regression coefficients and to select the number of data involved in each regression. The results of simulation experiments show that the algorithm can reduce the amount of transmitted data and has better real-time performance than the benchmark algorithms.

The rest of the paper is organized as follows. Section 2 describes the related work, which is followed by the introduction of some preliminary knowledge and the problem formulation in Section 3. Section 4 presents the proposed compression scheme, base selection algorithm, and incremental calculation of regression coefficient. A self-adaptive regression-based multivariate data compression algorithm with error bound is detailed in Section 5. After the presentation of the intensive simulation results and performance analysis in Section 6, the paper is concluded in Section 7.

2. Related Work

Multivariate streams measured from one sensor node are correlated. The same is often true in other domain. Deligiannakis et al. [12] pointed out that the scatter diagram with indexes of industry and insurance from the New York stock market as the x-axis and y-axis coordinates, respectively, appears to be an approximated straight line. Each original time series is not linear, but he proposed a SBR algorithm using the base signal as an independent variable and the regression model to piecewise approximate other series. The values of the base signal are extracted from the real measurements and maintained dynamically as data changes. When the multivariate correlation measured by a sensor node is larger, this algorithm is better. But SBR algorithm does not consider the problem of error bound, and it just compresses data in the maximum degree under satisfying the condition of data compression. It may lead to two problems: (1) the algorithm is terminated before the error margin reaches its predetermined requirement; (2) the data is still continuously compressed when the error has met the requirement.

For the data generated by the single sensor node, the RACE algorithm [19] proposes a Haar wavelet compression algorithm with adaptive bit rate. It can output CBR (constant bit rate) or LBR (limited bit rate) streams by selecting the significant wavelet coefficients based on a threshold. RACE runs on a single node. In spite of reducing the transmission of redundant data by eliminating the temporal correlation, it does not consider the spatial correlation among neighboring nodes or multiattribute correlation in the same node. Ciancio et al. [20] study a distributed wavelet compression algorithm, which exchanges information among neighboring nodes and distributes the discovered spatial correlation of the sensor network data before the data are transmitted to the sink node. Although the algorithm has greatly reduced the transmission of redundant data, however, information exchange among nodes would result in some cost, such as power consumption

and network delay, which needs further quantitative analysis in theory.

The literature [21] designs a new algorithm based on multiattribute correlation. The algorithm can effectively reduce spatial, temporal, and multivariate correlations, but there are two problems limiting its performance. (1) All the raw data of each node in a cluster must be sent directly to the cluster head and must be processed in the cluster head. The data with different attributes but from different nodes are not differentiated, but rather they are abstracted into a column of the processed data matrix. (2) Before sending data, the cluster head must call data preprocessing algorithm to analyze and find out the attribute pairs between which the correlation is large. In this processing, the estimated attribute data are fitted using the least square method. Our previous algorithms proposed in [22] run independently in each of the sensor nodes, and no collaboration exists between nodes. It uses the single data stream wavelet compression algorithm with error bound (SWCEB) to do wavelet decomposition to the maximum level resulting in full elimination of the temporal correlation of the data to satisfy the error bound in the coefficient selection. Meanwhile, it eliminates multiattribute correlations and uses the regression-based multiple data streams wavelet compression algorithm with error bound (MWCEB), in which the regression intervals are continuously bisected if needed to ensure that the error is bounded. The binary partition is arbitrary, so this paper will try to self-adaptively determine the number of data involved in each regression calculation.

It is worth mentioning that both multiple data streams (the data streams collected by a cluster head from all other nodes in the cluster) [23] and multiattribute data stream (the data streams collected by the single sensor node having multiple sensing units) can be effectively compressed by our proposed AR-MWCEB scheme. Furthermore, after the multiple data streams being processed by the cluster head, its spatial correlation is reduced, and after the multiattribute data stream is being processed in the multimodal node, its multivariate correlation is also reduced. This paper focuses on reducing the multivariate correlation of multiattribute data stream, but the result can also improve the performance of DLRDG [23].

Sadler and Martonosi [24] propose a lossless compression algorithm for sensor nodes in WSNs, called S-LZW. They discuss some design problems in detail about implementation, adaptability improvement, customizing compression techniques, and so on. Research on lossless data compression in WSNs is still in its early stage with very few published papers.

3. Preliminaries and Problem Statement

Many monitoring data collected by sensor nodes such as temperature, humidity, light, and vibration always slightly varies within a continuous time, and most of successive data are the same or similar. When these successive data are decomposed by wavelet transform, the majority of the energy is concentrated in the low-frequency coefficients. High-frequency coefficients are 0 or close to 0. Even if the monitored object changes abnormally, which causes unusual fluctuations of sensor data, the multiresolution characteristic of the wavelet transform can ease the impact of these unusual fluctuations on the overall data. It maintains values of some detail components as approximately 0. Compressing (discarding) the detail components with value 0 does not affect the data reconstruction; compressing the nonzero detail components would affect the data accuracy. The more detail components are compressed, the higher the compression ratio of the data is, but the larger the caused data error is. Therefore, under the premise of guarantee of error bound, detail components should be compressed maximally.

The distributed wavelet compression algorithm for the sensor networks is designed to complete the task by multiple nodes together. In this algorithm, the wavelet transform is calculated dispersedly in each node, and the wavelet coefficients are also dispersed at each node. Thereby, the amount of calculation for each node is small. The performance of using two-level wavelet transform is slightly better than that of using single-level transform, because the second level of the wavelet transform can better eliminate the correlation, in spite of increasing the additional energy consumption and time delay. Communication overhead and time delay are increased constantly with increasing the level of wavelet decomposition; thus, the distributed compression algorithm cannot always rely on increasing the wavelet decomposition level to improve its performance. Our algorithm runs independently on each sensor node, there is no collaboration among nodes; therefore, wavelet transform can decomposes the data with the maximum level to eliminate the temporal correlation of enough data.

Garofalakis and Gibbons [25] firstly proposed a wavelet-based compression technique with error guarantees of data reconstruction by introducing probabilistic wavelet synopses. Based on the error tree from the literatures [25, 26], a single-attribute data wavelet compression algorithm with error bound named SWCEB is proposed by us in the literature [22]. It includes four steps: wavelet decomposition, coefficient selection, quantization, and entropy coding. In the step of coefficient selection, it guarantees to satisfy the error bound. And it eliminates the temporal correlation in a single data stream through Haar wavelet transform. Furthermore, literature [22] proposes a multiattribute data compression algorithm with error bound named MWCEB, which is based on regression and divided into three steps:

(i) selecting some attributes as the base attributes according to the correlation coefficient matrix of multiattribute sampling data;

(ii) using SWCEB algorithm to compress the base attributes data;

(iii) describing other nonbase attributes data by linear regression coefficients of one of the base attributes.

MWCEB algorithm can guarantee the satisfaction of error bound by increasing the number of base attributes. However, in the worst case, it degenerates into the SWCEB algorithm without taking advantage of multiattribute correlation. Another way to reduce error for MWCEB algorithm is

to reduce the number of data involved in the regression calculation, that is, to carry out piecewise linear regression. As MWCEB algorithm arbitrarily divides the regression intervals equally without considering the data correlation, its processing parameters required manual intervention, and its regression performance is not ideal.

Within the framework of the above processing procedure, this paper tries to self-adaptively determine the number of data involved in the regression calculation each time to guarantee that the error is bounded. If successive data is a dramatic change, it is difficult to use a linear regression model to describe the nonbase attribute. Thus, the self-adaptive piecewise linear regression which automatically determines the data number involved in the regression calculation each time or direct transmission of the raw sampling data is selected automatically.

Assuming the data buffer size of a sensor node is M, the collected data is denoted as $s[0], \ldots, s[M-1]$, and the reconstructed approximation data is $s^*[0], \ldots, s^*[M-1]$. Dealing with multiattribute data, it usually uses normalized infinite norm error.

Definition 1 (normalization of the sampling data). $s[i]$ is normalized to $\text{norm}(s[i]) = (s[i] - s_{\min})/(s_{\max} - s_{\min})$, where $s_{\max}$ and $s_{\min}$ are the maximum and minimum values of the sampling data collected in a period, respectively. Obviously, the value of $\text{norm}(s[i])$ is located in $[0, 1]$. When dealing with multiattribute data, the normalization of sampling data can prevent the attribute with small amplitude being concealed by the ones with large amplitude.

Definition 2 (normalization error). $e_i^{\text{norm}} = |\text{norm}(s[i]) - \text{norm}(s^*[i])|$.

Definition 3 (∞-norm average error). $\|e\|_\infty = \max_{0 \le i < M} |s[i] - s^*[i]|$. ∞-norm average error bound guarantees the error bound of each reconstructed data and the corresponding sampling data. The normalized infinite norm average error is defined as $\|e\|_\infty^{\text{norm}} = (\max_{0 \le i < M} |s[i] - s^*[i]|)/(s_{\max} - s_{\min})$.

Definition 4 (the correlation coefficient matrix). M times samples x_i, $i = 0 \cdots M-1$ of an attribute can be recognized as M times experiments of a discrete random variable $\mathbf{X}$. The relationship between the attributes $\mathbf{X}$ and $\mathbf{Y}$ can be measured by the correlation coefficient, which is defined as follows:

$$r_{xy} = \frac{\text{Cov}(X, Y)}{\sqrt{D(X)}\sqrt{D(Y)}} = \frac{\sum_{i=0}^{M-1} \left[(x_i - E(X))(y_i - E(Y)) \right]}{\sqrt{\sum_{i=0}^{M-1} (x_i - E(X))^2} \sqrt{\sum_{i=0}^{M-1} (y_i - E(Y))^2}}, \quad (1)$$

where $E(X) = \sum_i p_i x_i = (1/M) \sum_{i=0}^{M-1} x_i$. If $\mathbf{X}$ and $\mathbf{Y}$ are positive (negative) correlations, r is positive (negative); when $r = 1(-1)$, their relationship is complete positive (negative) correlation, and all the data points lie on the regression line. The more scattered the data points are, the smaller the absolute value of r is, and the lower the correlation is. If each node can collect N attributes, the correlation coefficient matrix R with the size $N \times N$ is defined to express the relationship among all attributes, in which the element of the jth column in the ith row represents correlation coefficient between the ith and the jth attributes.

Definition 5 (the best correlation of attribute X_j for the base attributes set). Suppose that N attributes streams X_0, $X_1, \ldots, X_{N-1}$ have some correlations, and that they have been classified into the base attributes set *BaseSet* and the candidate attributes set *CandSet*, where the index used in both sets represents the corresponding stream. The best correlation between the element X_j in *CandSet* and all of the elements in *BaseSet* is defined as follows:

$$bestfit_j = \max_i \left(\left| r_{ij} \right| \right), \quad i \in BaseSet. \quad (2)$$

Then, an element X_j in *CandSet* can be represented by regression coefficients of using the element X_i (recorded in variable $position_j$) in *BaseSet*, in the condition that the absolute value of the correlation coefficient between these two elements is the largest. If the error is too large, the element would move to *BaseSet*. Obviously, the best correlation of the elements in the base attributes set is 1.

Definition 6 (the expected income of adding candidate attribute X_j to the base attributes set). The N attributes are divided into the base attributes set *BaseSet* and the candidate attributes set *CandSet*. If attribute X_j is added to *BaseSet*, the expected income is defined as:

$$income_j = \sum_k \left(\left| r_{jk} \right| - bestfit_k \right), \quad k \in \left\{ k \in CandSet \wedge \left| r_{jk} \right| > bestfit_k \right\}. \quad (3)$$

Initially, *CandSet* contains N attributes, and *BaseSet* is empty. After summing the absolute value of the elements in the correlation coefficient matrix row by row, choose the attribute corresponding to the maximum sum and add it to *BaseSet* as a base. Then, according to error bound, decide if you will continue to look for other bases. In the process of each base selection, add the X_j from *CandSet* with the largest expected income to *BaseSet*; at the same time, update $bestfit_i$ of the remaining attribute X_i in *CandSet*, and correspondingly update $position_i$ to express that we regress the X_i in *CandSet* on the X_j in *BaseSet*.

4. Base Selection and Coefficient Incremental Calculation for Regression

4.1. Base Selection and Linear Regression. Suppose that a sensor node has N sensing units, and the collected attribute is A_j, $j = 1, 2, \ldots, N$. The overall operation of the regression-based compression scheme is as follows.

Step 1. Using our base selection algorithm proposed in [22], classify all attributes into several groups G_i, $i = 1, 2, \ldots, K$, according to their correlation coefficient matrix. $\bigcup_{i=1}^{K} G_i =$

```
(1) typedef struct {
(2)     double sum_x;
(3)     double sum_xx;
(4)     double sum_xy;
(5)     double sum_y;
(6) }COEFF;
(7) COEFF coeff, firsthalf_coeff, *pcoeff;
```

Algorithm 1

$\bigcup_{j=1}^{N} A_j$ and for all $i \neq j, G_i \bigcap G_j = \emptyset$. The attributes with large correlation are located in the same group. Definitions 5 and 6 in Section 3 have implicitly described our base selection method. The base selection algorithm can also be implemented by consulting the K-Means and other clustering algorithms.

Each group $G_i, i = 1, 2, \ldots, K$, has only one base attribute, and other attributes in the same group are represented by the base and linear regression coefficients. For the convenience of description, denote the selected base in a group as attribute X and a nonbase attribute located in the same group with attribute X as attribute Y. The attribute Y can be represented by several regression coefficient pairs generated by linear regression.

Step 2. Run the SWCEB algorithm on M sampling data of the base attribute X. After wavelet decomposition, set some wavelet coefficients to 0 in the case of error bound. The local reconstructed data by wavelet transform is directly used as the base signal X of the regression. As a result, it avoids to spread the base reconstruction error to other nonbase attributes, and the reconstructed base signal is more regular and more suitable for use as a base than the original one.

Step 3. Regress the nonbase attributes on the base attribute from the group in which the nonbase attributes locate, and transmit the calculated regression coefficients. The raw data of the nonbase attributes is no longer required to be transmitted.

The implementation of Step 3 is analyzed below in details. When the estimation of nonbase attribute Y is $\widetilde{Y} = aX + b$, find the regression coefficients a and b so that $||Y - \widetilde{Y}||_2$ is minimum; namely, minimize the objective function $Q = \sum_i (y_i - ax_i - b)^2$. When

$$\frac{\partial Q}{\partial a} = 0, \qquad \frac{\partial Q}{\partial b} = 0 \tag{4}$$

solve the linear equations with two unknowns

$$\begin{aligned} \sum_i (y_i - ax_i - b)(-x_i) &= 0, \\ \sum_i (y_i - ax_i - b)(-1) &= 0, \end{aligned} \tag{5}$$

to find a, b.

Then,

$$\begin{aligned} a &= \frac{M * \sum_{i=1}^{M} (x_i * y_i) - \sum_{i=1}^{M} x_i * \sum_{i=1}^{M} y_i}{M * \sum_{i=1}^{M} x_i^2 - \left(\sum_{i=1}^{M} x_i\right)^2}, \\ b &= \frac{\sum_{i=1}^{M} y_i - a * \sum_{i=1}^{M} x_i}{M}. \end{aligned} \tag{6}$$

In this way, a candidate attribute can be represented by a pair of regression coefficients. The base attribute is compressed by Haar wavelet in Step 2, and the number of data being processed each time is some power of 2. It is worth noting that the more the number of data is involved in each regression, the bigger the regression error is. According to the number M of the processed data each time, our proposed MWCEB algorithm employs the piecewise linear regression with the equal number of data (obviously, this will lead to a large error), and simply transmits the regression coefficients a and b. If $M = 2$, then the regression would not cause any compression; furthermore, in order to achieve the convenience of wavelet compression of the selected base, the number of data is also preferable to be as some power of 2, so the number of processed data in each time should be 4 at least.

4.2. Incremental Calculation for Regression. Now, we will introduce how to use self-adaptive regression to make decisions automatically to transmit raw data or regression coefficients, and how to automatically obtain the number of the data involved in each regression calculation. The number of data in each regression is set to some power of 2 for convenience. If the regression results satisfy the error bound, the start number *start* and the count number *length* of the data participating in the regression should be transmitted, as well as the regression coefficients a and b. If $length = 4$, then the regression would not cause any compression, so *length* should be greater than or equal to 8. If the regression results of 8 (or some power of 2 greater than 8) data satisfy the error bound, only 4 data on representation of regression results need to be transmitted; Otherwise, the raw sampling data is directly transmitted. In order to obtain the best compression performance, *length* is doubled constantly until find the maximum number of regressed data in the condition of error bound.

Through analyzing the equations of calculating the regression coefficients a and b, it is found that the regression coefficients can be incrementally calculated. Define a data structure as in Algorithm 1.

```
(1)  Function AuxCalc (int start , int length  COEFF *pcoeff )
(2)  input: the sampling data of related attributes X[start ··· start+length-1], Y[start ··· start+length-1]
(3)  output: coeff
(4)  begin
(5)     pcoeff ->sum_x=0;
(6)     pcoeff->sum_xx=0;
(7)     pcoeff->sum_xy=0;
(8)     pcoeff->sum_y=0;
(9)     for i=start to start+length-1 {
(10)        pcoeff->sum_x += X[i];
(11)        pcoeff->sum_xx += X[i]∗X[i];
(12)        pcoeff->sum_xy += X[i]∗Y[i];
(13)        pcoeff->sum_y += Y[i];   }
(14) end
```

Algorithm 2: The AuxCalc function.

```
(1) Function IncRegress (int start , int length , double eps , double *a , double *b , int *cnterr )
(2) input: the predefined regression error bound eps; the sampling data of related attributes X[start ··· start+length-1],
    Y[start.. start+length-1]
(3) output: the regression coefficients a and b; the number errcnt of data exceeding the error bound
(4) begin
(5)   if length==8 then
(6)      AuxCalc (start, length, &coeff);
(7)   else{
(8)      AuxCalc (start+ length/2, length/2, &coeff);    // length/2 data does not need to be summed once again
(9)      coeff += firsthalf_coeff; }
(10)  firsthalf_coeff = coeff;
(11)  *a= (length *coeff.sum_xy - coeff.sum_x*coeff.sum_y)/(length *coeff.sum_xx - coeff.sum_x*coeff.sum_x);
(12)  *b= (coeff.sum_y - *a*coeff.sum_x)/length; // recursive form of (6)
(13)  *cnterr = 0;
(14)  double max_y=−32768, min_y = 32767;
(15)  for i=0 to M-1{
(16)     if max_y < Y[i] then max_y= Y[i];
(17)     if min_y > Y[i] then min_y=Y[i]; }
(18)  for i=start to start+length-1{
(19)     if fabs(*a *X[i]+ *b-Y[i])> eps *(max_y - min_y) then (*cnterr)++;   }
(20) end
```

Algorithm 3: The IncRegress function.

The *AuxCalc* is an auxiliary function used by the incremental regression, which is implemented as in Algorithm 2.

When the regression results satisfy the error bound condition, the number of regressed data length is doubled repeatedly for exploring the maximum *length*. Known by analysis of (6), when *length* is increased to 2 times of the raw one, the calculated *coeff* in the previous regression calculation can still be effectively used. Therefore, they can be saved as a static variable or global variable *firsthalf_coeff* for the use in the next regression calculation. In addition, assignment operators for the variables of COEFF type can be implemented by the addition and assignment of each corresponding fields.

The incremental calculation function of regression coefficients is described as in Algorithm 3.

5. A Self-Adaptive Regression-Based Multivariate Data Compression Algorithm with Error Bound

5.1. The Proposed Algorithm. The proposed self-adaptive regression-based multivariate data wavelet compression scheme with error bound is abbreviated to AR-MWCEB. Its basic idea is as follows. (1) Calculate the regression error of the first 8 being processed data in the buffer of a sensor node. (2) If the calculated regression error does not satisfy the predefined error bound, transmit directly these 8 raw data, or else, and recalculate the regression error after doubling the number of regressed data until that the maximum number of the regressed data with satisfaction of the predefined error bound is found in order to obtain the best compression

```
(1)  Function AdapRegressCompress (double eps , int start , int size )
(2)  input: the predefined regression error bound eps denoted by the average error of normalized infinite norm;
      the sampling data of related attributes X[0 ··· M–1], Y[0 ··· M–1], M is the number of buffer data in a sensor node;
      initially start is 0 and size is M.
(3)  output: the compression representation of non-base attribute Y, in the receiving end which can be used to
      reconstruct the raw sampling data satisfied the predefined regression error bound
(4)  begin
(5)     double a, b, old_a, old_b;
(6)     int counterror;
(7)  loop:
(8)     int startpos= start;
(9)     int count= 8;
(10)    while (startpos+ count <= start+ size) do{
(11)       IncRegress (startpos, count, eps, &a, &b, &counterror);
(12)       if (counterror> 0) then{
(13)          if (count==8) then{
(14)             Directly transmit the eight raw data Y[startpos ··· startpos+7] to the receiving sensor node;
(15)             startpos += 8;    }
(16)          else{
(17)             Transmit the 4-tuples (old_a, old_b, startpos, count/2) for regression representation of count/2 data;
(18)             //the approximation of Y[startpos.. startpos+count/2-1] can be reconstructed by the 4-tuples
                  in the receiving end
(19)             startpos += count/2;
(20)             count=8;  }
(21)       }
(22)       else   {
(23)          count *= 2;
(24)          old_a= a;
(25)          old_b= b;  }
(26)    }  //end while
(27)    if (startpos!= M) then {
(28)       Transmit the 4-tuples (old_a, old_b, startpos, count/2) for regression representation of count/2 data;
(29)       start= startpos+ count/2;
(30)       size= M - start;
(31)       if (size) then AdapRegressCompress (eps, start, size);
(32)       //The above statement can also been expressed as: if (size) then goto loop;
(33)    } //end if
(34) end
```

Algorithm 4: The AR-MWCEB algorithm.

performance. In this case, it only needs to transmit 4 data that described the regression process to represent one segment of the raw sampling data of the nonbase attribute. (3) repeat the above process starting from the first unprocessed data to the last one.

Assuming that the number of the regressed data is l and the error bound is satisfied, but the error bound is not satisfied when it is $2 * l$. Comprehensively considering the convenience of calculation, the computation complexity, the storage capacity, and so forth, the regression with $l + 1$ to $2 * l - 1$ data is not been explored calculation. The new segment to be processed directly starts from $l + 1$ after transmitting the regression representation of the l data.

The compression algorithm based on self-adaptive regression is described as in Algorithm 4.

5.2. Properties of the Algorithm

Property 1. After transmitting the part of the data each time, the number of the remaining data shall be a multiple of 8.

Proof. We assume that the size M of the data buffer in a sensor node is some power of 2. This assumption accords with the actual situation of the memory hardware, and it is also convenient for making wavelet transform on the base attribute in Step 2 of Section 4.2. The total number of data of each attribute in a buffer is usually some power of 2 and more than 2^{10}. Set it to $2^3 * 2^x$; namely, it is $8X$, where X is a natural number. Algorithm 3 either directly transmits 8 raw data or transmits 4 tuples to represent 8 (or 8 multiplying by some power of 2) raw data; that is, the number of processed raw data each time is also a multiple of 8, denoted as $8Y$, where Y is a natural number. Therefore, $8(X - Y)$ raw data are left after

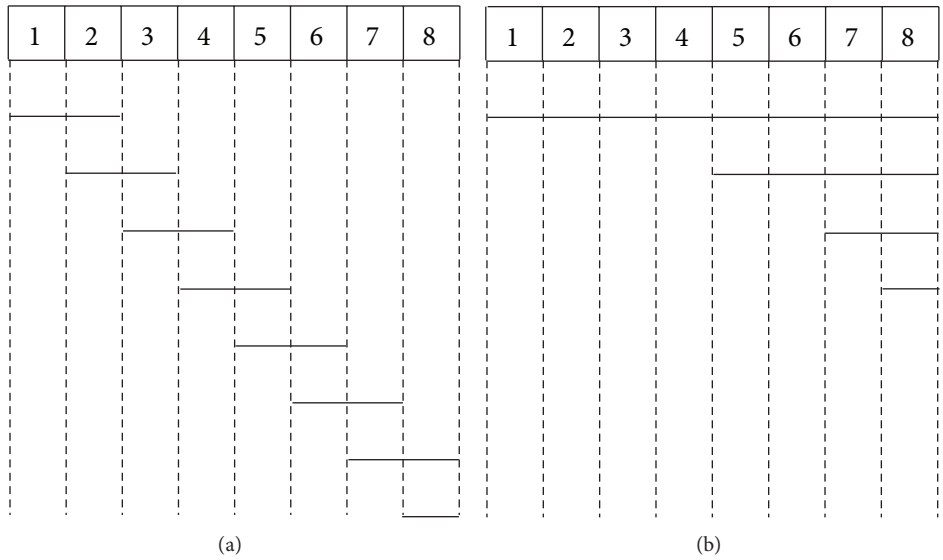


Figure 1: Illustration of data segments involved in regression.

processing a part of the data each time; that is, the number of remaining data in the buffer must be a multiple of 8. □

Property 2. AdapRegressCompress algorithm complying with the aforementioned process is correct.

Proof. When the condition of the while loop in line 10 of AdapRegressCompress algorithm is true, the codes within the loop body are executed. However, only one of three cases can be executed in each loop. The three cases are (1) if the conditions in both line 12 and line 13 are true, the block statement from lines 14 to 15 will be executed; (2) if the condition in line 12 is true but the condition in line 13 is false, the block statement from lines 17 to 20 will be executed; (3) if the condition in line 12 is false, the block statement from lines 23 to 25 will be executed.

The values of variables *start* and *size* are kept unchanged in the process of repeatedly executing the loop body. For each loop, if some data are transmitted, then the increment of variable *startpos* is 8 at least; if no data are transmitted, then the value of variable *count* is doubled. Thus, the while loop must be terminated after finite loops. Next, analyze the states of terminating the while loop by the three cases, respectively.

(1) If the while loop is terminated after execution of line 15. The number of remaining data is a multiple of 8 according to Property 1. After executing statement 15, only 8 data are processed. Now, the loop condition is no longer satisfied; it has shown that only 8 data are left before this loop, and this loop happens to deal with all data. At this time, *startpos* equals to M. The algorithm ends.

(2) If the while loop is terminated after execution of line 20. Denote *startpos*, *count* as *startpos*1, *count*1 and *startpos*2, and *count*2 before and after this loop, respectively. Because $count1 \neq 8$, then *count1* 16. So $startpos1 + count1 = startpos1 + count1/2 + count1/2 = startpos2 + count1/2\ startpos2 + 8 = startpos2 + count2$. As the condition $startpos1 + count1 \leq start + size$ is true when statement 17 is executed, then $startpos2 + count2 \leq start + size$, and this loop must be executed once after executing statement 20; that is, the while loop cannot be terminated by the second case.

(3) If the while loop is terminated after execution of line 25. Lines 27 to 33 deal with the third case. At this time, the first *count/2* data can be represented by regression model, and only 4 tuples need to be transmitted. With the statement 29 and 30, the starting position and the number of the remaining data are calculated. If there are still some remainder data unprocessed, the above process can be repeated by the goto statement in line 32. A more intuitive expression is to recursively invoke this algorithm, namely, the statement 31. □

5.3. Complexity Analysis of the Algorithm. The body of AdapRegressCompress algorithm mainly includes a while loop. In addition to the while loop, statement 11 and statement 31 are the two most time-consuming operations. Similar to the extreme cases of the three terminations of the while loop in previous correctness analysis, the average performance of AdapRegressCompress falls into the range among these extreme cases.

(1) When the linear correlation between the M data and the data of the base attribute is small, 8 raw data must be transmitted each time resulting in no compression. Each raw data is just used one time (i.e., in statement 11) according to *AuxCalc*. The time complexity is the least. The while loop from statements 10 to 26 must be executed $M/8$ cycles before termination, and statements 27 to 33 will not be carried out.

(2) Set $M = 2^{m+3}$ to represent the raw data is divided into 2^m segments, each containing 8 data. As shown in Figure 1(a), when 8 data satisfy the error bound, but 16 data do not in each regression, the compressed data size is half of the raw

one. In Figure 1, 8 segments are given for illustrations, and the horizontal line in each row represents the interval of the raw data involved in regression before transmitting data each time. The first cycle of the while loop uses the first segment with the conclusion that the regression error bound is satisfied. Then the second cycle is followed to be executed after doubling the regression interval to 16 data. Now, the processed 16 data do not satisfy the error bound, and then transmit the representation of the first 8 data with regression coefficients. However, statement 11 has to be executed on 8 data in the second segment again. Except that the 8 data in the first segment are just passed to *AuxCalc* one time for calculation, all the remaining $M - 8$ are required to be used twice by *AuxCalc*, whose time complexity is the largest. The while loop from statements 10 to 26 must be executed $M/4-1$ cycles before termination, and statements 27 to 33 will not be done.

As shown in Figure 1(b), when calculation with the first half part of the processed data in each time satisfies the error bound, but with the whole data does not, its compression performance is better than that of the left subgraph. The number of data used by *AuxCalc* is $8 * 2^m + 8 * 2^{m-1} + \cdots + 8 = 2M - 8$, the same as the case of the left subgraph. The while loop from statements 10 to 26 must be executed $(m+1) + (m) + \cdots + 1 = m(m+1)/2$ cycles to be terminated, and statements 27 to 33 will not be done.

(3) When all the M data are satisfied the regression error bound, the compression performance is best. Each data is simply used once in *AuxCalc*, its time complexity is the smallest. The while loop with statements 10 to 26 must be executed $m + 1 = \log_2 M - 2$ cycles to be terminated, and statements 27 to 33 need to be done once, while the condition in statement 31 is not satisfied.

6. Experiments

The used dataset is provided by Samuel Madden et al. (http://db.csail.mit.edu/labdata/labdata.html), containing more than 2.30 million data collected by 54 Mica2Dot nodes at the same time. Each Mica2Dot node collects four kinds of attribute data: temperature, humidity, light intensity, and voltage, denoted as attributes no. 0, no. 1, no. 2, and no. 3, respectively.

Each real number such as sampling data, wavelet coefficient, regression coefficient, which is stored in Micaz nodes, needs 2 bytes for storage. 2 bytes are enough for storing an integer such as start number *start* and the count number *length* of data. The number of samples in a sensor node buffer is usually no more than $4K$ in order to prevent too long time delay; thus the variable *start* and *length* can be totally stored by 3 bytes. The proposed algorithms are implemented by using VC++ 6.0 on a PC.

Data compression performance can be measured with space savings rate, defined as the reduced data amount by compression to the raw data amount. Suppose that a node's buffer can store M times sampling data, it may send raw data directly or run regression calculation several times for the sake of bounded error. The processed part of data in each linear regression is called a segment. In the following experiments, the normalized error bound for the selected base attribute is set to 0.01, the normalized error bound for the temperature (attribute no. 0) is 0.07; that for the voltage (attribute no. 3) is 0.19.

6.1. Stationary Multivariate Correlation. The correlation degree of the attributes in the experimental dataset can be learned by calculating their correlation coefficient matrix. It can be seen from the experimental results that the multivariate correlation decreases with the increase in the number of sampling data in a buffer. However, if the samples in a buffer are too few, the proposed algorithm cannot take full advantage of the temporal and multivariate correlations. When the data used for the base selection is the same as the being transmitted ones, the multivariate correlation can be recognized as being stationary.

The used dataset consists of the initial $1K$, $2K$, and $4K$ times sampling data, and they are grouped by the proposed base selection algorithm [22]. Two extreme cases are not studied here: the error is too large (the attribute no. 1 is selected as the base; attributes no. 0, no. 2, and no. 3 are represented by some regression coefficients); the multivariate correlation is invalid (all the four attributes no. 0, no. 1, no. 2, and no. 3 are as base; the algorithm degenerates into transmitting directly each attribute independently with no multivariate compression). The remaining two cases are (1) attributes no. 0 and no. 3 take attribute no. 1 as the base signal, while attribute no. 2 is a single base signal. (2) Attribute no. 3 takes attribute no. 0 as the base signal, while both attributes no. 1 and no. 2 are single base signals.

For the above case 1, set $G_1 = \{no\ 0, no\ 3, no\ 1\ (\text{base})\}$, $G_2 = \{no\ 2\ (\text{base})\}$. As attribute no. 2 (light intensity) is a single base signal, it just needs to execute SWCEB algorithm on it, the same as attribute no. 1 (humidity) is. The higher space savings rate can be obtained with the used data as many as possible under the satisfaction of the error bound. Here, the compression performance of the AR-MWCEB algorithm is discussed with different volume of sampling data in a node buffer and both nonbase attributes no. 0 and no. 3 taking attribute no. 1 as the base signal. The experimental results are shown in Figures 2 and 3.

The experimental results have showed that (1) the space savings rate of the AR-MWCEB algorithm is improved with the increase of the normalized error bound. When the normalized error is 0.191, the space savings rate of RACE algorithm is only 87.5% [19]. For the same error bound, the space savings rate of RACE algorithm is always less than that of AR-MWCEB algorithm. (2) To obtain higher space savings rate, PMC-MEAN algorithm [27] should accumulate the processed data in each time as many as possible without exceeding the buffer size of a node. With the number of the samples increasing, multivariate correlation will be weakened so that the absolute error bound increases. However, the AR-MWCEB algorithm determines self-adaptively the number of data involved in the regression by the error bound, and then it can search regression segments in a bigger interval to achieve better compression performance. (3) With respect to the amplitude of attribute no. 3, it has many large high-frequency noises; thus the compression performance is not good for the

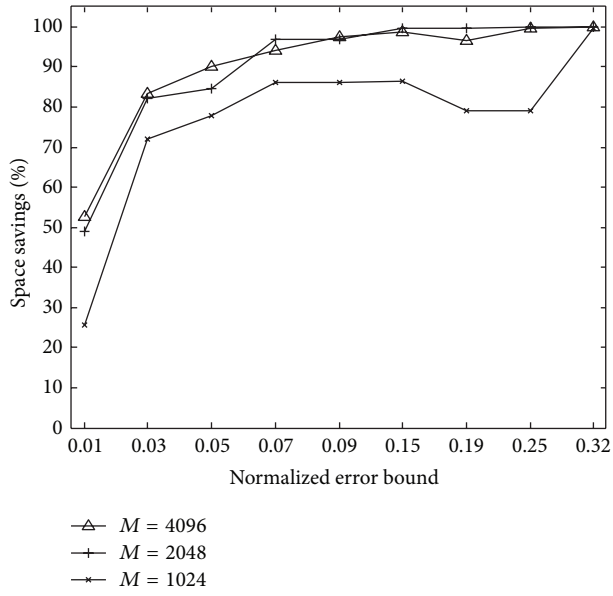


Figure 2: Compressing no. 0 attribute data based on no. 1 using AR-MWCEB.

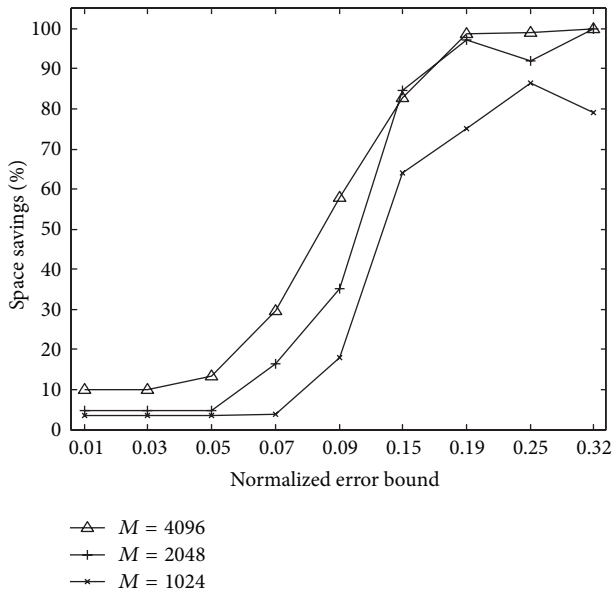


Figure 3: Compressing no. 3 attribute data based on no. 1 using AR-MWCEB.

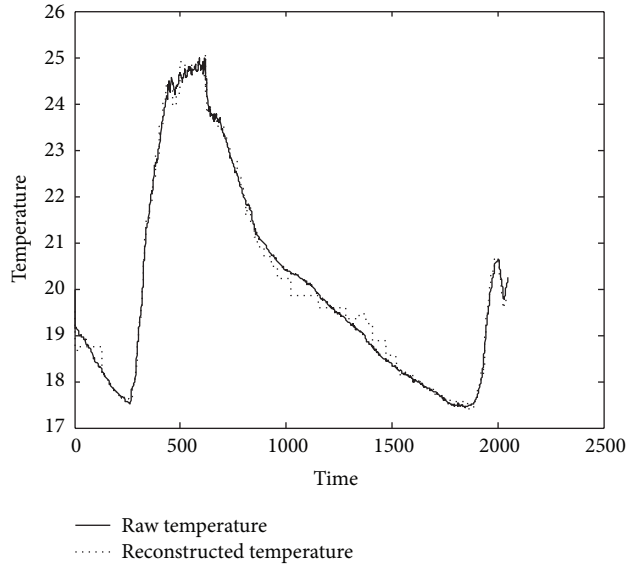


Figure 4: Reconstruction by adaptive regression (no. 0 based on no. 1).

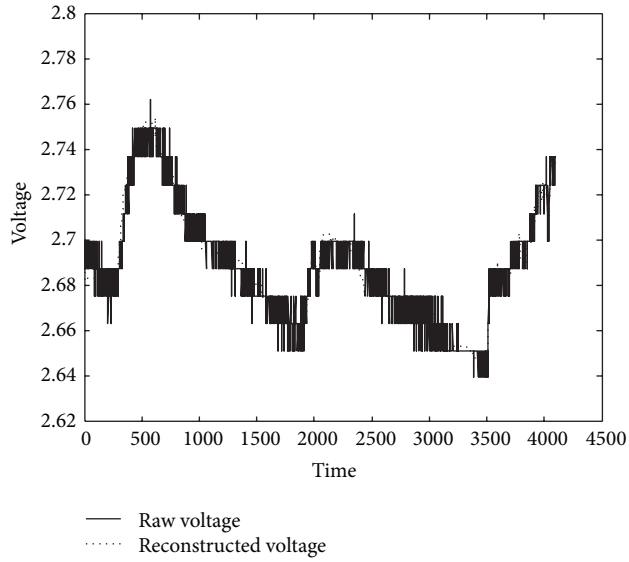


Figure 5: Reconstruction by adaptive regression (no. 3 based on no. 1).

case of small error bound. (4) When the predefined error bound is small, MWCEB may degenerate into the SWCEB without taking advantage of multivariate correlation, and AR-MWCEB solves this problem.

Figure 4 shows the comparison of the reconstructed temperature data with the raw sampling ones, where $M = 2048$. The AR-MWCEB algorithm has self-adaptively divided these experimental data into 13 segments for regression and the 16 data from the 1521th to the 1536th raw data for direct transmission leading to a space savings rate of 96.9971%. Figure 5 shows the comparison of the reconstructed voltage data with the raw sampling ones, where $M = 4096$. The AR-MWCEB algorithm has self-adaptively partitioned these experimental data into 10 segments for regression and the 3841th-3856th 16 data for direct transmission resulting in a space savings rate of 98.7549%.

6.2. Nonstationary Multivariate Correlation. When the correlation coefficient matrix varies with time, each attribute may be reallocated into a group and may act as a new role by using the base selection algorithm in terms of new correlation coefficient matrix every once in a while. In literature [21], before sending data each time, the cluster head had to call

Table 1: AR-MWCEB's performances on compression of no. 0 attribute data.

Size of data buffer (M)	1	2	3	4	5	6	7	8	Average space savings
$1K$	86.1328%	80.4199%	89.3555%	85.7910%	95.8008%	95.7031%	95.4590%	83.3008%	88.9954%
$2K$	96.9971%		91.2598%		96.7285%		90.7471%		93.9331%
$4K$	94.1284%				95.6055%				94.8670%

Table 2: AR-MWCEB's performances on compression of no. 3 attribute data.

Size of data buffer (M)	1	2	3	4	5	6	7	8	Average space savings
$1K$	75.0488%	49.3652%	71.0938%	87.6953%	96.8262%	96.3867%	84.3262%	83.6426%	80.5481%
$2K$	97.1680%		82.2510%		99.3164%		93.5791%		93.0786%
$4K$	98.7549%				98.6084%				98.6817%

preprocessing algorithm to analyze and find the attributes with large correlations, so the overhead was large. This problem also occurred in our previous proposed MWCEB algorithm [22]. But the AR-MWCEB algorithm has been greatly improved by using the base selection algorithm to obtain the information about correlations between attributes. It can avoid considering the change of the correlation coefficient matrix and can regroup the attributes in a long time interval. Although the number of the samples involved in regression calculation is automatically determined to ensure that the error is bounded, the time-varying multivariate correlation may lead to worse compression performance. When the data used for the base selection is not the same as the being transmitted ones, the multivariate correlation can be recognized as being nonstationary.

Equally dividing the first $8K$ times samples of the dataset into 8 segments and calculating the corresponding 8 correlation coefficient matrixes, it is easy to find that these matrices change over time. Next experiments are conducted on transmission of the first $8K$ times sampling data, where the base selection is based on the beginning $1K$ sampling data, and attributes no. 0 and no. 3 are determined to take attribute no. 1 as the base signal by the correlation analysis. The attribute correlation is no longer analyzed in a long time. The compression performances of AR-MWCEB algorithm at different time are analyzed when M is $1K$, $2K$, and $4K$, respectively, and are listed in Tables 1 and 2. The used normalized error bound for the temperature (attribute no. 0) is 0.07; the normalized error bound of the voltage (attribute no. 3) is 0.19.

The experiments have shown that (1) since the attribute correlation is reduced with the increase of M, MWCEB algorithm can only divide the sampling data into equal segments, while AR-MWCEB algorithm can self-adaptively segment the data resulting in a smaller error. (2) The attributes are grouped and located roles by the data firstly filling the node buffer. Although the correlation coefficient matrix may change with time, the compression performance of AR-MWCEB algorithm is slightly impacted by the time-varying multivariate correlation. (3) The compression performance is also affected by the distribution of the sampling data. The amplitudes of attribute no. 0 change greatly, and it has few high-frequency noises. For attribute no. 3, its amplitudes change slightly, but it has many large high-frequency noises. Thus, the default error bound for attribute no. 0 is set to a low value, while for attribute no. 3 it is of a larger value.

7. Conclusions

Aiming at effectively compressing the sampling data from the sensor networks node, between which the spatial correlation is nonexistent or nonstationary, this paper proposed a self-adaptive regression-based multivariate data compression scheme with error bound. The algorithms can run independently on each node. Determined by the predefined error bound and compression income, our algorithms can automatically select to transmit the raw data or the regression coefficients and explore the optimal number of the data involved in each regression. The compression performances of the proposed algorithms are also effective when multivariate correlations are reduced or nonstationary. The proposed algorithm is applicable for processing linear multivariate data by researching on using the linear relationship between base and nonbase attributes to represent the compressed results of nonbase attributes.

It is worthy of further research as to how to use a less complicated method to represent the nonlinearity between the base and nonbase attributes. In addition, for the model-based data collection in WSN, how to construct a model with dynamic evolution over time is also going to be our future work.

Acknowledgments

The work in this paper was partly funded by UK EPSRC Project DANCER (EP/K002643/1), EU FP7 Project MONICA (GA-2011-295222), the Science and Technology Planning Project of Hunan Province of China (Grant no. 2011SK3081), the Scientific Research Fund of Hunan Provincial Education Department (Grant no. 12B003), and the National Natural Science Foundation of China (Grant no. 61202439).

References

[1] R. V. Kulkarni, A. Förster, and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: a survey,"

IEEE Communications Surveys and Tutorials, vol. 13, no. 1, pp. 68–96, 2011.

[2] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home M2M networks: architectures, standards, and QoS improvement," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 44–52, 2011.

[3] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

[4] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC '05)*, pp. 8–13, April 2005.

[5] C. X. Wang, D. Yuan, H. H. Chen, and W. Xu, "An improved deterministic SoS channel simulator for multiple uncorrelated Ravleigh fading channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 9, pp. 3307–3311, 2008.

[6] X. Cheng, C. X. Wang, H. Wang et al., "Cooperative MIMO channel modeling and multi-link spatial correlation properties," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 388–396, 2012.

[7] C. X. Wang, X. Hong, H. H. Chen, and J. Thompson, "On capacity of cognitive radio networks with average interference power constraints," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 1620–1625, 2009.

[8] Q. Ni and C. Zarakovitis, "Nash bargaining game theoretic scheduling for joint channel and power allocation in cognitive radio system," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 70–81, 2012.

[9] S. Bai, W. Y. Zhang, G. L. Xue, J. Tang, and C. G. Wang, "DEAR: delay-bounded energy-constrained adaptive routing in wireless sensor networks," in *Proceedings of the 31st Annual IEEE International Conference on Computer Communications (INFOCOM '12)*, pp. 1593–1601, 2012.

[10] J. Kabara and M. Calle, "MAC protocols used by wireless sensor networks and a general method of performance evaluation," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 834784, 11 pages, 2012.

[11] L. Zhang and Y. Zhang, "Energy-efficient cross-layer protocol of channel-aware geographic-informed forwarding in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 6, pp. 3041–3052, 2009.

[12] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos, "Dissemination of compressed historical information in sensor networks," *VLDB Journal*, vol. 16, no. 4, pp. 439–461, 2007.

[13] B. Kanagal and A. Deshpande, "Online filtering, smoothing and probabilistic modeling of streaming data," in *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE '08)*, pp. 1160–1169, April 2008.

[14] F. Kazemeyni, E. B. Johnsen, O. Owe, and I. Balasingham, "MULE-based wireless sensor networks: probabilistic modeling and quantitative analysis," in *Proceedings of the 10th International Conference on Integrated Formal Methods*, vol. 7321 of *Lecture Notes in Computer Science*, pp. 143–157, 2012.

[15] H. Najafi, F. Lahouti, and M. Shiva, "AR modeling for temporal extension of correlated sensor network data," in *Proceedings of the International Conference on Software, Telecommunications andComputer Networks (SoftCOM '06)*, pp. 117–120, October 2006.

[16] D. Tulone and S. Madden, "PAQ: time series forecasting for approximate query answering in sensor networks," in *Proceedings of the 3rd European Workshop for Wireless Sensor Networks (EWSN '06)*, vol. 3868 of *Lecture Notes in Computer Science*, pp. 21–37, 2006.

[17] Y. L. Borgne and G. Bontempi, "Time series prediction for energy-efficient wireless sensors: applications to environmental monitoring and video games," in *Proceedings of the 4th International ICST Conference on Sensor Systems and Software (S-Cube '12)*, vol. 102, pp. 63–72, Lakshmi Narain College of Technology, 2012.

[18] Y. L. Borgne and G. Bontempi, "Unsupervised and supervised compression with principal component analysis in wireless sensor networks," in *Proceedings of 1st International Workshop on Knowledge Discovery from Sensor Data (SensorKDD '07)*, pp. 55–79, 2007.

[19] H. Chen, J. Li, and P. Mohapatra, "RACE: time series compression with rate adaptivity and error bound for sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 124–133, October 2004.

[20] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, "Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN '06)*, pp. 309–316, April 2006.

[21] T. J. Zhu, Y. P. Lin, S. W. Zhou, and X. L. Xu, "Adaptive multiple-modalities data compression algorithm using wavelet for wireless sensor networks," *Journal on Communications*, vol. 30, no. 3, pp. 48–53, 2009 (Chinese).

[22] J. M. Zhang, Y. P. Lin, S. W. Zhou, and J. C. Ouyang, "Haar wavelet data compression algorithm with error bound for wireless sensor networks," *Journal of Software*, vol. 21, no. 6, pp. 1364–1377, 2010.

[23] X. Song, C. R. Wang, J. Gao, and X. Hu, "DLRDG: distributed linear regression-based hierarchical data gathering framework in wireless sensor network," *Neural Computing and Applications*, 15 pages, 2012.

[24] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 265–278, November 2006.

[25] M. Garofalakis and P. B. Gibbons, "Wavelet synopses with error guarantees," in *Proceedings of the ACM SIGMOD International Conference on Managment of Data (SIGMOD '02)*, pp. 476–487, June 2002.

[26] M. Garofalakis and A. Kumar, "Deterministic wavelet thresholding for maximum-error metrics," in *Proceedings of the 23rd ACM SIGMOD—SIGACT—SIGART Symposium on Principles of Database Systems (PODS '04)*, pp. 166–176, June 2004.

[27] I. Lazaridis and S. Mehrotra, "Capturing sensor-generated time series with quality guarantees," in *Proceedings of the 19th International Conference on Data Ingineering*, pp. 429–440, March 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 429065, 13 pages
http://dx.doi.org/10.1155/2013/429065

Research Article

Efficient Deterministic Anchor Deployment for Sensor Network Positioning

Yongle Chen,[1,2] Ci Chen,[3] Hongsong Zhu,[1] and Limin Sun[1]

[1] *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*
[2] *University of the Chinese Academy of Sciences, Beijing 100190, China*
[3] *The School of Software and Microelectronics, Peking University, Beijing 102600, China*

Correspondence should be addressed to Limin Sun; sunlimin@iie.ac.cn

Received 10 January 2013; Accepted 24 February 2013

Academic Editor: Jianwei Niu

Sensor network positioning systems have been extensively studied in recent years. Most of the systems share a common assumption that some known-position anchor nodes have existed. However, a more fundamental question is always being overlooked, that is, how to acquire the anchor's position. In general, GPS-based measures and the artificial calibration are two dominant methods to acquire anchor positions. Due to the high energy cost and failures in occlusion regions of the GPS modules, the artificial calibration method is adopted extensively. Nevertheless, numerous disadvantages of the artificial calibration, such as the expensive labor cost and error-prone features, also make it hard to be an efficient solution for the anchor positioning. For this reason, we design an efficient mapping algorithm between anchors and their positions (MD-SKM) to avoid the complicated artificial calibration. Additionally, we propose a best feature matching (BFM) method to further relax the restriction of MDS-KM where three or more calibrated anchors are needed. We evaluate our MDS-KM algorithm under various topologies and connectivity settings. Experiment results show that at a slightly higher connectivity level, our algorithm can achieve the exactly correct matching between anchors and their positions without any calibrated anchors.

1. Introduction

Sensor networks have been extensively studied due to their salient advantages of monitoring and controlling related applications, such as the battlefield monitoring, medical surveillance, and structure monitoring. Furthermore, in these applications, a basic service requirement implied is to determine the exact location of the happening event through pre-deployed sensors, in order that the operators are prone to execute appropriate control actions in response to the event. In this sense, the positioning technology of the sensor network is getting more and more attentions. Currently, most developed positioning systems depend on four metrics, including TOA, AOA, RSS, and the connectivity of the signals. In addition, these systems also have a common assumption; that is, some known-position nodes exist in the sensor network, which are also named as anchors. Based on the known-position anchors, they are able to localize other unknown-position nodes. However, how to acquire the anchor's position is still an unsolved question.

At present, two methods are mainly used for acquiring anchors' positions. One is leveraging GPS modules, and the other is called the artificial calibration [1]. In practice, the placement ways to construct the sensor network decide which method to be adopted to acquire anchors' positions. The placement ways can be summarized in two types: the stochastic way and the deterministic way. In terms of the stochastic way, the most typical example is the battlefield monitoring. In the battlefield scenario, military sensors are randomly scattered from the air, where the anchors' positions are stochastic. In this case, to acquire anchors' positions, we have to rely on the GPS modules attached with sensors. The GPS modules are usually limited by a series of disadvantages, such as high energy costs and failures in occlusion regions, which make this method not applicable for the low-power sensor networks. Different from the stochastic way, in some

scenarios, the anchors' positions are deterministic according to predesigned placement blueprint, such as that in medical surveillance and structure monitoring. In these scenarios, the correspondence is recorded between anchors' physical positions in the blueprint and anchors' IDs, such that each anchor unambiguously knows its own position. We named this method the artificial calibration. Also, many works [2, 3] have pointed out that optimizing the anchor placement is able to accelerate the convergence of the positioning algorithm and improve the positioning accuracy. Nevertheless, this kind of methods always suffers from the complicated and error-prone mapping between physical locations and the node IDs, which is even more severe in a large sensor network.

In this paper, we design an efficient MDS-KM matching algorithm to avoid the artificial calibration cost in deterministic anchor placement. To the best of our knowledge, we are the first to consider solving the artificial calibration problem of anchors placement. Given sufficient calibrated anchors (C-anchors for short), we first design a distributed MDS-MAP(A) method to construct an absolute radiomap by using estimated distance or hop distance between anchors. In the absolute radiomap, each anchor has an absolute coordinate of its position, corresponding to the physical position in the blueprint. In order to map the radiomap with the blueprint, we use the kNN method to select the k-nearest physical positions in the blueprint away from the anchor absolute positions in the radiomap and then build a complete bipartite graph. Based on the bipartite graph, we adopt the Kuhn-Munkres (KM) algorithm to get a maximum weighted matching. Accordingly, we achieve the correspondence between anchor nodes in the radiomap and physical positions in the blueprint. Meanwhile, in order to relax our MDS-KM method for the cases without calibrated anchors, we design a best feature matching (BFM) method to actively map parts of anchors in the radiomap to positions in the blueprint. Our method will greatly improve the efficiency of anchor placement through avoiding the artificial calibration. The experiment in Section 6 shows that the mapping from the radiomap to the blueprint is exactly correct when the connectivity level of network is not excessively low.

The remainder of the paper is organized as follows. The related work is shown in Section 2. We formulate the problem in Section 3 while leaving the details of our algorithm design for Section 4. The further improved strategy is presented in Section 5. Then we show the experiment and simulation results of our MDS-KM algorithm in Section 6. Finally, we conclude the paper in Section 7.

2. Related Work

Many works have pointed out that anchor placement ways will help to improve positioning performance. The pioneer anchor placement ways are mainly based on the empirical evidence in positioning system. For example, Shang et al. [2] randomly place anchors in their experiment and find that a selection of collinear anchors in one test is rather unlucky. Recently, Akl et al. [3] study the anchor placement for passive positioning, and they find that the optimal placement is that no three anchor nodes are collinear at the center of network. The authors of [4] point out that the optimal placement of anchors should be around the corners of the network and also find that the more nonlinearity results in the better positioning performance.

Doherty et al. [5] place the anchors at the corners of the network to acquire a better positioning results. However, the algorithm has a constraint requirements that all the unknown nodes should be placed within the convex hull of the anchors, which reduces the algorithm generality. Ash and Moses [6] analyse and prove that the anchors on the corners of network will help to improve positioning result when the network is a rectangle. Hara and Fukumura [7] also propose an anchor placement algorithm applied to the rectangle network, and that they point out the anchors must be placed in the centers of subrectangle regions divided from the rectangle network.

Some anchor placements focus on the effect of the environment. For example, the authors of [8] conduct some experiments where anchors are placed either on the ceiling or the floor. The study find that anchors or on the floor are better for monitoring moving people in the room. Although many anchor placement works are developed, they only focus on how to improve the positioning performance based on anchor positions and ignore how to acquire the positions of anchors. This paper analyses the artificial calibration problem to acquire the anchor positions after deterministic placement. In order to efficiently acquire the anchor positions, we introduce MDS method to construct a radiomap corresponding to the blueprint. Then anchors physical positions can be self-calibrated by mapping the radiomap to the blueprint.

MDS method is a series of analysis techniques used for displaying the data proximity as a geometrical picture [9]. At present, there are many variants of MDS positioning algorithm, including classical metric MDS-MAP(C), distributed MDS-MAP(P), local MDS, and weighted dwMDS(G). Centralized MDS-MAP(C) [10] algorithm is the earliest usage of MDS techniques in sensor network positioning. Since MDS-MAP(C) uses the shortest hop distance as the estimate of the true Euclidean distance, it is not good for irregular network. A distributed MDS method, MDS-MAP(P) [11], is proposed to be applied to different network topologies. MDS-MAP(P) first constructs a 2-hop local map by executing MDS-MAP(C) method for nearby nodes then merges each local map into a global map based on the common nodes. Local MDS [12] is another distributed variant of MDS-MAP(C) improved for irregular topologies. The difference from MDS-MAP(P) is that the nearby nodes of constructing local map only include 1-hop neighbors and the weights are restricted to 0 or 1. Meanwhile, a least square optimization method is used for refining the local maps. The dwMDS(G) [13] is a weighted distributed MDS method, in which a weighted (Gauss kernel) cost function is adopted for adaptively emphasizing the most accurate range measurements. Besides, dwMDS(G) designs a neighbor selection method to avoid the biasing effects of noisy range measurements neighbors.

In this paper, we design a distributed MDS-KM method to increase the efficiency of anchor placement. At first, we design an MDS-MAP(A) method focusing on the anchor

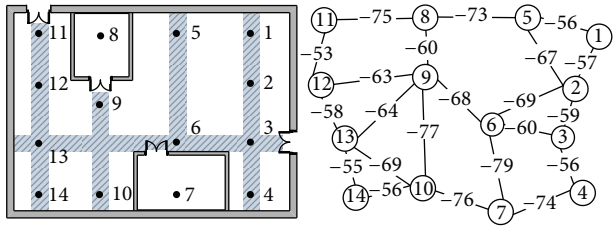


Figure 1: The blueprint and the corresponding radiomap.

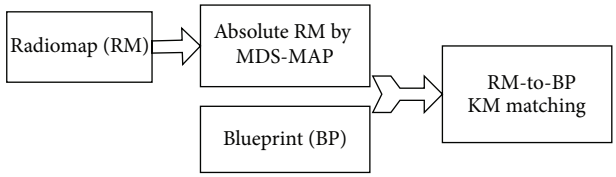


Figure 2: The MDS-KM algorithm framework.

positioning to construct a radiomap with absolute coordinates, which is not subject to the irregular anchors distribution. Afterwards, we use the KM algorithm to obtain the maximal weighted matching of complete bipartite graph constructed by the radiomap and the blueprint. Besides, our MDS-KM method can also avoid the error-prone mapping during the artificial calibration.

3. Problem Specification

The optimized placement of anchors has a very important impact for the positioning performance. For improving the positioning accuracy, a predefined blueprint is usually constructed to guide the anchor placement before deploying the positioning system. That is called deterministic anchor placement. For example, the left graph in Figure 1 is a blueprint, where the black cycles are the positions to place anchors. During the placement, anchor node ID will be one-to-one mapped to the anchor position marked on blueprint, which is called artificial calibration. This process will consume a higher labor cost and lead to error-prone mapping. In order to solve this problem, we build a radiomap using the connectivity in large sparse network or signal strength between anchors in small dense network and then adaptively map the radiomap to the blueprint with little or no artificial calibration. As shown in the right graph of Figure 1, the vertices in the radiomap represent anchor nodes and the edge weights represent the signal strength in the small network. In sparse network, many anchor nodes may have only few neighbor anchors or even none. Here, we make the shortest hop distance from one anchor to another anchor as the weight in the radiomap.

Accordingly, the problem to be solved becomes the exact mapping from the radiomap to the blueprint. Intuitively, the radiomap has similar characteristics with the blueprint. The radiomap-to-blueprint mapping should be graph isomorphism (GI) problem [14]. But in the small network, it does not

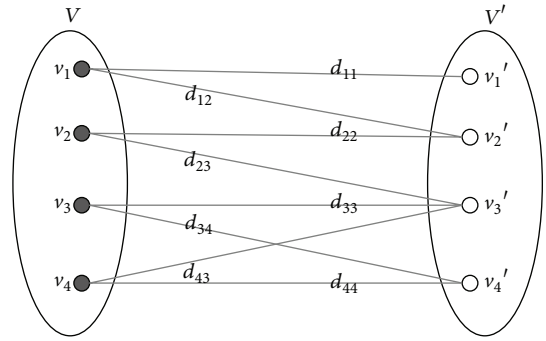


Figure 3: A complete bipartite graph.

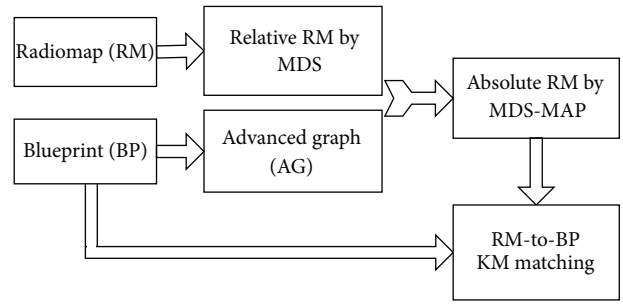


Figure 4: The improved MDS-KM algorithm framework.

strictly belong to graph isomorphism problem. Supposing the radiomap and the blueprint are isomorphic, each vertex and edge in both graphs must have a corresponding bijection. As the physical distance increases in the blueprint, the RSS in the radiomap damps and even disappears, but the physical distance can still be measured. Thus the blueprint is a complete graph, while the radiomap is a subgraph of the blueprint. Even though we limited the maximum measure distance in the blueprint, the edges in the blueprint may still not have a corresponding bijection to the edges in the radiomap due to the effect of the surrounding noise. The edges in the radiomap only have a corresponding bijection with the subset of the blueprint. This is a typical subgraph isomorphism problem [15].

However, subgraph isomorphism is an NP-complete problem [16]. Furthermore, the distances between vertices in the blueprint do not exactly reflect the RSS values in the radiomap subjected to the surrounding noise. Therefore, the existing heuristic subgraph isomorphism algorithm is not suitable for the radiomap-to-blueprint mapping. In this paper, we design an MDS-KM matching algorithm to solve this mapping problem in the small network or the sparse network. We introduce the multidimensional scaling (MDS) method in the anchor placement, which is well suited to compute a relative coordinates map in a low-dimensional space by one matrix representing distance information between nodes. Based on MDS method and sufficient known-position calibrated anchors (3 or more), we design a distributed MDS-MAP(A) method to construct the radiomap with absolute coordinates. Then the Euclidean distances of vertices in the

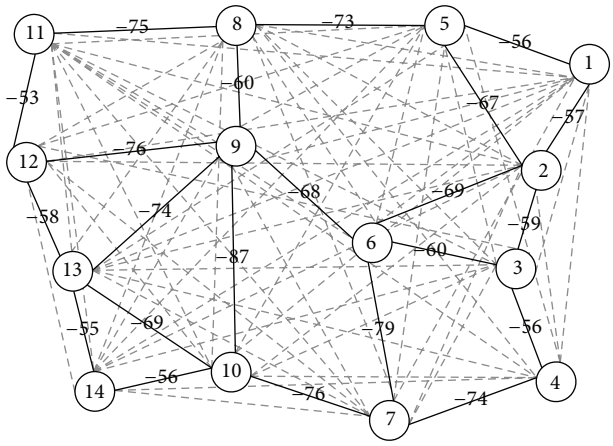


Figure 5: The advanced graph in the small network.

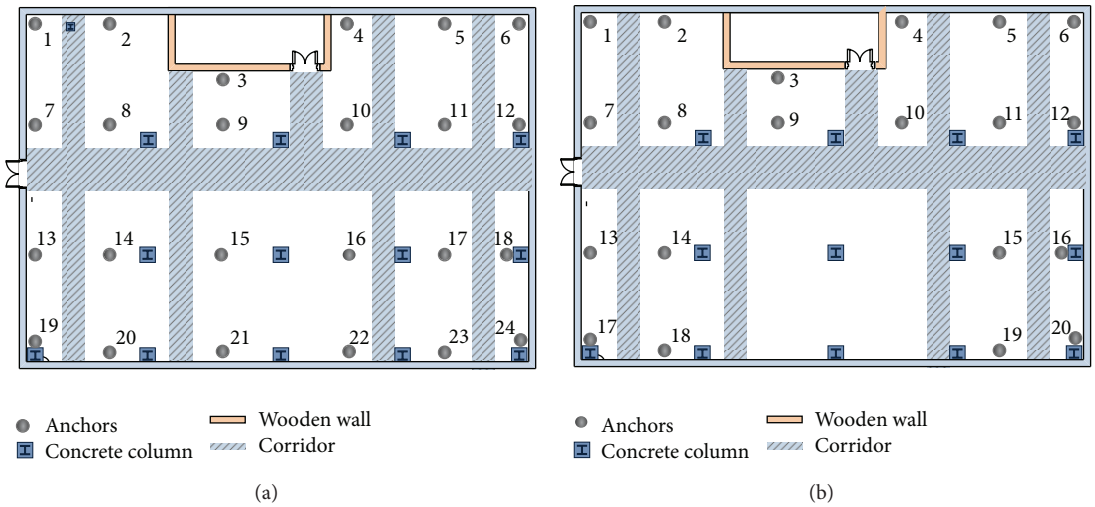


Figure 6: Two experiments in uniform and n-sharp distribution.

radiomap and the blueprint are computed as the weights to construct a weighted bipartite graph, where one part of the bipartite graph includes all the vertices in the radiomap, and the other part of the bipartite graph includes all the vertices in the blueprint. Afterwards, we adopt the classical Kuhn-Munkres (KM) method [17] to carry out a maximum weight matching of the bipartite and then get a one-to-one mapping between anchor node IDs in the radiomap and positions in the blueprint.

4. Radiomap-to-Blueprint Mapping

4.1. Algorithm Overview. As mentioned above, the matching between the radiomap and the blueprint is our primary objective. The MDS-KM matching process is illustrated in Figure 2. In general, the MDS method utilizes the physical distance between anchors to construct a relative coordinate radiomap. But the edge weights in the radiomap of the small network represent the RSS values. We need to transform RSS value to the estimated distance according to the signal propagation model. Then we use the MDS method to get a radiomap with relative coordinates. Having sufficient anchor node positions (3 for 2D networks and 4 for 3D networks), we can map the relative coordinates of anchors to absolute coordinates through a linear transformation [10]. Then we use KM algorithm to compute the optimal complete matching between the blueprint and the radiomap with absolute coordinates. Since the KM algorithm is applied to the weighted bipartite graph matching, we need to construct a bipartite graph utilizing the radiomap and the blueprint. Thus we design an error-torrent kNN vertex selection method to build a bipartite graph. Finally, we achieve the mapping from the radiomap to the blueprint through computing the maximum weighted matching of the bipartite. In Section 5,

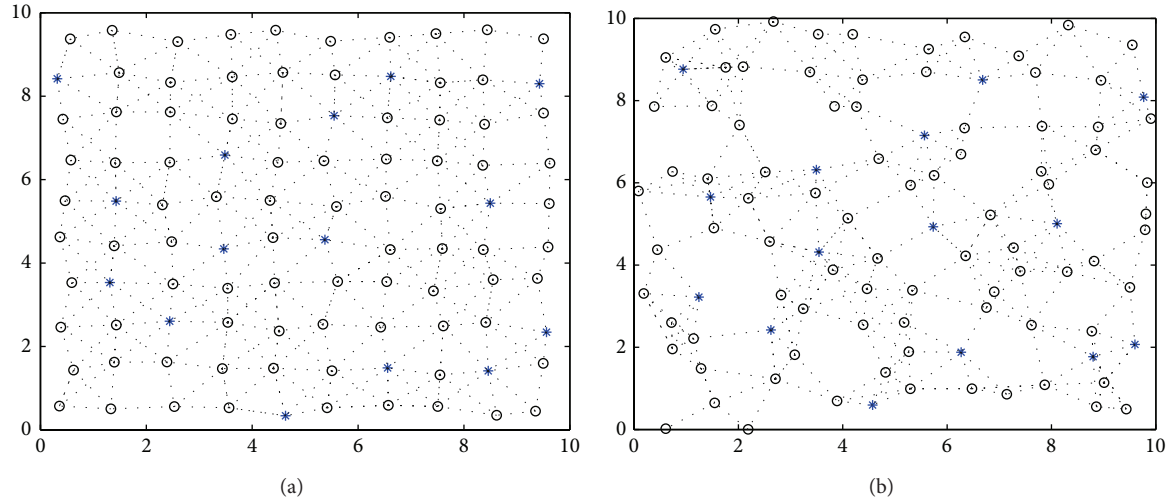


Figure 7: Two simulations in grid and random distribution.

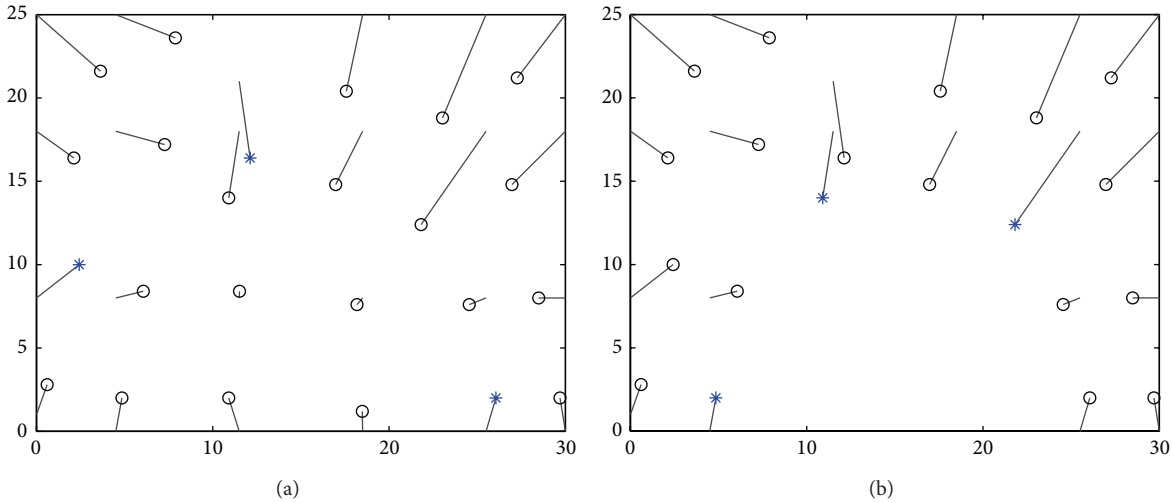


Figure 8: The average errors in both experiment scenarios.

we further design a best feature matching (BFM) method to relax the restriction of MDS-KM where three or more calibrated anchors are needed.

4.2. Absolute Radiomap Construction

4.2.1. Collecting Distance Information. In order to construct a radiomap, we need to compute the estimated Euclidean distance based on the RSS or hop distance between anchors. The RSS or hop distance of each pair of anchors should be obtained at first. Intuitively, flooding is a better selection. In the small dense network, each anchor node broadcasts the beacon packet periodically and keeps on receiving the beacon packets from other anchors then computes the RSSs of these beacon packets. After a while, each anchor will record an RSS sequence from other anchor nodes within its 1-hop communication range. Finally, each anchor sends its node ID and RSS sequences to the backend positioning server for constructing the radiomap. In order to avoid the sending collision, we will make the broadcast cycle of each anchor different in our experiment.

Additionally, in large sparse network, many anchors may not be within the communication range of any other anchors. These anchors are isolated. We will use the shortest hop distances as the estimated distance. There are some intermediate unknown-position nodes scattered within the anchors. The shortest hop distance is defined as the minimum hop count between anchors multiplied by the average signal hop distance. In this process, each anchor will broadcast its beacon packet periodically. Each intermediate unknown-position node records the minimal hop value and adds itself to the value and then forwards the hop count continually with initial anchor ID until the beacon packet arrives to a new anchor or achieves our hop limit. In order to reduce the

communication cost, we set a hop upper limit (e.g., 10) to construct local map. Each anchor records all the minimum hop counts from nearby anchors and sends them and their node IDs to the positioning server.

4.2.2. Estimated Distance. In the large sparse network, we can compute the Euclidean distance between the calibrated anchors. According, to the minimum hop counts between them, we further compute the average single hop distance. Accordingly, we can compute the hop distance between each pairwise anchors as the estimated distance. In the small dense network, we need to use signal propagation model to compute the estimated distance based on the RSS value. According to whether the travel distance is short or large, the propagation models can be classified into large scale and small scale [18]. In general, the small-scale model needs to characterize the rapid fluctuations of RSS over short travel distance. It has a better accuracy than large-scale model, but it is very difficult to determine the model parameters. In this paper, we concentrate on the generality of the designed algorithm and do not consider a specific scenario. Hence, we select a good compromise between simplicity and accuracy, which is called the wall attenuation factor propagation model (WAF) [19]. This model provides flexibility when applied to indoor scenario while considering outdoor large-scale fading. This model is described as

$$p(d)\,[dbm] = p(d_0)\,[dbm] - 10\alpha \log\left(\frac{d}{d_0}\right) - \delta$$
$$\delta = \begin{cases} nw \times \mathrm{WAF}, & nw < C \\ C \times \mathrm{WAF}, & nw \geq C, \end{cases} \tag{1}$$

where d is the transmitter-receiver distance, $P(d_0)$ is the signal power at some reference distance d_0, α indicates the rate at which the signal fades, C is the maximum number of obstacles up to which the attenuation factor makes a difference, nw is the number of obstacles between the transmitter and the receiver, and WAF is the obstacle attenuation factor. In general, the values of α and WAF depend on the specific propagation environment and should be derived empirically. Given the RSS value, we can further compute the estimated distance d_e as follows:

$$d_e = d_0 \times 10^{(p(d_0)[dbm] - p(d)[dbm] - \delta)/(10\alpha)}. \tag{2}$$

Additionally, there are some optimization methods to tune parameters of propagation model so that the RSS measurements can characterize the accurate distances [20–22]. In our algorithm, the MDS method can tolerate error gracefully due to the overdetermined nature of the solution [9]. Hence we do not need exactly RSS values depending on optimizing the propagation model.

4.2.3. Constructing Absolute Radiomap. In this part, we will use the MDS method to construct the absolute radiomap. At present, many types of MDS techniques have been developed [9]. In our algorithm, we design a distributed MDS-MAP(A) algorithm focusing on the anchor placement. The MDS-MAP(A) algorithm consists of four main steps as follows.

First, we use the above estimated distance to construct the 1-hop proximity matrix P for each anchor, where the 1-hop neighbors of anchors in large network will be the anchors in the range of hop upper limit. We denote the proximity measure between anchor i and j as p_{ij}. Then assuming an m-dimensional space, given the anchor i coordinates $X_i = (x_{i1}, x_{i2}, \ldots, x_{im})$ and the anchor j coordinates $X_j = (x_{j1}, x_{j2}, \ldots, x_{jm})$, the practical Euclidean distance between anchor i and j is denoted by d_{ij} which will construct a Euclidean distances matrix D as

$$d_{ij} = \sqrt{\sum_{k=1}^{m} \left(x_{ik} - x_{jk}\right)^2}. \tag{3}$$

In theory, the matrix P should be equal to the matrix D. But the estimated distance with errors makes them unequal. In this case, the MDS method can ensure P is approximate to D as far as possible.

Second, we run the MDS algorithm for each distance matrix P to get a local map with relative coordinates. In classical metric MDS, the proximity matrix P can be transformed to a double centered matrix B, which is symmetric and positive semidefinite matrix as

$$B = -\frac{1}{2}\left(p_{ij}^2 - \frac{1}{n}\sum_{j=1}^{n} p_{ij}^2 - \frac{1}{n}\sum_{i=1}^{n} p_{ij}^2 + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}^2\right). \tag{4}$$

When we shift P to the center, B can also be expressed as follows:

$$B = XX^T = \sum_{k=1}^{m} x_{ik} x_{jk}. \tag{5}$$

We perform the singular value decomposition (SVD) on B to get $B = VAV^T$, which has complexity of $O(k^3)$, where k is the number of anchors in the local map. Thus, the complexity of computing n local maps is $O(k^3 n)$, where n is the number of anchors in the radiomap. The coordinate matrix is $X = VA^{(1/2)}$, where $A = \mathrm{diag}(l_1, l_2, \ldots, l_n)$ is the eigenvalue diagonal matrix in descending order. $V = [V_1, V_2, \ldots, V_n]$ is the eigenvector corresponding to the eigenvalue. We select the first m eigenvectors to construct a coordinate matrix in lower dimension. This is the best low-rank approximation between matrix P and D in the least-squares sense.

Third, we merge all local maps to the whole relative radiomap. Each local map is a group of 1-hop neighbors. We randomly select a local map as the base map and then sequentially merge the neighbor local map according to the common nodes. Eventually, the base map grows to cover the whole radiomap. As known from [11], the complexity of this step is the same as step 2.

Finally, given sufficient calibrated anchors, we map the relative coordinates to the absolute coordinates of anchors through a liner transformation [10], which include scaling, reflection, and rotation. The radiomap with absolute positions can be achieved eventually. For r anchors, the complexity of this step is $O(r^3 + n)$.

4.3. Radiomap-to-Blueprint Matching. Since the surrounding noise and irregular topology affect the precision of estimated distance and lead to the inaccuracy absolute coordinates of anchors in the radiomap, the absolute coordinates in the radiomap are not completely consistent with the coordinates of anchor physical positions in the blueprint. Hence, the above two groups of coordinates cannot be corresponding completely. We only search for the most approximate matching of two coordinates. Therefore, the objective of the radiomap-to-blueprint matching turns into minimizing the sum of corresponding Euclidean distances between the physical positions in the blueprint and the absolute coordinate positions in the radiomap. We present a k-nearest neighbor (kNN) method to find the best approximate positions in two graphs. The k-nearest neighbor is a simple classification method in the data mining field. This algorithm can select the k-nearest ones through evaluating Euclidean distance between positions. For each anchor in the radiomap, we utilize the kNN method to find the k-nearest positions in the blueprint away from it. Then we can build a weighted bipartite graph, whose weights on edges are the Euclidean distances. An example with $k = 2$ is shown in Figure 3. Additionally, the value of parameter k is task specific. In our algorithm, we select the minimal k to guarantee that all the positions in the blueprint will be selected into V' when all anchors V in the radiomap have been carried out in the kNN operation. Thus the bipartite graph has a complete matching, where every vertex of the graph is exactly incident to only one edge.

Accordingly, the radiomap-to-blueprint matching problem will be transformed into a minimum weighted matching problem in a weighted bipartite graph, where the sum of the weight of all the edges in the bipartite matching is minimal. Such a matching is also known as the optimal assignment problem. It can be solved by Kuhn-Munkres (KM) algorithm in polynomial time. However, the KM algorithm just applies to solving the maximum weighted matching problem. We need to pick the minus of the weights in the bipartite so that the minimum weighted matching problem is further transformed into a maximum weighted matching problem. The KM algorithm will use vertex labeling method to transform the maximum weighted matching into complete matching in unweighted bipartite graph and then use the classical Hungarian algorithm to solve the maximum matching problem of unweighted bipartite graph.

Algorithm 1 is a simplified KM algorithm procedure. We first initialize a feasible vertex labeling. Normally, each vertex in one side of the bipartite graph is labeled with the maximum weight of its incident edges connected to the vertices in the other side, and each vertex in the other side is labeled zero (line 2–6). The bipartite graph will become an unweighted bipartite graph. Then we seek a maximum matching using Hungarian algorithm and decide whether the maximum matching is a complete matching or not (line 7-8). If the maximum matching is a complete matching, we save the matching and return. Otherwise, we need to relabel the vertices following the KM algorithm rules and literately carry out the Hungarian algorithm (line 12-13). Finally, we can achieve a complete matching and get the mapping relationships between the radiomap and the blueprint.

5. Without Calibrated Anchors

In this section, we try to relax our MDS-KM algorithm to be applied to the situation without any artificial calibrations. We design a best feature matching (BFM) method to actively get parts of mapping from anchors in the radiomap to positions in the blueprint without any artificial calibration. In order to distinguish the feature of vertices in the radiomap and the blueprint, we bring in the vertex weighted sequence as the feature metric, where the edge weight is RSS value or hop count. Then some vertices with best unique feature in the radiomap can be selected and their corresponding vertices are found in the blueprint by our BFM method. However, the edge weight in the blueprint is physical distance. The vertex weighted sequences in the radiomap are not comparable to those in the blueprint because of the different types of the edge weight. Hence, we transform the blueprint to an advanced graph (AG), whose vertex features are the RSS sequences in the small network and hop count sequences in the large network. The new matching process of MDS-KM algorithm is also changed to Figure 4. The advanced graph is used to seek the parts of anchors with a unique feature instead of the calibrated anchors to construct the absolute radiomap.

5.1. Blueprint to the Advanced Graph. In the small network, the distances between vertices in the blueprint are not exactly reflecting the RSSs in the radiomap due to the surrounding obstacles and noise. We first use the signal propagation model mentioned in the above subsection to transform the distances between vertices in the blueprint into the RSS values, which is constructed in an advanced graph denoted by $G_A = (V_A, E_A)$. These RSS values represent the weights of the edges in the advanced graph, and the number of vertices and edges in the advanced graph is the same as that of the blueprint. Since any two vertices in the blueprint have one edge, the advanced graph is also a complete graph. Figure 5 is an example of the advanced graph from the blueprint in Figure 1. In the large network, we compute the minimal hop counts between pairwise anchors in the blueprint after setting the communication range of node and then construct an advanced graph whose edge weights represent minimal hop counts. Similarly, the advanced graph in the large network is also a complete graph.

5.2. Best Feature Matching. Before executing the MDS-MAP(A) method, the radiomap $G_R = (V_R, E_R)$ has the vertex set V_R and edge set V_R. The edge weight represents the RSS or hop count. We first make the vertices distinguishable depending on their invariants, which are the fixed properties of vertices during matching. A simple invariant is the vertex degree. However, in a graph, the vertex degree is not unique. There is likely to be many vertices having the same degree. Therefore, we bring the weights into the vertex invariants, for example, $I(v_i, W) = (v_i, w_1, w_2, \ldots, w_d)$ by following the arrangement $w_1 > w_2 > \cdots > w_d$, and d is the degree of the vertex v_i. Similarly, we can formulate the corresponding vertex invariants of the advanced graph. For

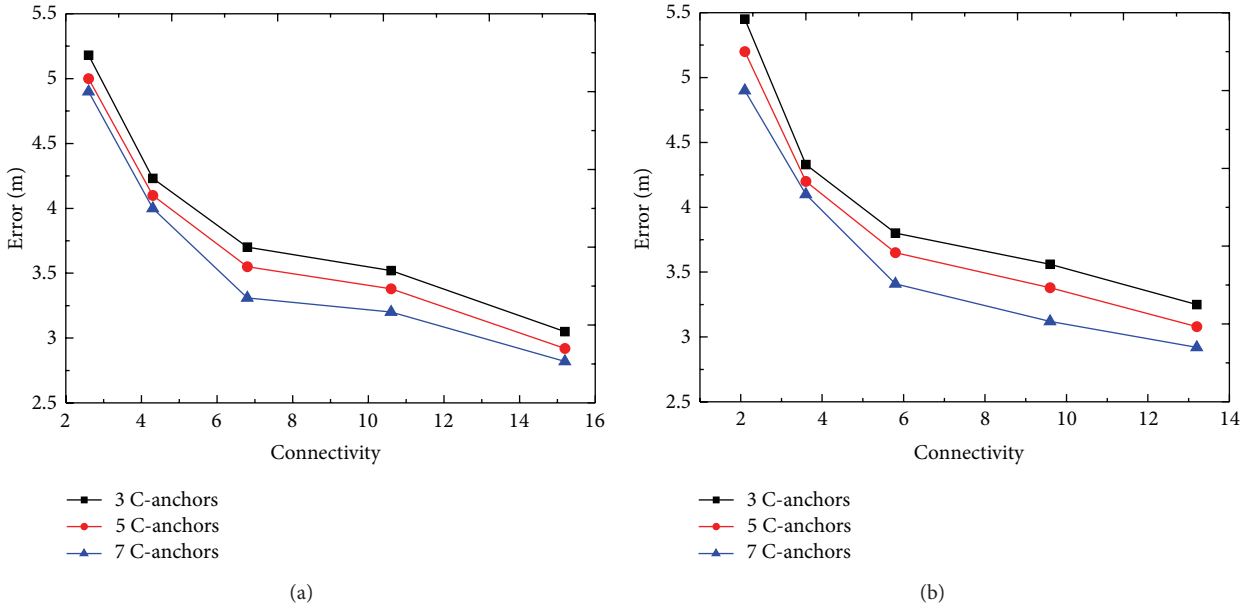


Figure 9: The error analysis in both experiment scenarios.

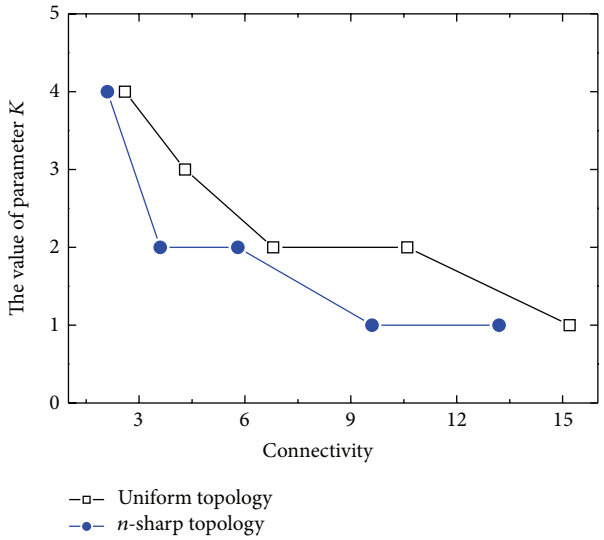


Figure 10: The selection of the K using for constructing bipartite graph.

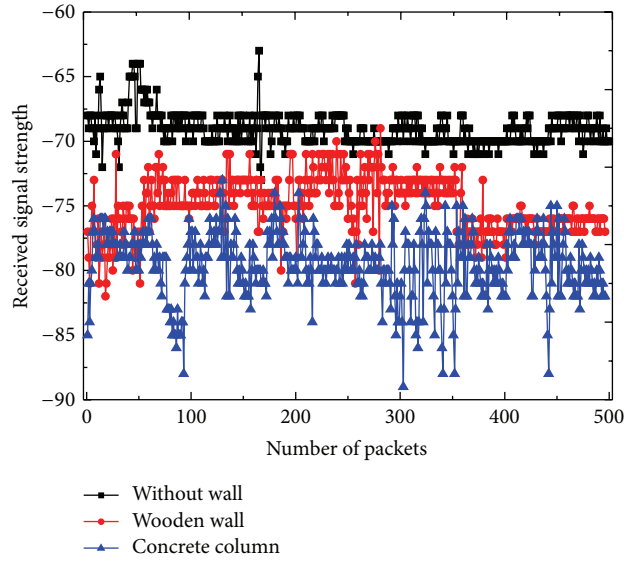


Figure 11: The obstacle affecting.

example, $I'(v_i', w') = (v_i', w_1', w_2', \ldots, w_{(n-1)}'), w_1' > w_2' > \cdots > w_{(n-1)}'$. n is the number of all vertices. Each vertex degree is $n-1$ since the advanced graph is a complete graph.

We will select the vertices invariants in the radiomap which are the most easy to distinguish. We noted that the degrees of many vertices in the radiomap are different so that the number of weights in some vertex invariants is inconsistent. This brings inconvenience to our feature comparison. Therefore, we need to normalize the vertex invariants of the radiomap. We first compute the maximal degree of all vertices Max(d) in the radiomap then extend the vertex invariant $I(v_i, W)$ from $(v_i, w_1, \ldots, w_d)$ to $(v_i, w_1, \ldots, w_d, w_{(d+1)}, \ldots, w_{\mathrm{Max}(d)})$, where $d \leq \mathrm{Max}(d), w_{(d+1)} = w_{(d+2)}, \ldots, = w_{\mathrm{Max}(d)} = w_{\min}$. $w_{\min}$ is the minimum RSS value measured from anchor device in the small network or hop count of zero in the larger network. We can compute the Euclidean distance d_{RA} between vertices invariants in two graphs as follows:

$$d_{RA} = \sqrt{\sum_{i=1}^{\mathrm{Max}(d)} (w_i - w_i')^2}. \quad (6)$$

We still adopt the k-nearest neighbor ($k = 2$) method to find the two minimum d_{RA} between vertices in the radiomap and vertices in the advanced graph. For each vertex in the

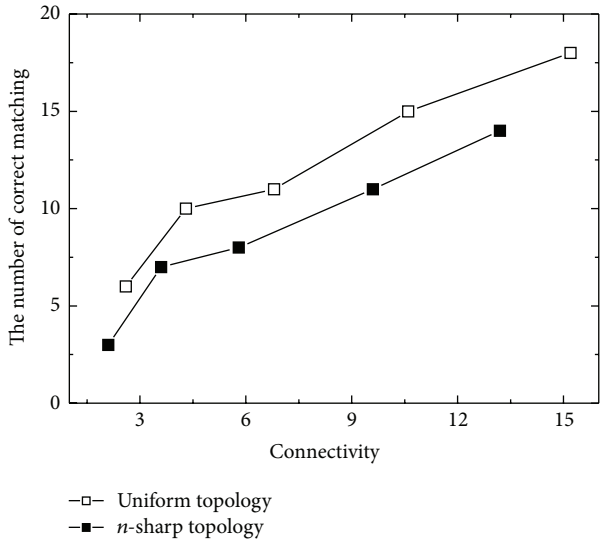


Figure 12: The results of BFM method in the small network.

radiomap, the absolute value of the difference of the two minimum Euclidean distances can be computed and sorted in descending order. The bigger the absolute value of the difference, the more unique the vertex features. So the vertices in front of the order are the most possible unique and distinguishable ones. They can actively catch their corresponding minimum Euclidean distance vertices in the advanced graph. To some extent, this method is subject to the symmetry of anchors in the blueprint. But we can artificially design the blueprint keeping asymmetric. Meanwhile, the irregular environment also affects the symmetry of the blueprint. Therefore, in practice, the weights of anchors in the blueprint are hardly perfectly symmetric.

6. Implantation and Experiment

6.1. Experiment Design. In our experiment, we will run MDS-KM algorithm on a variety of anchor topologies in the small and large networks. In the 30 m × 25 m room, the anchors are installed on the ceiling or concrete columns. (1) Figure 6(a) is the placement blueprint, where there are 24 positions to place anchor nodes. Concrete columns and wooden walls in the room are the principal obstacles affecting communication quality between anchors. (2) We simplify the topology of Figure 6(a) into an n-sharp topologies of 20 positions as shown in Figure 6(b).

In the large network, we simulate the anchors in the MATLAB placed with grid distribution and random distribution, respectively, as shown in Figures 7(a) and 7(b). A number of 100 nodes are placed uniformly and randomly in a $10\,r \times 10\,r$ multihop network, where 85 nodes are intermediate unknown-position nodes denoted by the circle, and 15 nodes are anchors denoted by the stars (*). For the purpose of facilitating the comparison of positioning error, we select the similar anchor positions in both topologies to construct the radiomap.

It should be noted that the complicated office room is more sensitive to the noise than outdoors. Meanwhile, the most indoor positioning systems are usually deployed deterministically according to the placement blueprint. Therefore, we choose the indoor environment as the case of the small network, which is more powerful to verify the MDS-KM performance.

6.2. The Small Network. During the radiomap construction, we set each anchor ID number multiplied by 100 milliseconds as its broadcast cycle to avoid the sending collision. After running 2 minutes, we compute the average RSS values between anchors. We use our MDS-MAP(A) method in the topologies (1) and (2) for constructing the absolute radiomap based on 3 random calibrated anchors, denoted by the stars (*) as shown in Figure 8. The circles represent the estimated absolute positions, and the solid lines represent the errors between the estimated positions and the true positions. The longer the solid line, the larger the positioning error. The transmitting power of TelosB in TinyOS system is classified into 1 to 31 levels. With the level rising, the transmitting power becomes higher. We set the highest level of transmitting power in this group of experiments. The results show that we have the average estimation errors of 3.05 m and 3.25 m in two topologies.

Figure 9 shows the average performance of MDS-MAP(A) positioning affected by connectivity and numbers of calibrated anchors. Figures 9(a) and 9(b) show the results of MDS-MAP(A) positioning of two topologies, respectively. We set the transmitting power levels as 11, 17, 21, 26, and 31, respectively, in our experiments. Three, five, and seven calibrated anchors are used. Then we get the connectivity levels of 2.6, 4.3, 6.8, 10.6, and 15.2 in the uniform topology, and 2.1, 3.6, 5.8, 9.6, and 13.2 in the n-sharp topology. With the lowering of the connectivity level, the positioning performance declines significantly. When the connectivity level is less than 3, the average error will be achieved to around 5.5 m. Besides, the positioning error becomes lightly lower with the increasing of C-anchors. Meanwhile, the different numbers of calibrated anchors also have very close positioning errors. Therefore, a certain range of a number of variations of calibrated anchors has no significant influence on positioning performance.

We obtain a radiomap with absolute coordinates after MDS-MAP(A) operation. Before running the KM matching, we need to set the parameter k for constructing a bipartite graph. In our experiment, we show the minimal k to producing a complete bipartite graph in Figure 10. With the connectivity level rising, the value of k reduces gradually. When the connectivity level is 15.2 in uniform topology and 9.6 and 13.2 in n-sharp topology, the value of k is 1. That means that the bipartite graph is already a one-to-one mapping complete graph. Then we can obtain the optimal matching between the blueprint and the radiomap without the KM method. Meanwhile, we find that this mapping is also exactly correct. Under other connectivity levels, we must use the KM method to find the optimal matching. We find that the rate of correct matching between anchors in the

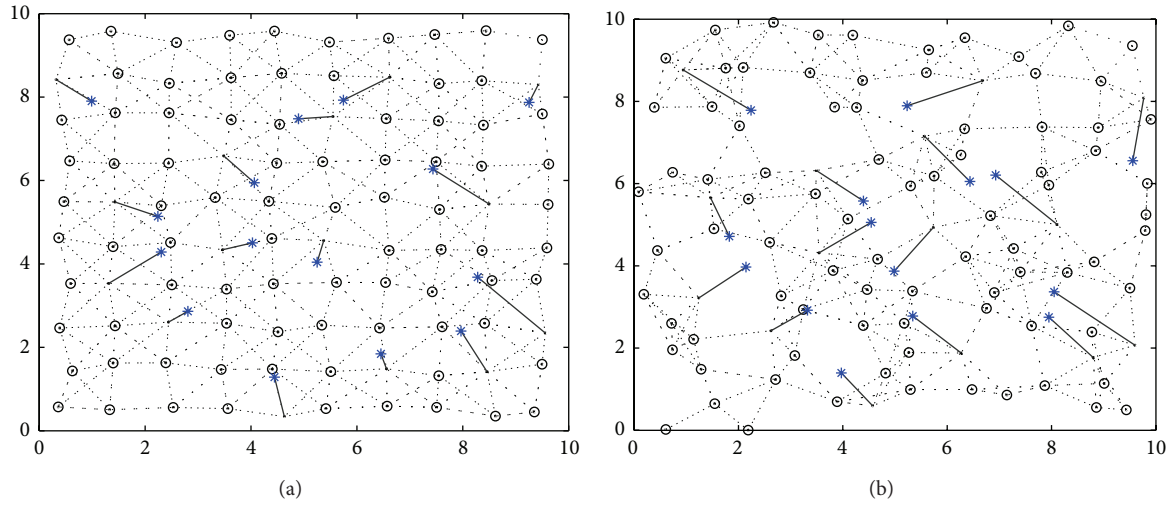


Figure 13: The average errors in both simulation scenarios.

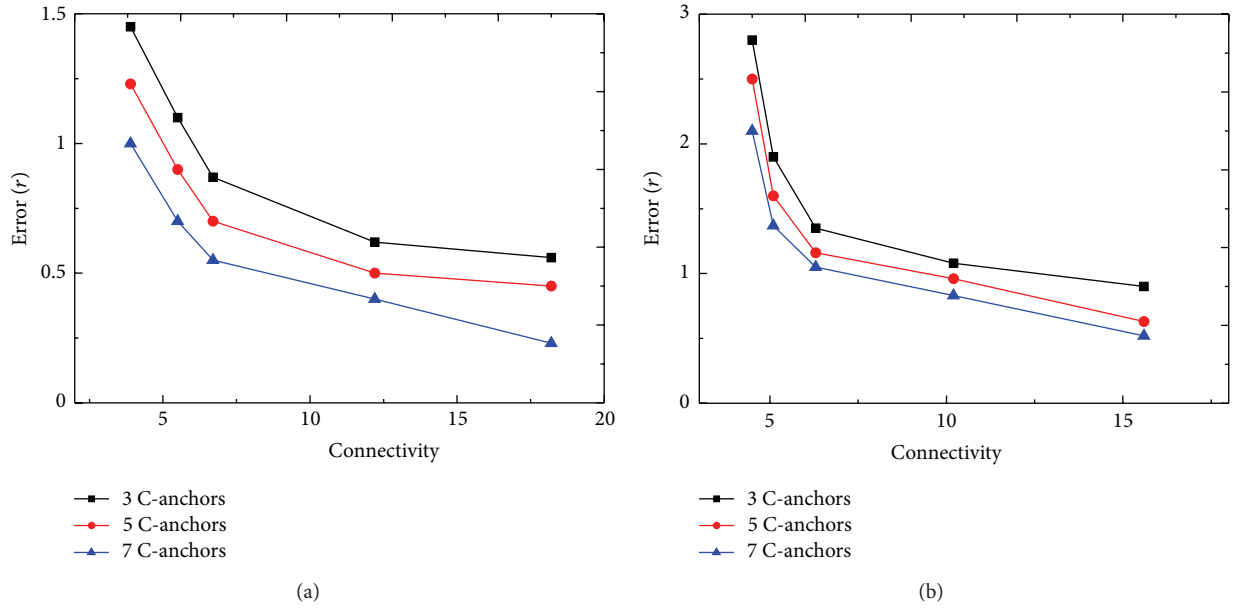


Figure 14: The error analysis in both simulation scenarios.

radiomap and positions in the blueprint can achieve 100% when connectivity level is over 3. Only when the connectivity level is less than 3, there are two anchor nodes with error mapping in both topologies, where the node IDs are 3 and 10, respectively. This is because both nodes are close to each other. The positioning error from the MDS-MAP(A) method will make their positions confused so that the maximum weighted matching of the KM method is not exactly the mapping from the radiomap to the blueprint. Meanwhile, we also observe that the more calibrated anchors cannot help the accuracy of the KM matching unless the anchors with error matching are calibrated anchors.

In order to validate the performance of our BFM algorithm, we need to exactly transform the physical distance of the blueprint into RSS value of the advanced graph. At first, we make a measurement test for determining the parameters WAF and α in (1). During our experiment, we test two types of obstacle materials, 40 cm width wooden wall and 60 cm $\times$ 60 cm width concrete column. Two TelosB nodes lie in two sides of obstacle and 2 m away from the obstacle. One node broadcasts beacon packet every 10 seconds, while another node receives the packet and computes the RSS value. We spend 80 minutes to get the results shown in Figure 11. We find that the wooden wall and concrete column can approximately reduce RSS 5 db and 10 db, respectively. Based on the measurement, we further compute the fading factor α in our environment, which is approximate to 3. Then we use the experimental values to construct the advanced graph.

Figure 12 is the number of correct matching anchors with connectivity increasing during the BFM process. The number

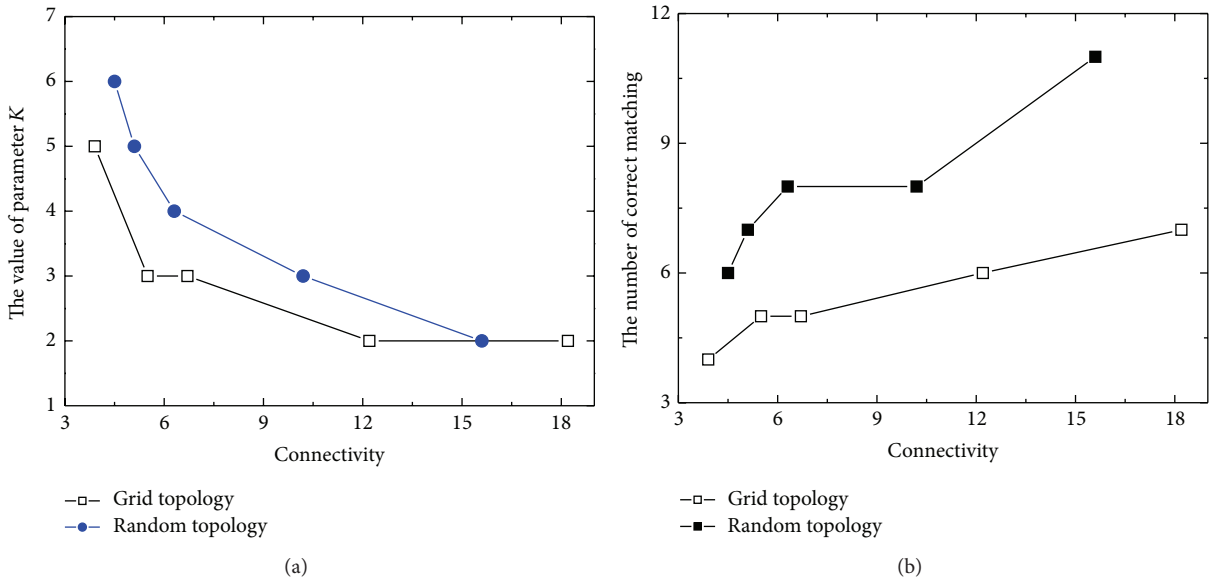


Figure 15: The results of BFM method in the large network.

1: $G(X, Y, W)$/*G is a bipartite Graph, W is the Weight.*/
2: **for all** ($x \in X$ and $y \in Y$) **do**
3: /*Initialize all vertices labeling*/
4: $l(x) = \text{Max}\{w(x, y), y \in Y\}$
5: $l(y) = 0$
6: **end for**
7: $M = \text{Hungarian}(G(X, Y, l))$
8: **if** (M is complete matching of G) **then**
9: $\widehat{M} = \text{save}(M)$
10: return $\widehat{M}$
11: **else**
12: relabeling(l)/*as KM rules*/
13: goto 7
14: **end if**

Algorithm 1: The Kuhn-Munkres Algorithm.

in the uniform case is lightly more than that in the random case, which is mainly due to more quantity of anchors in uniform topology. Meanwhile, we find that there are three or more anchors at least with correct matching even when the connectivity is lower than 3 in two topologies. Therefore, we can run our MDS-KM method in all the above experiments without any calibrated anchors, which further reduces the labor cost. But unfortunately our BFM method cannot help to solve the error mapping of the MDS-KM method under the lower connectivity.

6.3. The Large Network. We run MDS-MAP(A) method for the grid and random topologies of the large network to construct the absolute radiomap based on 3 random calibrated anchors as shown in Figure 13. The circles represent unknown-position intermediate nodes. The stars represent the anchor nodes, and the solid lines represent the errors between the estimated positions and the true positions. In the $10\,r \times 10\,r$ area, we set the communication range as $1.5\,r$ and $2\,r$, respectively, in the grid and random topologies. The average connectivity levels of both topologies are 6.7 and 6.3, respectively. Although both connectivity levels are similar, the positioning errors have a big difference. After running the MDS-MAP(A) method for the radiomap, we have the corresponding average estimation errors of $0.87\,r$ and $1.35\,r$ in both topologies. This is because the connectivity level of nodes in the random case is uneven so that its estimated error of hop distance is significantly bigger than that in the grid case. Therefore, the corresponding absolute radiomap in the random case has also a bigger average estimation error.

Additionally, we compare the performance of the MDS-MAP(A) method in different connectivity levels and calibrated anchors. In both topologies, we select 3, 5, and 7 calibrated anchors randomly to construct the absolute radiomap

during every trail. In the grid topology, the radio ranges are from $1\,r$ to $2\,r$, with an increment of $0.25\,r$, which result in the connectivity of 3.9, 5.5, 6.7, 12.2, and 18.2, respectively, as shown in Figure 14(a). We find that the higher connectivity level will bring about a better positioning result, and the more calibrated anchors also improve the positioning performance. When connectivity level is lower than 6.7 especially, the average estimated error will increase significantly. In the random topology, the radio ranges are from $1\,r$ to $3\,r$, with an increment of $0.5\,r$, which lead to average connectivity of 4.5, 5.1, 6.3, 10.2, and 15.6, respectively, as shown in Figure 14(b). This design is to compare the performance of the MDS-MAP(A) algorithm under the similar connectivity levels of both topologies. We can see that the positioning performance in the random topology has a significant reduction than that in the grid topology. The maximum average estimated error is even twice that in the grid topology. That is mainly because the estimated hop distance in the random topology is rather inaccurate.

Figure 15(a) is the k-value selection of both topologies. We can find that the k in the random topology has a higher value than that in the grid topology. This is because the higher errors of the estimated hop distance in the random topology produce the bigger position errors of the absolute radiomap. Thus the anchors in the radiomap cannot exactly correspond with the positions in the blueprint. In order to get a complete bipartite graph, k-value must be increased. Afterwards, we find that the KM method can reach a 100% rate of correct matching except that there are 3 and 2 error-matching anchors, respectively, under the connectivity of 4.5 and 5.1 in the random topology. It is further suggested that the MDS-KM algorithm is well suited to the higher connectivity network.

Figure 15(b) reflects the BFM method performance in both topologies of the large network. In the random topology, the BFM method can obtain a better feature matching result. This is because many vertices in the grid topology have the same hop count sequences subjected to the symmetry of anchor distribution. Therefore, the vertices invariants in the grid topology are hard to be distinguished, while in the random topology there are more distinguishable vertices with unique invariants. But in both topologies, we can also find that there are more than three anchors with correct feature matching. In other words, the MDS-KM method can run successfully in two simulation scenarios of the large network without any calibrated anchors.

7. Conclusion

In this paper, we consider the anchor self-positioning problem in detail. During the deterministic anchor placement, we design an efficient mapping algorithm between anchors and positions (MDS-KM) to avoid the expensive labor cost and error-prone features of artificial calibration. Additionally, we propose a best feature matching (BFM) method to obtain some mappings between anchors and positions in advance so that any calibrated anchors are not needed. Experimental results show that the MDS-KM algorithm can achieve the 100% correct matching between anchors and positions under a higher connectivity level. Meanwhile, in our experiments and simulations, the BFM method can obtain sufficient known-position anchors to support the successful running of the MDS-KM method.

Acknowledgments

This work is supported by the General Program of National Natural Science Foundation of China (NSFC) under Grant no. 61073180 and the National Key Basic Research Program of China (973) under Grant no. 2011CB302902.

References

[1] H. S. AbdelSalam and S. Olariu, "Towards enhanced RSSI-Based distance measurements and localization in WSNs," in *Proceedings of the IEEE INFOCOM Workshops 2009*, pp. 1–2, April 2009.

[2] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 11, pp. 961–974, 2004.

[3] R. Akl, K. Pasupathy, and M. Haidar, "Anchor nodes placement for effective passive localization," in *Proceedings of the International Conference on Selected Topics in Mobile and Wireless Networking (iCOST '11)*, pp. 127–132, October 2011.

[4] T. Kunz and B. Tatham, "Localization in wireless sensor networks and anchor placement," *Journal of Sensor and Actuator Networks*, vol. 1, no. 1, pp. 36–58, 2012.

[5] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '01)*, vol. 3, pp. 1655–1663, April 2001.

[6] J. N. Ash and R. L. Moses, "On optimal anchor node placement in sensor localization by optimization of subspace principal angles," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, pp. 2289–2292, April 2008.

[7] S. Hara and T. Fukumura, "Determination of the placement of anchor nodes satisfying a required localization accuracy," in *Proceedings of the IEEE International Symposium on Wireless Communication Systems (ISWCS '08)*, pp. 128–132, October 2008.

[8] R. Zemek, M. Takashima, S. Hara et al., "An effect of anchor nodes placement on a target location estimation performance," in *Proceedings of the IEEE Region 10 Conference (TENCON '06)*, pp. 1–4, November 2006.

[9] I. Borg and P. Groenen, "Modern multidimensional scaling: theory and applications," *Journal of Educational Measurement*, vol. 40, no. 3, pp. 277–280, 2003.

[10] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pp. 201–212, ACM, New York, NY, USA, June 2003.

[11] Y. Shang and W. Ruml, "Improved MDS-based localization," in *Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '04)*, vol. 4, pp. 2640–2651, March 2004.

[12] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proceedings of the 23th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '04)*, vol. 4, pp. 2652–2661, March 2004.

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

[14] D. C. Schmidt and L. E. Druffel, "A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices," *Journal of the Association for Computing Machinery*, vol. 23, no. 3, pp. 433–445, 1976.

[15] Sansone and M. Vento, "Subgraph transformations for the inexact matching of attributed relational graphs," *Computing*, vol. 12, pp. 43–52, 1998.

[16] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the 3rd annual ACM symposium on Theory of computing (STOC '71)*, pp. 151–158, ACM, New York, NY, USA, 1971.

[17] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[18] T. S. Rappaport, *Wireless Communications: Principles and Practice*, IEEE Press, Piscataway, NJ, USA, 1st edition, 1996.

[19] P. Bahl and V. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *Proceedings of the 9th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM '00)*, vol. 2, pp. 775–7784, 2000.

[20] K. Benkič, M. Malajner, P. Planinsic, and Z. Cucej, "Using RSSI value for distance estimation in wireless sensor networks based on ZigBee," in *Proceedings of the 15th International Conference on Systems, Signals and Image Processing (IWSSIP '08)*, pp. 303–306, June 2008.

[21] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, "Virtual calibration for RSSI-based indoor localization with IEEE 802.15.4," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, June 2009.

[22] K. Srinivasan and P. Levis, "RSSI is under appreciated," in *Proceedings of the 3rd Workshop on Embedded Networked Sensors (EmNets '06)*, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 867385, 11 pages
http://dx.doi.org/10.1155/2013/867385

Research Article

An Overlapping Clustering Approach for Routing in Wireless Sensor Networks

Zhenquan Qin, Can Ma, Lei Wang, Jiaqi Xu, and Bingxian Lu

School of Software, Dalian University of Technology, China

Correspondence should be addressed to Lei Wang; lei.wang@dlut.edu.cn

Received 10 January 2013; Accepted 13 February 2013

Academic Editor: Lei Shu

The design and analysis of routing algorithm is an important issue in wireless sensor networks (WSNs). Most traditional geographical routing algorithms cannot achieve good performance in duty-cycled networks. In this paper, we propose a k-connected overlapping clustering approach with energy awareness, namely, k-OCHE, for routing in WSNs. The basic idea of this approach is to select a cluster head by energy availability (EA) status. The k-OCHE scheme adopts a sleep scheduling strategy of CKN, where neighbors will remain awake to keep it k connected, so that it can balance energy distributions well. Compared with traditional routing algorithms, the proposed k-OCHE approach obtains a balanced load distribution, consequently a longer network lifetime, and a quicker routing recovery time.

1. Introduction

Studying the behavior of dynamic sensor networks becomes a hot topic. Movements of nodes make the wireless sensor networks (WSNs) [1] a dynamic one. These nodes can communicate with each other in wireless communication radius without any static network interactions. An important issue in dynamic geographical networks is the design and analysis of routing algorithms. Due to the limited communication range of the wireless transceivers, mobile nodes cannot communicate with other nodes unless they are within each other's geographical regions [2, 3]. Thus, it may be necessary for a mobile node to require aids of other nodes after checking their geographical routing information in forwarding data packets to its destination.

It will be much more uncertain when we consider the routing issue in a duty-cycled network, since the duty-cycle scheduling aims to prolong the network lifetime [4, 5] by making some nodes sleep and wake up when packets transmission occurs. Studies on adopting conventional routing protocols in wireless sensor network in a dynamic convention have been generally discussed [3, 6]. Our interest in the routing problem in a duty-cycled network falls into the following two aspects: (1) existing routing algorithm could place a heavy load on a newly joined node due to routing table updates and (2) the wireless network connectivity.

Conventional routing algorithms concentrate on finding the shortest path, without much concern about critical issues such as energy efficiency and network lifetime. The problem we discuss here is how to route efficiently in a duty-cycled sensor network. The basic idea behind the algorithms is to divide the network into a number of overlapping clusters. A node's sleep scheduling leads to a change in the network topology, then the membership of cluster changes as well. We propose a cluster formation scheme called k-OCHE, which selects cluster heads considering energy availability firstly, and then the cluster heads recruit cluster members. This cluster creation scheme can well be adopted in different routing circumstances.

The rest of this paper is organized as follows. Section 2, we survey related work. Section 3 defines network model, sleeping scheduling model, energy consumption model, and some notations. In Sections 4 and 5, we further detail the k-OCHE scheme and routing algorithm. We present specifics of simulation experiments which validate the correctness of the proposed algorithm in Section 6. Finally, Section 7 concludes the paper.

2. Related Work

Similar to other networks, overload balance, prolonging lifetime, and scalability are the major design concerns of wireless sensor networks. In conventional multihop communications in WSNs, sensors close to the sink are often overloaded, resulting in increased latency and reduced network life span. Such overload might cause latency in communication and reduce life span of network. In addition, the original architecture is not scalable for larger set of sensors covering a wider area of interest. To allow the network to cope with additional load and to be able to afford a large area of interest, clustering routing has been pursued. The main aim of clustering routing is to efficiently maintain the energy consumption of sensor nodes by involving them in multihop communication with a particular cluster and by performing data aggregation and fusion to reduce the number of transmitted messages to sink.

2.1. Routing Protocols. Many routing protocols [7–11] have been studied in the field of wireless sensor networks. Karp and Kung present greedy perimeter stateless routing (GPSR) [7], a novel routing protocol for wireless datagram networks that uses the positions of routers and a packet's destination to make packet forwarding decisions. GPSR makes greedy forwarding decisions using only information about a router's immediate neighbors in the network topology. When a packet reaches a region where greedy forwarding is impossible, the algorithm recovers by routing around the perimeter of the region. By keeping state only about the local topology, GPSR scales better in perrouter state than shortest-path and ad-hoc routing protocols as the number of network destinations increases. Under mobility's frequent topology changes, GPSR can use local topology information to find correct new routes quickly.

Shu et al. propose an efficient two-phase geographic greedy forwarding (TPGF) [8] routing algorithm for WMSNs. TPGF takes into account both the requirements of real-time multimedia transmission and the realistic characteristics of WMSNs. It finds one shortest (near shortest) path per execution and can be executed repeatedly to find more on-demand shortest (near shortest) node-disjoint routing paths. TPGF supports three features: (1) hole bypassing, (2) the shortest path transmission, and (3) multipath transmission, at the same time. TPGF is a pure geographic greedy forwarding routing algorithm, which does not include the face routing, for example, right/left hand rules and does not use planarization algorithms, for example, GG or RNG. This point allows more links to be available for TPGF to explore more routing paths and enables TPGF to be different from many existing geographic routing algorithms.

However, these traditional routing algorithm will overload relay nodes with the increase in sensor density. Besides, convergence characteristics of these algorithm are not good enough to meet the need of dynamic networks, such as duty-cycled networks.

2.2. Cluster-Based Approach for Routing. In the last few years, many algorithms have been proposed for clustering routing in wireless sensor networks [12–24].

In [14], Heinzelman et al. have proposed a distributed algorithm for wireless sensor networks (LEACH) in which sensors randomly select themselves as cluster heads with some probability and broadcast their decisions. The remaining sensors join the cluster of the cluster head that requires minimum communication energy. LEACH is one of the most popular clustering routing algorithms for sensor networks and is completely distributed. However, LEACH uses single-hop routing where each node can transmit directly to the cluster head and the sink. Besides, there are a number of clustering algorithms constructing clusters not more than 1-hop away from a cluster head, such as DCA [15] and DMAC [17]. Similar to these, Baker and Ephremides [13] propose overlapping cluster with $k = 1$. In large networks single-hop clustering, as shown in Figure 2, may generate a large number of cluster heads and eventually lead to the same problems as if there is no clustering.

In addition, the TEEN [16] and APTEEN [18] are hierarchical protocols designed to be responsive to sudden changes in the sensed attributes such as temperature. Younis et al. [19] have proposed a different hierarchical routing algorithm based on a threetier architecture.

To the best of our knowledge, there is only one clustering algorithm that specifically controls overlapping in the formation of clusters, that is, KOCA [20]. Goal of KOCA is to ensure that the entire network is covered with connected overlapping clusters considering a specific average overlapping degree. KOCA is still a static clustering in which the cluster formation is not changed all the time. This condition causes an unbalance load among all nodes. A node that roles as cluster head (CH) will get more load than a non-CH and so that it will die faster. The death of CHs will break the whole network because the link between nodes and center will be broken. Therefore, KOCA cannot be applied in actual situation commendably. Rotating the CH role distributes this higher burden among the nodes, thereby preventing the CH from dying prematurely. To overcome this problem, we propose k-OCHE which allows a node to go to sleep while keeping its neighbors k connected, thus the role of CH can be rotated and the load of CH can be balanced. The most important is that k-OCHE is the first approach to combine the interior cluster routing with exterior cluster routing which achieves balanced load distribution, longer network lifetime, and quicker routing.

3. Assumptions and Notations

3.1. Network Model. We consider a multihop wireless sensor network where all nodes are alike. We assume that each node has a unique id. The locations of sensor nodes can be obtained by GPS. All sensors transmit at the same power level and hence have the same transmission range T_r. Each sensor node is aware of its geographic location and its 1-hop neighbor nodes' geographic locations. We assume that sensor nodes can know the location of base station by receiving the packet, which comes from there. This assumption is the same as that used in [7, 25, 26].

All communications are over a single shared wireless channel. A wireless link can be established between a pair

```
(*Run the following at each node u*)
(1) Get the information of current remaining energy EA.
(2) Set CH_table_wait timer;
(3) if EA ≥ Threshold then
(4)   status = CH;
(5)   Broadcast CH_AD containing (CHID, CHEA, HC, SEA) to neighbors.
(6)   if Receive a CH_AD then
(7)     If CH_AD.CHEA > EA then
(8)       status = NORMAL;
(9)     end if
(10)  end if
(11)  Receive J_AD from neighbors.
(12)  if CH_table does not contains J_AD.NID then
(13)    Add J_AD.NID to CH_table;
(14)    CH_table_wait timer fires;
(15)    Broadcast CH_table to its members.
(16)  end if
(17) else
(18)  status = NORMAL;
(19)  Receive a message.
(20)  if timer ≥ HALF_OF_CKN_CYCLE then
(21)    status = CH;
(22)  end if
(23)  if message == CH_AD from CH(s) then
(24)    Add CH_AD.CHID to CH_table;
(25)    CH_table.PID = CH_AD.SID;
(26)    CH_table_wait timer fires;
(27)    Send J_AD to its parent node.
(28)    if CH_AD.HC < n then
(29)      HC=HC+1;
(30)      Broadcast CH_AD to neighbors.
(31)    end if
(32)  else
(33)    if message == J_AD then
(34)      Send the J_AD to its parent node.
(35)    else
(36)      if message == CH_table from CH(s) then
(37)        Update its CH_table.
(38)      end if
(39)    end if
(40)  end if
(41)  if its CH_table has more than one CHID then
(42)    Status = Boundary;
(43)    Add CH_table.CHID to NCH_table;
(44)    Send its CH_table and NCH_table to its parent node.
(45)  else
(46)    Receive CH_table and NCH_table from Boundary nodes.
(47)  end if
(48) end if
```

Algorithm 1: The k-OCHE algorithm.

of nodes only if they are within wireless range of each other. The k-OCHE algorithm only considers bidirectional links. It is assumed that MAC layer will mask unidirectional links and pass bidirectional links to k-OCHE. We refer to any two nodes that have a wireless link as 1-hop or immediate neighbors. Nodes can identify neighbors using beacons.

3.2. Sleeping Scheduling Model. To ensure the network connectivity and prolong its lifetime, we assume that all nodes operate under CKN-based [27] sleep/awake duty cycling. Time is divided into epochs, and each epoch is T. On each epoch, nodes run CKN, shown in Algorithm 2, to decide whether to be awake. A node can go to sleep assuming that at least k of its neighbors remain awake to keep it k connected.

(*For each node u^*)
(1) Pick a random rank $rank_u$.
(2) Broadcast $rank_u$ and receive the ranks of its currently awake neighbors N_u. Let R_u be the set of these ranks.
(3) Broadcast R_u and receive R_v from each $v \in N_u$.
(4) If $|N_u| < k$ or $|N_v| < k$ for any $v \in N_u$, remain awake.
Return.
(5) Compute $C_u = \{v | v \in N_u$ and $rank_v > rank_u\}$;
(6) Go to sleep if both the following conditions hold. Remain awake otherwise.
(i) Any two nodes in C_u are connected either directly themselves or indirectly through nodes within u's 2-hop neighbors that have *rank* larger than $E\,rank_u$;
(ii) Any node in N_u has at least k neighbors from C_u.
(7) Return.

Algorithm 2: Connected K neighborhood (CKN).

And nodes reach a consensus to take turns to sleep, while the whole network is globally connected. Therefore, by changing the value of k, the network can manipulate the sleep rate s, so that it can proceed with further work with clustering.

3.3. Energy Consumption Model. We use the same radio model defined in [28]. The amount of energy required to transmit an L-bit message over a distance x is $E_{\mathrm{TX}}(L, x)$ given by (1)

$$E_{\mathrm{TX}}(L, x) = \begin{cases} E_{\mathrm{elec}} \cdot L + \epsilon_{fs} \cdot L \cdot x^2, & \text{if } x \le d_0, \\ E_{\mathrm{elec}} \cdot L + \epsilon_{mp} \cdot L \cdot x^4, & \text{if } x \ge d_0. \end{cases} \quad (1)$$

E_{elec} is the energy dissipating to power the transmitter or receiver circuitry. The parameters ϵ_{fs} and ϵ_{mp} are the amount of energy dissipating per bit in the radio frequency amplifier according to the distance d_0, which is given by (2)

$$d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}. \quad (2)$$

The energy consumed by receiving this packet is $E_{\mathrm{TX}}(L, x)$ shown by (3),

$$E_{\mathrm{TX}}(L, x) = L \cdot E_{\mathrm{elec}}. \quad (3)$$

3.4. Notations. (1) Network size (n): the number of nodes in the network. Sensor nodes are deployed randomly in a square area with side length of l.

(2) Energy consumption rate (EC): for each node, $\mathrm{ER_i}$ denotes residual energy on the battery, and then EC can be defined as (4),

$$\mathrm{EC} = \frac{\mathrm{ER_i}}{E_{\mathrm{initial}}}, \quad (4)$$

where E_{initial} is the initial energy of each node.

(3) Energy availability (EA):

$$\mathrm{EA_i} = E_{\mathrm{initial}} - \mathrm{EC} * t. \quad (5)$$

(4) Minimum number of awake neighbors in an epoch for each node (k): through varying the value of k, we can keep the network k connected and optimize the geographic routing performance.

(5) Average node degree (d): the average degree of a node u is the number of its neighbor nodes. The relation between the average node degree (d) and the radio range (r) of a node is given by

$$d = \frac{n\pi r^2}{l^2} = \mu\pi r^2. \quad (6)$$

4. The k-OCHE Algorithm

4.1. The Proposed Algorithm. In k-OCHE, shown in Algorithm 1, a node can go to sleep assuming that at least k of its neighbors remain awake to keep it k connected. Given a k, we can obtain a new network topology within the range of awake nodes in each cycle. Each node can have three possible states: cluster heads (CHs), boundary nodes (BNs), and normal nodes. A cluster head possesses information about not only its own cluster (such as member nodes' IDs) but also adjacent clusters (such as boundary node and adjacent clusters' members). A boundary node belongs to multiple overlapping clusters connecting different clusters to transfer and forward data, and it improves the network robustness effectively. Normal nodes are internal nodes that belong only to one cluster.

In this section, we present the description of the cluster head's selection process as well as the clusters' generation; we then give an example to illustrate a cluster generated by k-OCHE.

4.2. Cluster Head Selection Procedure. In k-OCHE approach, the important operation is to select a set of cluster heads (CHs) among the nodes in the network and recruit the normal nodes as these CHs' members. The k-OCHE approach adopts EA to select CHs, and EA is defined as the battery residual energy after certain consumption during a certain period. At the beginning of each cycle, each node compares its EA to the threshold (the threshold is an empirical value, which is used to control the number of CHs in the network. The optimal threshold is obtained when the CH nodes take 15% [20] of all the nodes in the network. And we use this

value for experiments presented in this paper.), if its EA is bigger than the threshold, then it becomes a CH and advertises itself as a CH to the sensors within its transmission range to recruit cluster members. This advertisement (*CH_AD*) is forwarded to all sensors that are no more than n hops away from the CH through controlled flooding. The recruitment message's (*CH_AD*) header includes CHID, SID, EA, and HC, where CHID is cluster head ID, SID is the sender node ID, and HC is the number of hops leading to the CH node. The HC field is used to limit the flooding of the *CH_AD* message to n hops. By receiving the recruit message from CHs, a sensor node joins those clusters no matter whether it has belonged to a cluster.

However, if a CH u receives a recruit from another CH v, and v has the higher EA, then u gives up being a CH and joins v's cluster. Since the *CH_AD* forwarding is limited to n hops, if a sensor does not receive a CH advertisement within a reasonable time duration, it can infer that it is not within n hops of any cluster head and hence become a CH. In k-OCHE, the maximum time that a node should wait for CH advertisement message is set to half cycle of CKN. Note that this is a distributed algorithm and does not demand clock synchronization between the sensors.

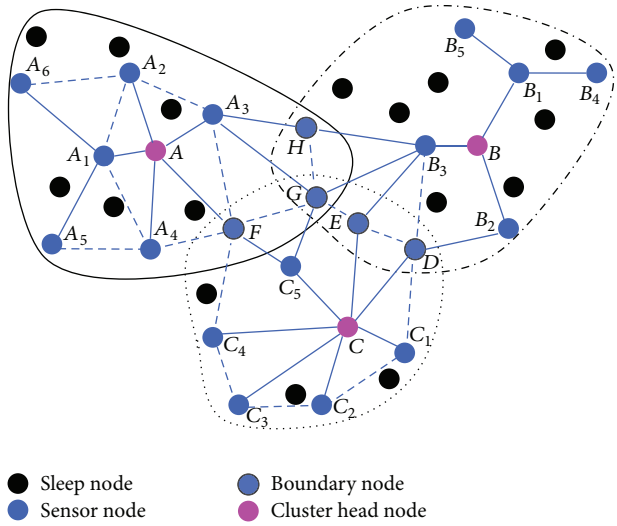


Figure 1: An example of overlapping clusters in a network with 38 nodes. After executing k-OCHE, the network selects three cluster heads, and each cluster contains 9 normal nodes. Clusters can communicate with each other through boundary nodes.

4.3. Cluster Generation. Each node maintains a table, *CH_table*, that stores information about the clusters it belongs to. Upon receiving a new *CH_AD* message, a node will add an entry in its *CH_table* and check the HC field in the message. Then the node updates HC and parent fields in the corresponding entry in the CH table if the recent message came over a shorter path. Often a message traveling the shortest path in terms of the number of hops would arrive first. However, delay may be suffered at the MAC or link layers. For every entry in its *CH_table*, a node sends a join advertisement (*J_AD*) message to CH in order to become a member of the corresponding cluster. To limit the flooding, the *J_AD* message is unicasted using the field *CH_table.parent*. The *J_AD* message has the form [SID, CHID, EA] where SID is the ID of the node that will join the cluster and CHID is the ID of the CH node responsible for this cluster. Upon receiving the *J_AD* message, the parent node will add SID to its children field. When a CH node receives join advertisement (*J_AD*) sent by an ordinary node, it will compare the number of member nodes to threshold to admit new member and update the count of cluster nodes if the size is smaller than threshold or else abandon the request. Supposing that the rejected node has cluster head already, the clustering process ceases. Otherwise, it looks for another appropriate cluster to join in. There is only one single CHID entry in a ordinary node's CH table, because it belongs to one cluster head, while the overlapping cluster node which connects different clusters has multiple CHID entries. Each cluster head maintains a list of all cluster members, a list of adjacent clusters, and a list of boundary nodes.

The k-OCHE algorithm avoids the fixed cluster head scheme, with periodic replacement done by sleep scheduling mechanism to balance the node energy consumption. All cluster members send the current state information to cluster

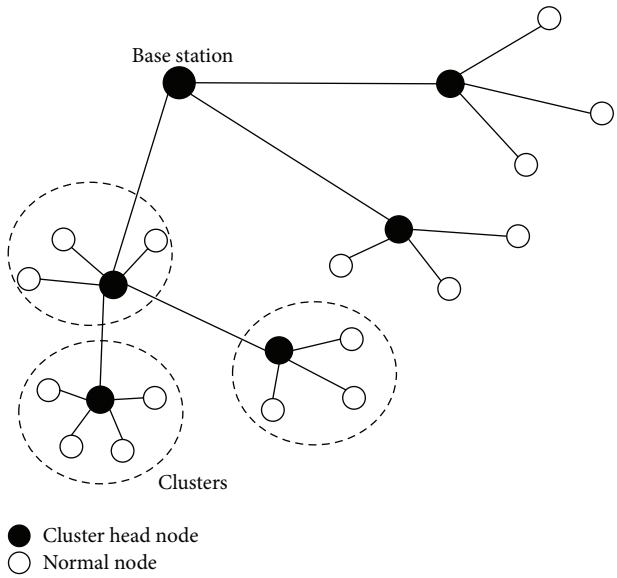


Figure 2: Single hop clustering. Each node in the network is not more than 1-hop away from a cluster head.

head, and k-OCHE chooses the node with the highest EA to be the new head. When the new cluster head gets cluster head's notification, it broadcasts recruitment message and the new cluster forming phase triggers. The process can reduce the energy consumption of broadcast of temporary head.

Note that k-OCHE stops in $O(n)$ steps. As n is a constant value, the clustering process terminates in a constant number of iterations regardless of the network size.

4.4. An Illustrative Example. We demonstrate our algorithm with Figure 1, which is selected from the simulation results

Table 1: CH table of A.

CHID	PID	PEA	CID	CRID	CREA
			A_1, A_2, A_3	A_1, A_2, A_3	5 J, 4 J, 4 J
A	Null	Null	A_4, A_5, A_6	A_4, A_1, A_1	3 J, 4 J, 5 J
			F, G, H	F, A_3, A_3	5 J, 3 J, 4 J

optionally, and we add some necessary information to make it more comprehensible.

In Figure 1, there are three clusters with three corresponding cluster heads A, B, and C, and nodes with black frame in the overlapping area are boundary nodes. Note that two cluster heads are not immediate neighbors. Since boundary nodes belong to multiple clusters, their tables contain CHs of those clusters. A, B, and C can communicate with each other through their common boundary nodes in their neighbor cluster head tables.

5. Routing Algorithm

We first discuss the necessary data structures to be maintained at each node for the routing algorithm, as shown in Table 7. We then explain the routing construction and recovery procedures in the network. The routing construction can be divided into two phases: interior cluster routing and exterior cluster routing.

During the interior and exterior routing phase, routes are constructed between all pairs of nodes. The routing recovery phase takes care of maintaining routing table considering sleep schedule and recovering from an individual node failure.

5.1. Interior Cluster Routing. After cluster head selection and cluster generation procedure, each node completes the construction of two tables: *CH_table* and *NCH_table*. *CH_table* stores the information of cluster it belongs to and *NCH_table* stores the information of its neighbor clusters. In interior cluster routing construction phase, each node constructs interior cluster routing table according to *CH_table* and *NCH_table*. At each node u,

(i) when *routing_table.Des* is u's child or grandchild, u updates *routing_table.next* by using *CH_table.CRID*,

(ii) when *routing_table.Des* is in the same cluster with u, u updates *routing_table.next* by using *CH_table.PID*,

(iii) when *routing_table.Des* is in neighbor cluster, u updates *routing_table.next* by using *NCH_table.RID*,

(iv) when *routing_table.Des* is one of the others, u updates *routing_table.next* by using *CH_table.PID*.

Let us illustrate it with an example. Tables 1, 2, 3, 4, 5, and 6 are *CH_tables* and *NCH_tables* of A, A_3, and H. For cluster A, we take the CH node A, a normal node A_3, and a boundary node H into consideration. As the structure of cluster is a spanning tree, the root is A, level-1 children are A_1, A_2, A_3, A_4, and F, and level-2 children are A_5, A_6, H, and G.

Table 2: CH table of A_3.

CHID	PID	PEA	CID	CRID	CREA
A	A	5 J	G, H	G, H	4 J, 5 J

Table 3: CH table of H.

CHID	PID	PEA	CID	CRID	CREA
A	A_3	5 J	Null	Null	Null
B	B_3	3 J	Null	Null	Null

Table 4: NCH table of A.

NCHID	NCHEA	RID	REA
B	4 J	A_3	4 J
C	5 J	A_3, F	3 J, 5 J

Table 5: NCH table of A_3.

NCHID	NCHEA	RID	REA
B	3 J	H, G	4 J, 3 J
C	4 J	G	5 J

Table 6: NCH table of H.

NCHID	NCHEA	RID	REA
A	3 J	A_3	5 J
B	5 J	B_3	4 J

In the routing table of A, when destinations are A_1 to A_6, F, G, and H, A will check *CH_table.CRID* to choose the next hop, as they are all its level-1 or level-2 children. When destinations are $B.*$ and $C.*$, A will check *NCH_table.RID* to choose the next hop, as they are in neighbor clusters. For the other destinations, A uses exterior cluster routing which will be discussed in the following section.

In the routing table of A_3, when destinations are G and H, A_3 will check *CH_table.CRID* to choose the next hop, as they are its children. When destinations are $B.*$ and $C.*$, A_3 will check *NCH_table.RID* to choose the next hop, as they are in neighbor clusters. Considering that *CH_table.CRID* has two nodes, G and H, A_3 will choose the node of higher EA to be the next hop. For the other destinations, A_3 will check *CH_table.PID* to choose the next hop.

In the routing table of H, when destinations are in neighbor cluster, H will check *NCH_table.RID* to choose the next hop, as they are in neighbor clusters. For the other destinations, H will check *CH_table.PID* to choose the next hop. Considering that *CH_table.PID* has two entries, H will choose the node of higher EA to be the next hop. The results of routing table of A, A_3, and H are shown in Tables 8, 9, and 10.

5.2. Exterior Cluster Routing Construction. We consider each cluster as a node in exterior cluster routing phase. Each CH node takes the responsibility of each cluster. An original routing algorithm (e.g., GPSR and TPGF algorithms) is running exterior clusters. As shown in Figure 3, each hexagon

Table 7: Data structures table.

Data structures	Description
CH_AD (CHID, CHEA, HC, SEA)	A message containing CHID (the ID of cluster head), CHEA (the energy availability of cluster head), HC (hop count), and SEA (the energy availability of message sender)
J_AD (ID, SEA)	A message containing ID (the ID of join node), SEA (the energy availability of message sender)
B_AD (NCHID)	A message containing NCHID (the ID of neighbor cluster head)
CH_table (CHID, PID, PEA, CID, CRID, CREA)	A table containing CHID (the ID of cluster head), PID (the ID of parent node), PEA (the energy availability of parent node), CID (the ID of its children node), CRID (the ID of children's relay node), and CREA (the energy availability of children's relay node)
NCH_table (NCHID, RID, REA)	A table containing NCHID (the ID of neighbor cluster head), RID (the ID of neighbor cluster head's relay node), and REA (the energy availability of neighbor cluster head's relay node)
Routing_table (destination, next hop)	A routing table containing destination and next hop

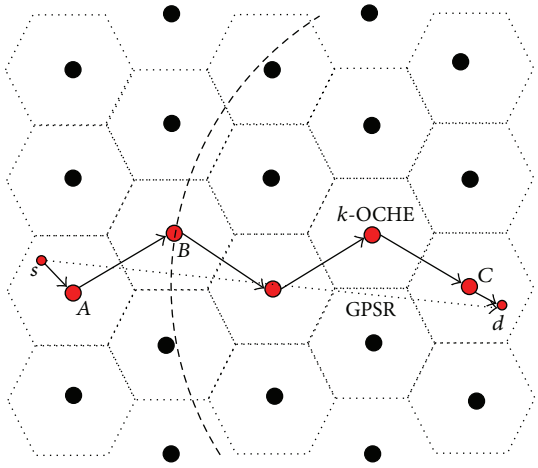


Figure 3: Exterior cluster routing. Nodes s and d are source and sink, respectively.

Table 8: Routing table of A.

Destination	Next hop
$A.A_1 \cdots A.A_6$	$A_1, A_2, A_3, A_4, A_1, A_1$
$A.F$	F
$A.G$	A_3
$A.H$	A_3
$B.*$	A_3
$C.*$	A_3/F (higher EA)
$*.*$	Outer Routing

presents a cluster and the black nodes are CH nodes. The normal nodes are ignored except source and sink. When source node s sends a packet to sink node d, s node will run interior cluster routing and then packet will be relayed to CH node A. At that time, A will run exterior cluster routing to decide which neighbor cluster to be the relay cluster and then run interior cluster routing to the CH node of relay cluster. In general, as shown in Figure 3, the routing path of GPSR is straight while k-OCHE's path is sinuous and adaptive to nodes' energy availability which can obtain a balanced load distribution and consequently a longer network lifetime.

Table 9: Routing table of A_3.

Destination	Next hop
$A.G$	G
$A.H$	H
$B.*$	H/G (higher EA)
$C.*$	G
$*.*$	A

Table 10: Routing table of H.

Destination	Next hop
$A.*$	A_3
$B.*$	B_3
$*.*$	A_3/B_3 (higher EA)

5.3. Routing Maintenance. This phase begins when nodes' status change due to duty-cycle scheduling. The route maintenance in our approach basically boils down to cluster maintenance. After a change in topology, all the nodes have the complete cluster information in the form of *CH_table* and *NCH_table*. If all CH nodes have a consistent view of the topology, routing loops will not form. However, due to long propagation delay, network partitions, and so forth, some nodes may have inconsistent topology information. This might lead to formation of routing loops. However, these loops are short term, because they disappear within bounded time (required to traverse the diameter of the network).

The new cluster information will be propagated throughout the network. Among exterior neighbor clusters, it should be noted that only the boundary nodes are responsible for broadcasting and rebroadcasting any new information. This helps in quick dissemination of information across the network. Thus, the convergence of the cluster-based protocols is very quick. When a node of a cluster stops working, after

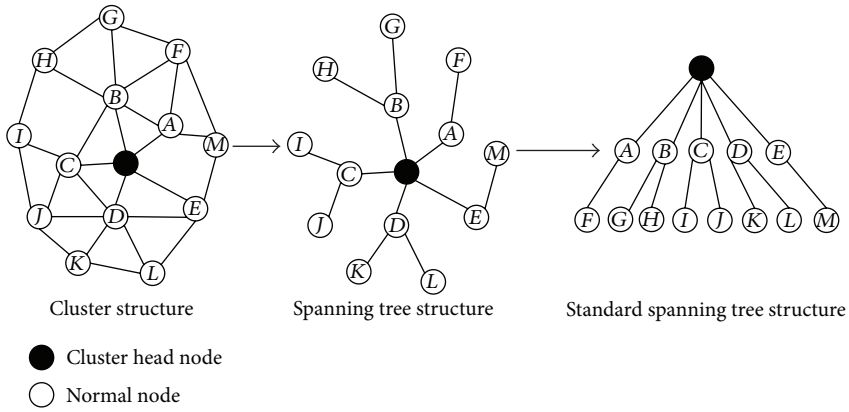


Figure 4: Each cluster can be transformed into a spanning tree.

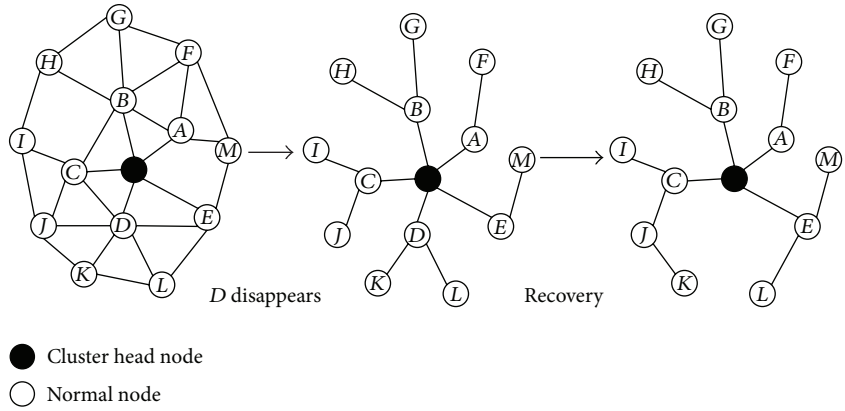


Figure 5: Routing recovery. When node D disappears, k finds its new parent J and L finds its new parent E.

a certain time, all its neighbors will detect this event. In interior cluster, only its parent node will update the information of *CH_table* and alarm this event to CH node. If the dead node has children, each child will select one neighbor who has the highest EA as its new parent.

Let us illustrate it with an example, as shown in Figures 4 and 5. Let node D disappear. This event will be detected by nodes C, E, L, K, and J and CH node. Since nodes C, E, L, K, and J are not parent nodes, they will just update *CH_tables* to indicate the change. The parent node does not need to forward this event to CH as CH is itself; otherwise it needs to relay the event to CH. Node K and node L, as D's children, have to look for their new parents. As J (E) is K's (L's) only neighbor which has the highest EA, it becomes K's (L's) new parent.

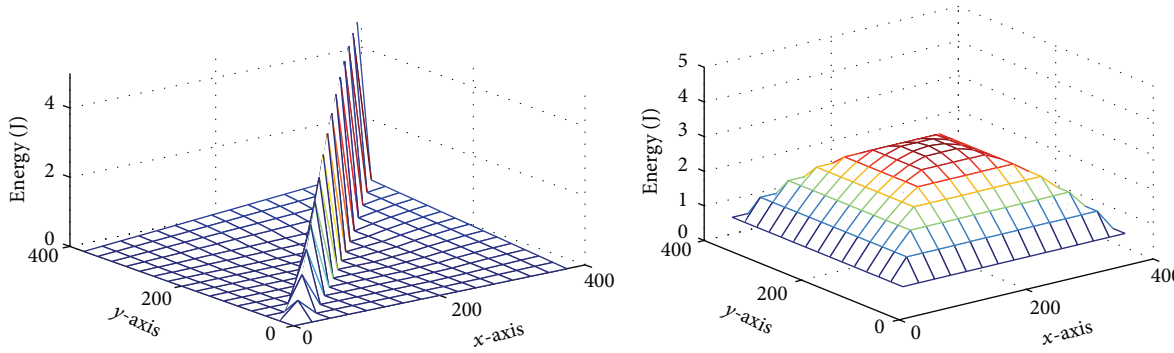
6. Performance Evaluation

6.1. Experiment Setup. To verify the correctness and effectiveness of the proposed k-OCHE algorithm, we conduct a detailed simulation using the NetTopo [29]. In our simulation, the studied WSN has the topology: $400 * 400\,m^2$. The number of deployed sensor nodes ranges from 200 to 1000 (each time increased by 100). The value of k changes from 1 to 10 (each time increased by 1). For each number of deployed sensor nodes, we use 100 different seeds to generate 100 different network deployments. A source node is deployed at the location of (50, 50), and a sink node is deployed at the location of (350, 350). We use the GPSR routing protocol implemented in routing layer of the simulator to deliver message. All simulation parameters [30] are listed in Table 11. In Figures 9(a), 9(b), 9(c), 9(d), 9(e), and 9(f), the execution of k-OCHE is demonstrated.

Table 11: Simulation parameters.

Variables	Values
Communication range	30 m
Number of nodes	400
Total energy of each sensor	5 Joules
Packet size	240, 1200 bits
Energy dissipated for receiving	50 nJ/bit
Energy dissipated for transmission	50 nJ/bit
Energy dissipated for transmit amplifier	100 pJ/bit/m^2

In this section, we evaluate traditional GPSR and GPSR in k-OCHE approach in terms of energy consumption, network lifetime, and recovery time.



(a) Energy consumption of traditional GPSR

(b) Energy consumption of k-OCHE

FIGURE 6: Comparison of energy consumption in traditional GPSR and in k-OCEH-based GPSR.

6.2. Energy Consumption. First we compare energy consumption of each sensor in traditional GPSR and k-OCHE-based GPSR, which can reflect lifetime of network. Figure 6 reports energy consumption of sensors in two protocols after 600 routing queries have been processed, where x-axis and y-axis together decide the location of each sensor node and z-axis represents the value of energy consumption. Figure 6(a) shows that some sensors in traditional GPSR consume a lot of energy, especially those located along the two edges and diagonal line of the sensor field to which the data sink belongs. So these sensors are energy hungry ones which consume all 5 joules, while sensors located outside this region just consume as little as 0.1 joules after 600 queries. Obviously, the energy consumption in traditional GPSR is very unbalanced. On the contrary, the load in k-OCHE-based GPSR balances very well, as shown in Figure 6(b), where no energy intensive nodes exist. In k-OCHE-based GPSR the maximum energy consumption is 3 J and the minimum energy consumption is 0.04 J. In other words, in the k-OCHE-based GPSR, by consuming 5 J, the sensor network can process routing at least 1000 times.

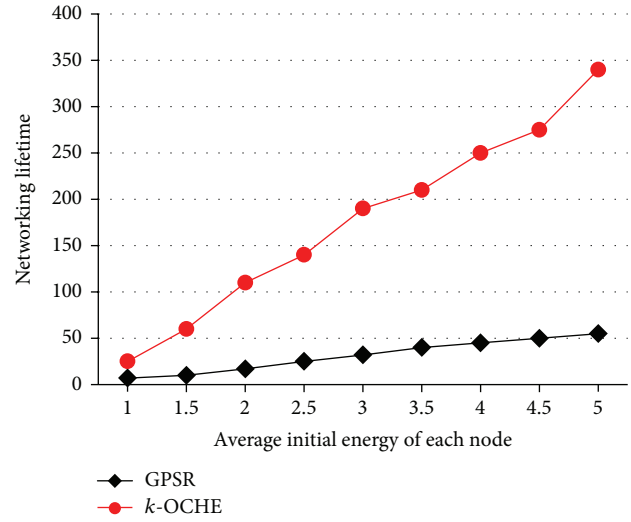


FIGURE 7: Comparison of network lifetime by using traditional GPSR and k-OCHE-based GPSR.

6.3. Network Lifetime. Finally, we compare network lifetime, which is more attractive to application scientists and system designers. We set the value of θ (the threshold determines the ratio of active nodes) 90%. The comparison between traditional GPSR and k-OCHE protocols is reported in Figure 7, where x-axis is the initial energy of each sensor and y-axis is the value of network lifetime. From the figure, it can be easily seen that network lifetime in the traditional GPSR is about 1/6 of that in k-OCHE. Additionally, if we decrease the value of θ, the gap between the traditional GPSR and k-OCHE-based GPSR will become much wider. Thus we conclude that k-OCHE indeed extends network lifetime much more than that of traditional GPSR.

6.4. Recovery Time. Figure 8 shows the impact of the network size (node density) on the time to repair from an individual node failure for GPSR and k-OCHE, respectively. The x-axis is the network size and the y-axis is the value of recovery time. It is obvious that the recovery time of k-OCHE is much

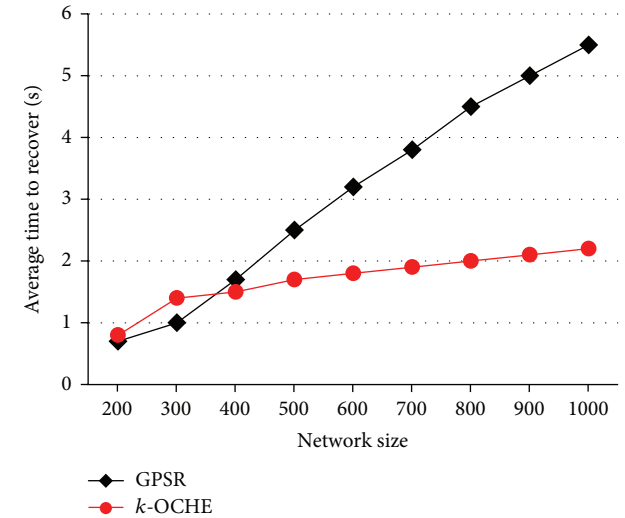
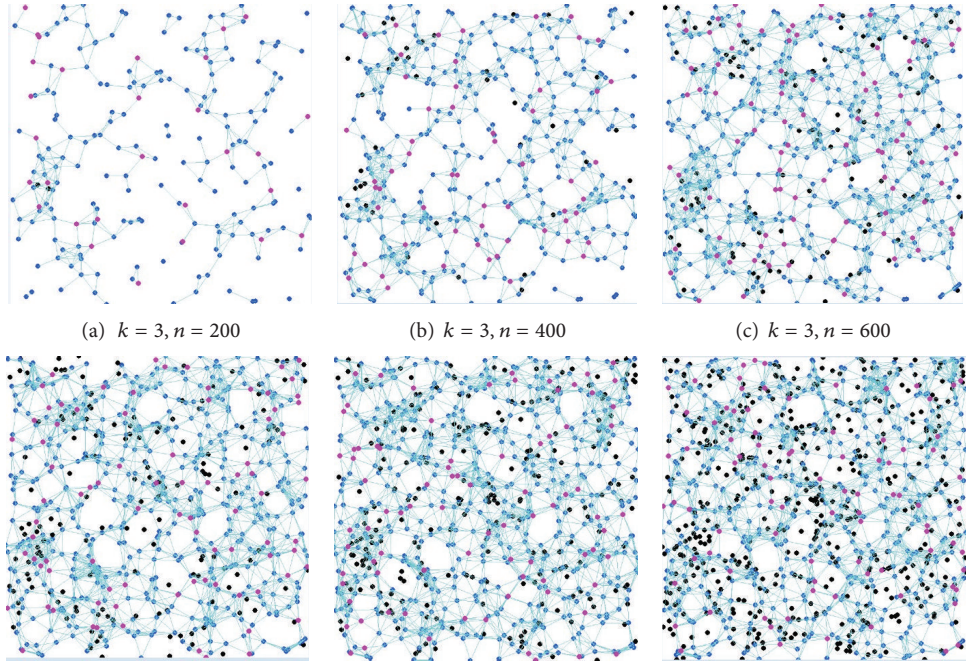


FIGURE 8: Comparison of routing recovery time by using traditional GPSR and k-OCHE-based GPSR.



(a) $k = 3, n = 200$ (b) $k = 3, n = 400$ (c) $k = 3, n = 600$

(d) $k = 3, n = 700$ (e) $k = 3, n = 800$ (f) $k = 3, n = 1000$

Figure 9: Demonstration of k-OCHE for different number of nodes n when $k = 3$.

more better than that of GPSR, especially at high values of network size. For low values of network size, that is network size <30, k-OCHE will consume a little bit more recovery time due to the maintenance of clusters. For high values of network size, the recovery time of GPSR increases with network size linearly, while the recovery time of k-OCHE increases very slowly.

7. Conclusion

In this paper, we propose a k-connected overlapping clustering approach with energy availability for routing and topology information maintenance in WSNs. We compare k-OCHE with the classical GPSR, and simulation results show that k-OCHE balances the load to extend the lifetime of sensor network. What is more, k-OCHE achieves shorter recovery time than GPSR, especially with large network size.

Appendix

We present the pseudo code of k-OCHE and review Connected K-Neighborhood (CKN) algorithm in [27] in this appendix.

Acknowledgment

This work is supported by the Natural Science Foundation of China under Grants no. 61070181, no. 61272524, and no. 61202442.

References

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

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

[3] D. Johnsort, "Routing in ad hoc networks of mobile hosts," in *Proceedings of the 1st Workshop on Mobile Computing Systems and Applications (WMCSA '94)*, pp. 158–163, IEEE, 1994.

[4] S. Soro and W. B. Heinzelman, "Prolonging the lifetime of wireless sensor networks via unequal clustering," in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, p. 8, IEEE, April 2005.

[5] J. H. Chang and L. Tassiulas, "Maximum lifetime routing in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, 2004.

[6] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers," *ACM SIGCOMM Computer Communication Review*, vol. 24, no. 4, pp. 234–244, 1994.

[7] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, ACM, August 2000.

[8] L. Shu, Y. Zhang, L. T. Yang, Y. Wang, M. Hauswirth, and N. Xiong, "TPGF: geographic routing in wireless multimedia sensor networks," *Telecommunication Systems*, vol. 44, no. 1-2, pp. 79–95, 2010.

[9] V. Rodoplu and T. H. Meng, "Minimum energy mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1333–1344, 1999.

[10] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the*

7th Annual International Conference on Mobile Computing and Networking, pp. 70–84, ACM, July 2001.

[11] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks," Tech. Rep., Citeseer, 2001.

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

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

[14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '00)*, p. 10, IEEE, January 2000.

[15] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings of the 4th International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN '99)*, pp. 310–315, IEEE, 1999.

[16] A. Manjeshwar and D. Agrawal, "Teen: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, vol. 22, 2001.

[17] S. Basagni, "Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks," in *Proceedings of the IEEE VTS 50th Vehicular Technology Conference (VTC '99)*, pp. 889–893, IEEE, September 1999.

[18] A. Manjeshwar and D. Agrawal, "Apteen: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," in *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, p. 48, 2002.

[19] M. Younis, M. Youssef, and K. Arisha, "Energy-aware routing in cluster-based sensor networks," in *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS '02)*, pp. 129–136, IEEE, 2002.

[20] M. Youssef, A. Youssef, and M. Younis, "Overlapping multihop clustering for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 12, pp. 1844–1856, 2009.

[21] S. Lindsey and C. Raghavendra, "Pegasis: power-efficient gathering in sensor information systems," in *Proceedings of the IEEE Aerospace Conference*, vol. 3, pp. 3–1125, IEEE, 2002.

[22] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 9, pp. 924–935, 2002.

[23] M. Youssef, M. Younis, and K. Arisha, "A constrained shortest-path energy-aware routing algorithm for wireless sensor networks," in *Proceedings of the Wireless Communications and Networking Conference (WCNC '02)*, vol. 2, pp. 794–799, IEEE, 2002.

[24] L. Subramanian and R. Katz, "An architecture for building selfconfigurable systems," in *Proceedings of the 1st Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHOC '00)*, pp. 63–73, IEEE, 2000.

[25] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: of theory and practice," in *Proceedings of the 22nd Annual ACM Symposium on Principles of Distributed Computing (PODC '03)*, pp. 63–72, ACM, July 2003.

[26] B. Leong, S. Mitra, and B. Liskov, "Path vector face routing: geographic routing with local face information," in *Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP '05)*, pp. 147–158, IEEE, November 2005.

[27] S. Nath and P. B. Gibbons, "Communicating via fireflies: geographic routing on duty-cycled sensors," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 440–449, April 2007.

[28] W. Heinzelman, *Application-specific protocol architectures for wireless networks [Ph.D. dissertation]*, Massachusetts Institute of Technology, 2000.

[29] L. Shu, C. Wu, Y. Zhang, J. Chen, L. Wang, and M. Hauswirth, "Nettopo: beyond simulator and visualizer for wireless sensor networks," in *Proceedings of the 2nd International Conference on Future Generation Communication and Networking (FGCN '08)*, vol. 1, pp. 17–20, IEEE, 2008.

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

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 174634, 12 pages
http://dx.doi.org/10.1155/2013/174634

Research Article

Amortized Fairness for Drive-Thru Internet

Zhi Li,[1,2] Limin Sun,[1] and Xinyun Zhou[1]

[1] *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Science, Beijing 100093, China*
[2] *Graduate University of Chinese Academy of Science, Beijing 100049, China*

Correspondence should be addressed to Limin Sun; sunlimin@iie.ac.cn

Received 10 January 2013; Accepted 20 February 2013

Academic Editor: Jianwei Niu

The drive-thru Internet is an effective mean to provide Internet access service for wireless sensor networks deployed on vehicles. In these networks, vehicles often experience different link qualities due to different relative positions to the access point. This makes fair and efficient system design a very challenging task. In traditional approaches, the network efficiency has to be greatly sacrificed to provide the fair share for vehicles with low link quality. To address this issue, we propose a novel amortized fairness MAC protocol. The basic idea is that vehicles with lower link quality can defer their fairness requests and let the lost fairness be "amortized" in the future when their links become the high quality. The amortized fairness MAC requires predictions of future link quality. For this, we fully exploit the inner and inter-AP correlations revealed from our extensive field studies and design a link quality prediction algorithm. Based on the predicted link quality, we formulate the optimal amortized fairness MAC as a convex programming problem, which can be solved with the desired precision in polynomial time. Extensive simulation on real traces shows that the amortized fairness MAC scheme is more efficient than the existing fairness schemes in terms of efficiency and fairness.

1. Introduction

Recent years, many wireless sensor networks (WSNs) have been deployed on vehicles [1, 2]. These systems require continuous access to the Internet for data collection. Cellular networks provide a universal access to the Internet, but its high cost and low throughput hinder their usage in reality. Recently, the vehicular *drive-thru Internet* networks [3, 4] are widely advocated to fulfill this need. Fixed wireless infrastructures such as the IEEE 802.11 access points (APs) are deployed along the roadside. WSNs nodes on vehicles access these APs occasionally when passing by. For example, CarTel [1] deploy IEEE 802.11b based sensor nodes to collect data as a car is driven. When the car enters the range of an AP, the data on cars is delivered to the Internet through the AP. In Beijing, many WiFi APs has been deployed in the downtown area, and the government recently initialed several projects to promote the Internet access for on-board wireless sensor networks.

Like in the traditional wireless local area networks (WLANs), in the drive-thru Internet the basic design goals are the throughput efficiency and fairness. We are in a dilemma when designing a fair and efficient drive-thru Internet, as it has been well known that these two objectives have an inherent trade-off. Vehicles may have different link qualities. Vehicles with a better link quality (e.g., higher signal noise ratio (SNR)) are more productive for the efficiency, while a fair AP access scheme has to allocate the transmission time to low quality vehicles to ensure the fairness, which surely will damage the throughput and efficiency.

In order to strike a better trade-off between efficiency and fairness, various fairness provisioning schemes have been proposed in WLANs (e.g., [5–7]) and in drive-thru Internet (e.g., [8, 9]). Most of these approaches provide instant fairness, in which only the present link qualities are exploited. In drive-thru Internet, vehicles typically experience highly dynamic link qualities, which enables more promising approaches. Rather than requesting the fairness immediately, a low quality vehicle may defer its requests on fairness and wait for a later time when it experiences high link qualities. In other words, its lost fairness is amortized to the future high link qualities, which may have a better price for the network efficiency. As a result, the impacts of low quality

links can be largely alleviated by this scheme, which we call *amortized fairness*.

The amortized fairness highly relies on accurate prediction on the future link qualities, so that low quality vehicles will have better link qualities and their lost fairness will be paid back. As people often drive through familiar routes [10], the same set of APs are encountered frequently. Among different passes of this set of APs, strong correlations between link qualities in an AP (called inner AP correlation) and between neighboring APs (called inter-AP correlation) are observed. Exploiting the inner and inter-AP correlations, we design a link quality prediction algorithm. Then, based on the predicted link quality, we propose an amortized fairness MAC protocol which can intelligently leverage high link qualities in the future to amortize the deferred requests of fairness. The main contributions of this paper are summarized as follows.

(1) We conduct extensive field studies on this drive-thru Internet and reveal the strong inner and inter-AP correlations of links between vehicles and APs.

(2) We design a link quality prediction algorithm for vehicles, whose prediction errors are proven to be bounded. Then, we formulate the optimal amortized fairness in vehicle drive-thru networks as a convex programming problem which can be solved in polynomial time.

(3) We carry out extensive simulations on real trace to evaluate the proposed amortized fairness MAC protocol. The results show that the amortized fairness MAC outperforms existing fairness schemes. In terms of system throughput, the amortized fairness improves the traditional throughput-based and speed-based fairness by up to 2.5 times and improves the time-based fairness by 40%.

The rest of this paper is structured as follows. Section 2 will introduce the motivation of the amortized fairness with a simple example. In Section 3, we study the wireless link characteristics in the drive-thru Internet to reveal the inner and inter-AP correlations. The proposed amortized fairness MAC protocol is present in Section 4, including the system framework, link quality estimation algorithm, and amortized fairness scheduling algorithm. We evaluate the amortized fairness protocol in Section 5, and overview the related works in Section 6. In the end, a simple conclusion will be drawn in Section 7.

2. Motivations

In this section, we introduce the drive-thru Internet considered in this paper and use a simple example to motivate the amortized fairness.

2.1. Drive-Thru Internet. In this paper, we consider the drive-thru Internet scenario in a one-way road with multiple lanes, as shown in Figure 1. Along the road, there are many WiFi APs deployed by inhabitants, network providers, governments, and so on. Some of them can be accessed by vehicles called open AP, while others called private AP are inaccessible for requiring password or unconnected to the Internet. Vehicles backlog their sensor data and intermittently connect to the Internet through open APs along the road. Usually, one AP may cover several vehicles simultaneously. Vehicles connecting to the same AP contend the transmission opportunities for uploading data.

Table 1: Summary of notations.

Symbols associated with link quality	
$x_{i,j}^{(p)}$	The average of link quality samples in the zone j when a vehicle passes the AP p ith time.
$\vec{x}_i^{(p)}$	The link quality vector of a vehicle passing the AP p at the ith time, written as $\vec{x}_i$ when no confusion, $\vec{x}_i^{(p)} = \langle x_{i,1}^{(p)}, x_{i,2}^{(p)}, \ldots, x_{i,K_p}^{(p)} \rangle$, where K_p is the number of zones.
$\overline{x}_i^{(p)}$	The mean of the link quality vector, $\overline{x}_i^{(p)} = (1/K_p) \sum_{k=1}^{K_p} x_i^{(p)}(k)$.
$\vec{X}^{(p)}$	The vector of link quality means of m passes, $\vec{X}^{(p)} = \langle \overline{x}_1^{(p)}, \ldots, \overline{x}_m^{(p)} \rangle$.
$\rho_{\text{inner}}^{(i,j)}$	The inner AP correlation coefficient of two link quality vectors $\vec{x}_i$ and $\vec{x}_j$.
$\rho_{\text{inter}}^{(p,q)}$	The inter-APs correlation coefficient of two neighboring AP p and q.
Symbols associated with throughput	
$s(u)$	Individual throughput of vehicle u, that is, the amount of data transferred during the pass of an open AP
$\vec{s}$	Individual throughput vector, $\vec{s} = \langle s(1), s(2), \ldots, s(n) \rangle$.

2.2. Motivation Example. Consider a simple scenario in which three vehicles u, v, and w are passing through an AP. Suppose because of their dynamic link qualities, vehicles experience different data rates at different times, as illustrated in Figure 2. For example, at $t = [0, 2]$, u can transmit at only 1 Mb/s, and the data rate improves to 5.5 Mb/s at $[2, 4]$. In the next, we give a detailed analysis on existing fairness provisioning protocols.

The throughput-based fairness, providing by original IEEE 802.11 DCF, ensures that every node has the same probability to transmit. By this scheme, at $[0, 1]$, both u and v can transmit 0.5 Mb data (i.e., $1\,\text{s} \times 1\,\text{Mb/s} \times 1/2 = 0.5\,\text{M}$). Similarly, we can compute the throughput of u, v, and w at other time slots and the results are depicted in Figure 2(a). The system throughput is 9.1 M. The time-based fairness [5, 6] gives each user u a fair share of transmission time regardless the individual link qualities. For example, at $[1, 2]$, both u and v share half of the airtime. So, u can transmit 0.5 M, and v can transmit 5.5 M in this second, as shown in Figure 2(b). The system throughput by time-based fairness is 20.5 M. The speed-based fairness [11] allocate, transmission probability based on the vehicle's speed. In this example, v and w have a doubled speed than u, so their transmission probabilities are doubled than u's. The individual throughput and the system throughput are shown in Figure 2(c).

Observing this example, we can find another more efficient and fair scheduling scheme. As depicted in Figure 2(d),

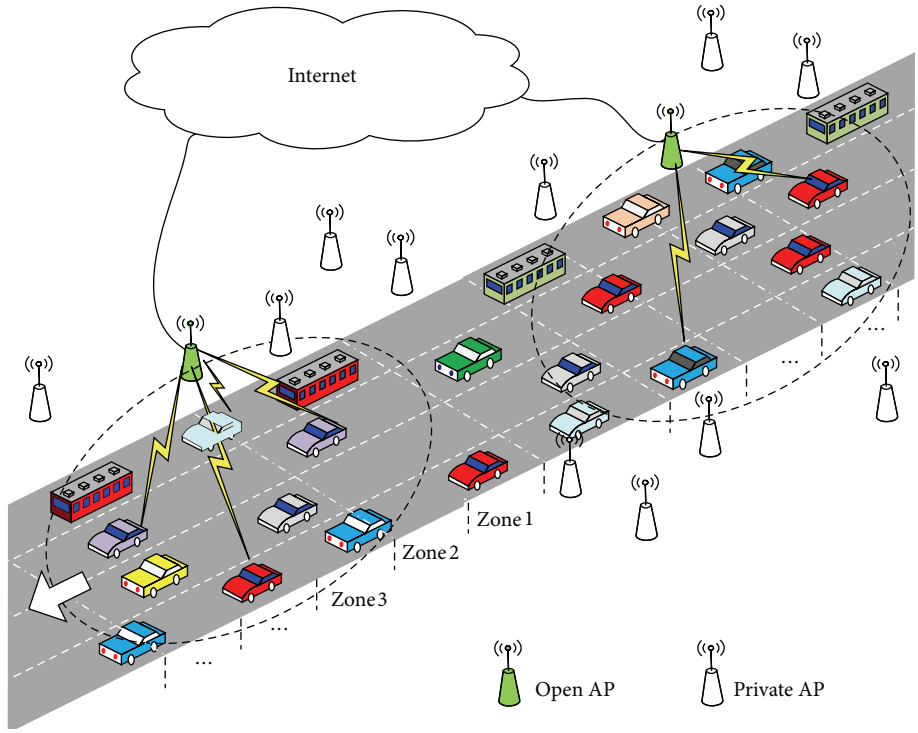


Figure 1: An illustration of vehicle drive-thru Internet.

	1	2	3	4	5	6	
u	0.5	11/12	11/13	11/13	11/12	0.5	4.5
v	0.5	11/12	11/13	0	0	0	2.3
w	0	0	0	11/13	11/12	0.5	2.3

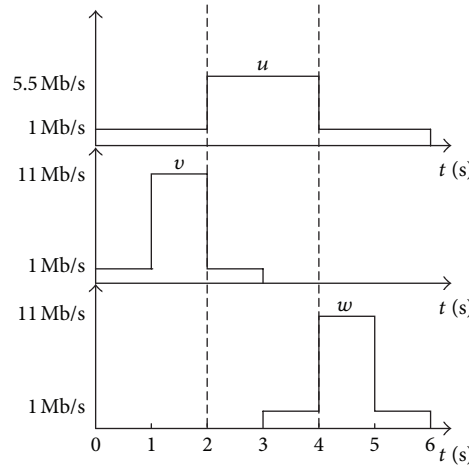
(a) Throughput-based 9.1 Mb

	1	2	3	4	5	6	
u	0.5	0.5	2.75	2.75	0.5	0.5	7.5
v	0.5	5.5	0.5	0	0	0	6.5
w	0	0	0	0.5	5.5	0.5	6.5

(b) Time-based 20.5 Mb

	1	2	3	4	5	6	
u	1/3	11/13	11/24	11/24	11/13	1/3	3.3
v	2/3	22/13	22/24	0	0	0	3.3
w	0	0	0	22/24	22/13	2/3	3.3

(c) Speed-based 9.9 Mb



	1	2	3	4	5	6	
u	0	0	5.5	5.5	0	0	11
v	1	11	0	0	0	0	12
w	0	0	0	0	11	1	12

(d) Amortized fairness 35 Mb

Figure 2: Motivation example.

Table 2: The mapping between link quality and the corresponding data rate.

Rate (Mb)	SNR (dB)
1	4+
2	8+
5.5	16+
11	21+

Inputs: $\vec{x}_i^{(q)}$, where $i \in [1, m]$, $q \in \Omega$
$\vec{x}_i^{(p)}$, where $i \in [1, m-1]$
Outputs: Estimated link quality vector $\hat{\vec{x}}_m^{(p)}$.
(1) **for all** $q \in \Omega \cup \{p\}$ **do**
(2) $\vec{X}^{(q)} = \langle \overline{x}_1^{(p)}, \overline{x}_2^{(p)}, \ldots, \overline{x}_{m-1}^{(p)} \rangle$
(3) **end for**
(4) $q = \arg\min_q(\rho_{\text{inter}}^{(p,q)}), q \in \Omega$
(5) $\sigma^{(p)} = \sqrt{\sum_{i=1}^{m-1}(\overline{x}_i^{(p)} - \overline{X}^{(p)})^2}$
(6) $\sigma^{(q)} = \sqrt{\sum_{i=1}^{m-1}(\overline{x}_i^{(q)} - \overline{X}^{(q)})^2}$
(7) $b = \rho_{\text{inter}}^{(p,q)} \cdot \sigma^{(p)}/\sigma^{(q)}$
(8) $a = \overline{X}^{(p)} - b \cdot \overline{X}^{(q)}$
(9) $\hat{\overline{x}}_m^{(p)} = b \cdot \overline{x}_m^{(q)} + a$
(10) $c = \hat{\overline{x}}_m^{(p)}$
(11) $\pi = \arg\min_\pi(|c, \overline{x}_\pi^{(p)}|), 1 \le \pi \le m-1$
(12) $\hat{\vec{x}}_m = \vec{x}_\pi + c - \overline{x}_\pi$

Algorithm 1: Estimation of link quality vector.

u gives up its fairness requests in the first two seconds and exclusively transmits in [3, 4], and again gives up in [5, 6]. Vehicles v and w have a similar strategy. By this scheduling scheme, the system throughput is 35 M, and each user obtains roughly 1/3 of the system throughput. The essence of this new allocation strategy is that vehicles defer their requests on the fairness and let the future better shares to amortize the current fairness loss. We call this new allocation strategy *amortized fairness*. The amortized fairness pays the fair shares with a better price and thus can effectively improve the network efficiency.

The amortized fairness highly relies on accurate predictions on link qualities so that users can justly calculate a best timing of their fair shares. In the next section, we will investigate the characteristics of links in drive-thru Internet which can help us to predict the link qualities.

3. Wireless Link Characteristics in Drive-Thru Internet

Accurate link quality predictions are crucial for the effectiveness of the amortized fairness. In this section, we investigate the wireless link characteristics in drive-thru Internet through empirical studies. We will show that link qualities present strong inner and inter-AP correlations, which can be greedily exploited for predictions.

In our experiments, we employ one programmable open AP and five vehicle nodes. The hardware for the AP and vehicle nodes are similar. They are made of a small embedded computer with an 1.6 GHz processor, 1 GB RAM, a magnetic RS232 GPS receiver for localization, and an Atheros-based CardBus 802.11 a/b/g wireless card. Linux with kernel 2.6.18 and Madwifi 0.9.4 are used to drive the wireless card. For the open AP, the wireless card works in AP mode and works in managed mode for vehicle nodes. Our experiments are carried at a segment of Zhongguancun Road in Beijing about 2 km length. Vehicle nodes are driven through the experiment segment and log the GPS and visible APs SNR at rates of 1 Hz and 5 Hz, respectively. We achieve this high AP scanning rate by programming vehicle nodes to only scan at channel 1. In total, we collect the data sets of 53 passes and discover 892 roadside APs (only one is ours, and others are deployed already).

3.1. Notations. Most notations used in this paper have been summarized in Table 1. Due to the error of commercial off-the-shelf GPS device (averaging to about 20 m [12]), it is difficult to accurately map an SNR sample to the location where this SNR was measured. So, we divide the coverage of a roadside AP into small zones, as shown in Figure 1.

Definition 1. Suppose that a vehicle passes a given AP p for the ith time, the vector of link qualities between the vehicle and the AP is defined as

$$\vec{x}_i^{(p)} = \left\langle x_{i,1}^{(p)}, x_{i,2}^{(p)}, \ldots, x_{i,K_p}^{(p)} \right\rangle, \quad (1)$$

where $x_{i,k}^{(p)}$, $k = 1, \ldots, K_p$ is the average of SNR samples from AP p in the zone k at the ith pass, and suppose that the coverage of AP p is divided into K_p zones. The AP id, p, may be ignored for the presentation simplicity when there is no confusion.

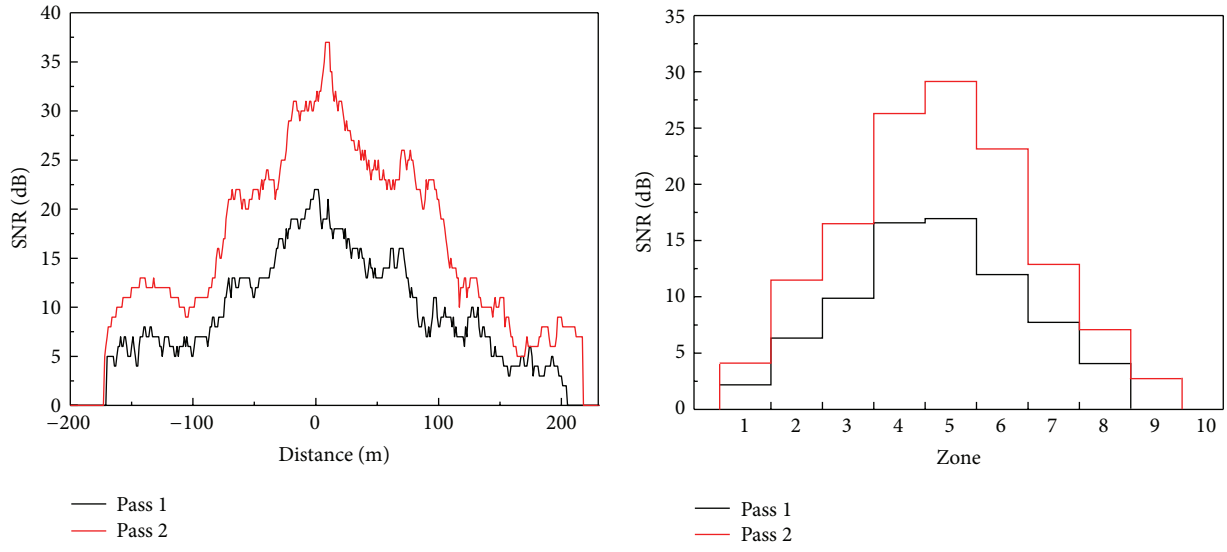
The mean of the link quality vector $\vec{x}_i^{(p)}$ can be calculated as follows:

$$\overline{x}_i^{(p)} = \frac{1}{K_p} \sum_{k=1}^{K_p} x_{i,k}^{(p)}. \quad (2)$$

Definition 2. Suppose that a vehicle has passed an AP p for m times, the vector of link quality means is defined as

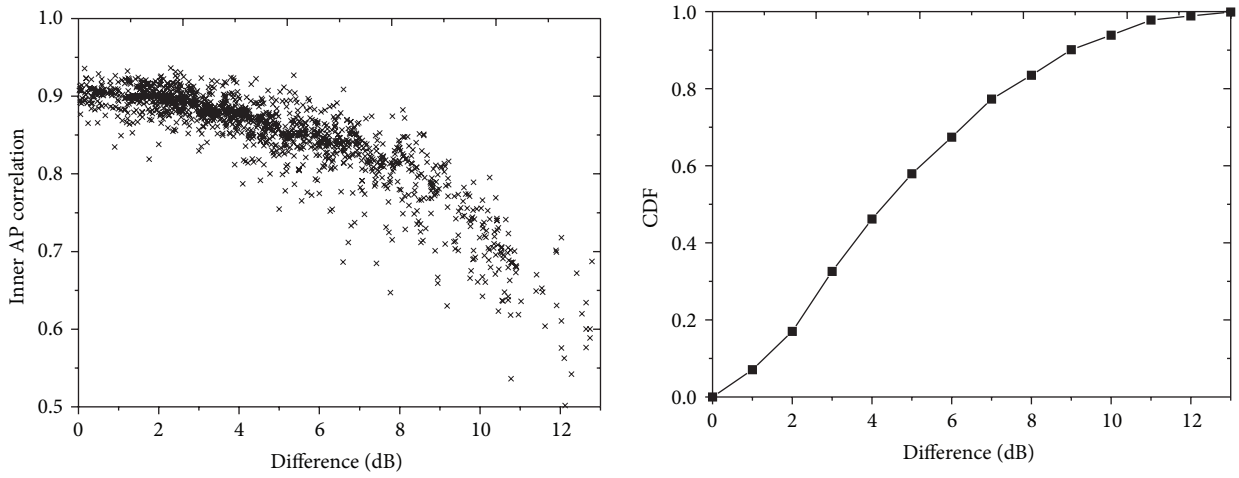
$$\vec{X}^{(p)} = \left\langle \overline{x}_1^{(p)}, \overline{x}_2^{(p)}, \ldots, \overline{x}_m^{(p)} \right\rangle. \quad (3)$$

3.2. Inner AP Correlation. We first investigate correlations among link qualities of different passes for a given vehicle and our open AP. Figure 3(a) shows link quality samples of two passes against the distance between the vehicle and the open AP. The x-axis is the distance, and the y-axis is the SNR in dB. We can observe that the shape of these two curves are very similar. Link quality vectors of these two passes are shown in Figure 3(b). To quantify the similarity, we define the *inner AP*



(a) Link quality samples (b) Link quality vectors

Figure 3: The SNR when a vehicle passes an AP twice.



(a) (b)

Figure 4: Inner AP correlation coefficients.

correlation coefficient of two link quality vectors $\vec{x}_i$ and $\vec{x}_j$ as follows:

$$\rho_{\text{inner}}^{(i,j)} = \frac{\sum_{k=1}^{K_p} \left(x_{i,k} - \overline{x}_i\right)\left(x_{j,k} - \overline{x}_j\right)}{\sqrt{\sum_{k=1}^{K_p} \left(x_{i,k} - \overline{x}_i\right)^2} \sqrt{\sum_{k=1}^{K_p} \left(x_{j,k} - \overline{x}_j\right)^2}}. \quad (4)$$

The inner AP correlation coefficient of these two link quality vectors shown in Figure 3 is up to 0.89.

To further explore characteristics of the inner AP correlation, we show the $\rho_{\text{inner}}^{(i,j)}$ of any two passes (i.e., pass i and j) in our 53 data set against the difference of link quality vector means (i.e., $|\overline{x}_i - \overline{x}_j|$) in Figure 4(a), and each dot represents one pair of passes. We can find that coefficients between different passes are pretty high when the difference of means is small, and a larger difference indicates a smaller coefficient. For example, when the difference is less than 2, the coefficients can be as high as 0.94 and the lowest one is 0.82. These results indicate that when appropriately scaled, link quality vectors of previous passes can be used to predict those of latter passes.

The CDF of any two link quality vector means difference is shown in Figure 4(b). We can find that the differences of link quality vector means are quite large. About half of the pass pairs have difference more than 5 dB. This conflict to existing measurement studies [3, 13], in which the link

qualities vary a little among different passes of the same AP. This is because their experiments were carried out in carefully planned and static environments, while our experiments were carried out in a real environment of city road where the vehicles might take different lanes and have different densities of neighboring vehicles at each pass. Although these dynamic factors change much at different passes, they tend to be stable when the vehicle passes by adjacent APs and have a similar impact on link qualities between the vehicle and these adjacent APs. This is the essential reason of the inter-AP correlation, which will be introduced in the next subsection.

3.3. Inter-AP Correlation. The link qualities are not only affected by the static factors, such as distance and the hardware of transceivers, but also by some dynamic factors such as the lanes vehicles take, the kinds and density of neighboring vehicles. These dynamic factors tend to be stable when a vehicle passes adjacent APs, and they incur similar attenuations of link qualities between the vehicle and those adjacent APs. In this part, we investigate the inter-AP correlations of link qualities. Suppose that a vehicle passes two geographically adjacent APs p and q for m times. So, we have two link quality mean vectors, that is, $\vec{X}^{(p)}$ and $\vec{X}^{(q)}$. The inter-AP correlation is defined as

$$\rho_{\text{inter}}^{(p,q)} = \frac{\sum_{i=1}^{m}\left(\overline{x}_i^{(p)} - \overline{X}^{(p)}\right)\left(\overline{x}_i^{(q)} - \overline{X}^{(q)}\right)}{\sqrt{\sum_{i=1}^{m}\left(\overline{x}_i^{(p)} - \overline{X}^{(p)}\right)^2}\sqrt{\sum_{i=1}^{m}\left(\overline{x}_i^{(q)} - \overline{X}^{(q)}\right)^2}}, \quad (5)$$

where the $\overline{X}^{(p)} = (1/m)\sum_{i=1}^{m}\overline{x}_i^{(p)}$, and the $\overline{X}^{(q)}$ is calculated similarly.

Although many roadside APs are found in our experiments, most of them have a few samples of SNR at each pass. It means that these APs are far away from the experiment road. As a result, the link qualities between a vehicle and these APs are impacted by many other dynamic factors. These APs are less useful for the analysis of inter-AP correlation. So we just select the 100 best APs for our analysis (i.e., closest to the road). For each AP we find the most correlated AP and compute the largest inter-AP correlation coefficient. We sort these 100 APs by their largest inter-AP correlation coefficient, and the result is shown in Figure 5. We can find that among all 100 APs, 34 APs have a inter-AP correlation coefficient over 0.9, and 78 APs have the coefficient over 0.8. Since we do not know the locations of all APs, It is difficult to show the relation between the inter-AP correlation and the distance between a pair of APs. We use the location of the largest SNR to approximate the AP's location, and find that not all pairs of nearby APs are high correlated. However, due to the large amount of APs along the road, it is probable to find a high correlated nearby AP for each open AP in practice.

4. Amortized Fairness Scheduling

In this section, we present the design of our amortized fairness scheduling protocol. First, we overview the system framework in brief. Then, we give detailed designs on the two entities involved in this protocol, that is, the vehicle part and the roadside AP part.

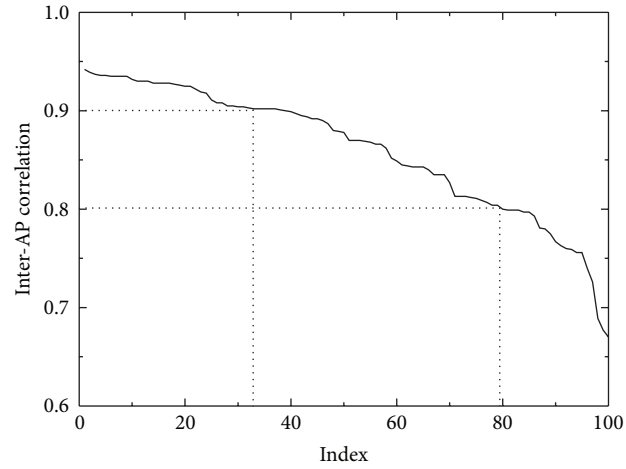


Figure 5: Inter-AP correlation.

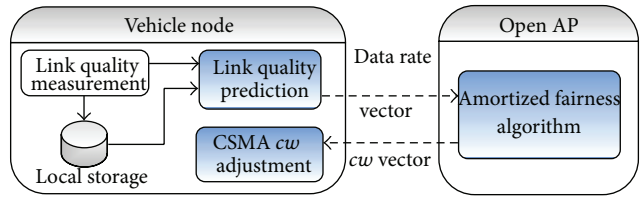


Figure 6: System framework of amortized fairness.

4.1. Overviews. The amortized fairness scheduling has a very simple system framework. Our design is on top of the CSMA/CA MAC protocols. Figure 6 draws the system framework of the amortized fairness MAC protocol. In general, it is realized by two entities. The vehicle nodes are responsible to collect the link quality measurements and GPS. These information will be stored in a local storage. Upon discovering a new open AP, vehicles retrieve the records and make an appropriate prediction for the future link quality vector within this AP. The vehicle will convert the link quality vector to a data rate vector according to an SNR-to-rate mapping (as shown in Table 2). This SNR-to-rate mapping is summarized from our field measurement experiments and is adopted by vehicle nodes for rate adaptation. Finally, the vehicle will deliver this data rate vector to the open AP.

Collecting the data rate vectors from all vehicles in the communication range, the AP will compute the optimal access strategy with the highest network efficiency and fairness by amortizing the low quality user's fair share to a better timing when necessary. The AP will then convert the optimal strategy to the corresponding minimum contention windows sizes (cw) in CSMA/CA and broadcast the cws to vehicles. The vehicles then make adequate adjustments on their minimum contention windows according to the cws.

4.2. Estimation of Link Quality Vector. In this section, we present the link quality vector estimation algorithm for

vehicles. To get an accurate estimation on link qualities, we exploit the inner and inter-AP correlations which we observed in Section 3. In a nutshell, we use the inter-AP correlation to estimate the mean of link quality vector and use the inner AP correlation to get the "shape" of the link quality vector. By intelligently integrating these two, we can get an accurate estimation of the link quality vector.

Suppose that this is a vehicle u's mth pass through an open AP p. In other words, there are $m-1$ history records, and we need to predict the mth link quality vector, that is, $\vec{x}_m^{(p)}$. Let Ω denote the set of p's preceding nearby APs. And for each AP in Ω, the vehicle u has passed it m times. Under this scenario, Algorithm 1 gives the pseudocode of the estimation algorithm. The algorithm inputs are the link qualities vectors of all APs in Ω and AP p. The output is estimated link quality vector $\hat{\vec{s}}_m^{(p)}$ for the open AP p. We first exploit the inter-AP correlation to estimate the mean of link quality vector $\overline{x}_m^{(p)}$ (from line 1 to line 9). For all AP in Ω and p, the vectors of link quality means of the past $m-1$ passes are computed from line 1 to line 3. In line 4, the inter-AP correlation coefficients between the AP p and each preceding AP are calculated, and we find out the most correlated preceding nearby AP, say the AP q. From line 5 to line 9, we adopt a simple linear model to predict the mean of link quality vector in the mth pass of the AP p, that is,

$$\hat{\overline{x}}_m^{(p)} = b \cdot \overline{x}_m^{(q)} + a, \tag{6}$$

where $b = \rho_{\text{inter}}^{(p,q)} \cdot \sigma^{(p)}/\sigma^{(q)}$ and $a = \overline{X}^{(p)} - b \cdot \overline{X}^{(q)}$ are two intermediate parameters for the linear prediction model.

Next, we explore the inner AP correlation to find the shape of the link quality vector. Line 11 finds the pass π from previous $m-1$ passes, the mean of whose link quality vector are closest to the predicted mean. We have observed in Section 3 that small difference on the means implies strong inner AP correlation, so the link quality vectors of the pass π and current pass m are strong correlated with high probability. The link quality vector of the pass m is predicted by adding the difference of means to the link quality vector of the pass π, that is,

$$\hat{x}_{m,k}^{(p)} = x_{\pi,k}^{(p)} + \hat{\overline{x}}_m^{(p)} - \overline{x}_\pi^{(p)}, \quad 1 \le k \le K_p. \tag{7}$$

For simplify, it can be rewrittne as

$$\hat{\vec{x}}_m = \vec{x}_\pi + \hat{\overline{x}}_m - \overline{x}_\pi. \tag{8}$$

We analyze the accuracy of the link quality prediction algorithm in the appendix, and make a comparison with an average-based prediction algorithm. The analysis results show that our proposed algorithm is much better than the average-based prediction algorithm. This is also validated in our simulations in Section 5.1.

4.3. Amortized Scheduling. In this subsection, we present the amortized scheduling algorithm for the drive-thru Internet. We first introduce the goal of the amortized fairness protocol (i.e., maximizing the proportional fairness), then we formulate the optimal amortized scheduling problem.

4.3.1. Proportional Fairness. For fairness, we adopt proportional fairness model that has been widely used in recent works [14]. Suppose that there are n vehicles passing an open AP, and let $s(u)$ denote the total data transmitted by the vehicle u (called *individual throughput*). All vehicles' individual throughput can be expressed as a vector $\vec{s}$

$$\vec{s} = \langle s(1), s(2), \ldots, s(n) \rangle. \tag{9}$$

An $\vec{s}$ is said proportionally fair if and only if for any other feasible solution $\vec{s}'$,

$$\sum_{u=1}^{N} \frac{s'(u) - s(u)}{s(u)} \le 0. \tag{10}$$

In other words, any change in the solution $\vec{s}$ must have a negative relative change. It has been proved in [14] that a proportionally fair allocation can be obtained by maximizing the system utility $J(\vec{s})$ over the set of feasible solutions,

$$J(\vec{s}) = \sum_{u=1}^{N} \ln(s(u)). \tag{11}$$

4.3.2. Scheduling Algorithm. The objective of amortized scheduling is to achieve proportional fairness for all vehicles in the coverage of an open AP, that is, $\max(J(\vec{s}))$. In other words, we need to schedule the transmission of each vehicle, so that their individual throughput maximizes the system utility. Suppose that each vehicle has infinite data to upload, and packets transmitted by all vehicles are equally sized. The individual throughput is determined by the minimum contention window size cw of the 802.11 DCF. So, the amortized scheduling problem is to determine the optimal cw for each vehicle.

Divide the time into L slots, and suppose that current time is the kth slot. Due to the short communication range of AP, we suppose that vehicles pass through the coverage of an open AP with constant speed. In addition, we assume that all vehicles' data rate keeps unchanged in each slot. Using $r_{u,j}$ and $\alpha_{u,j}$ to denote the data rate and the transmission time of the vehicle u at time slot j, respectively. The expected individual throughput $s(u)$ can be expressed as

$$\begin{aligned} s(u) &= \sum_{j=1}^{k-1} \alpha_{u,j} r_{u,j} + \sum_{j=k}^{L} \alpha_{u,j} \hat{r}_{u,j} \\ &= S_u(k) + \sum_{j=k}^{L} \alpha_{u,j} \hat{r}_{u,j}, \end{aligned} \tag{12}$$

where $S_u(k)$ is the amount of data that has already been transferred by vehicle u, and $\sum_{j=k}^{L} \alpha_{u,j} \hat{r}_{u,j}$ is the amount of data that will be transferred by this vehicle. In order to provide the proportional fairness, we need to maximize the system utility function. So, we formally define the optimal

amortized fairness scheduling problem as a convex programming problem as follows:

$$\begin{aligned} \max \quad & \sum_{u=1}^{N} \ln\left(S_u(k) + \sum_{j=k}^{L} \alpha_{u,j}\widehat{r}_{u,j}\right) \\ \text{subject to} \quad & \forall u, j \geq k, \quad 0 \leq \alpha_{u,j} \leq T \\ & \forall j \geq k, \quad \sum_{u=1}^{n} \alpha_{u,j} \leq T, \end{aligned} \tag{13}$$

where T is the length of a time slot. The first constraint says that the transmission time of any vehicle in any time slot cannot be longer than the time slot. The second one ensures that the sum of all the transmission time of vehicles in any time slot is no longer than the time slot.

Convex programming problem can be solved to the desired precision in polynomial time [15]. The fractional solution of $\alpha_{u,j}$ for each vehicle u in time slot j is an exact solution for transmission scheduling. In CSMA wireless networks, the transmission time is mainly controlled by the minimum contention window size cw. In order to grant the throughput according to the solution, we calculate the minimum contention window size for each vehicle according to its $\alpha_{u,j}$ and $\widehat{r}_{u,j}$. Let $cw_{u,j}$ be u's minimum window size at the time slot j. So, we have

$$cw_{1,j} : \cdots : cw_{n,j} = \frac{1}{\alpha_{1,j}\widehat{r}_{1,j}} : \cdots : \frac{1}{\alpha_{n,j}\widehat{r}_{n,j}} \tag{14}$$

for all $\alpha_{u,j}\widehat{r}_{u,j} > 0$.

The default minimum contention window size of the IEEE 802.11b is 32. For coexisting with other IEEE 802.11 protocol, we set the average of all vehicles' minimum contention window size also to be 32. As a result, we have

$$cw_{u,j} = \frac{32n_j}{\alpha_{u,j}\widehat{r}_{u,j}}\left(\sum_{u}^{n_j} \frac{1}{\alpha_{u,j}\widehat{r}_{u,j}}\right)^{-1}, \tag{15}$$

where n_j is the number of vehicles satisfying $\widehat{\alpha}_{u,j}\widehat{r}_{u,j} > 0$. For the vehicle whose $\widehat{\alpha}_{u,j}\widehat{r}_{u,j}$ is zero, the $cw_{u,j}$ is set to be a default large value.

5. Performance Evaluation

We have implemented a simulator to simulate the drive-thru Internet scenario with 1 open AP and 20 vehicles. In order to emulate the link qualities when a vehicle drives through the open AP, for each vehicle we randomly choose link quality traces of m passes from the data set collected in Section 3, where m is a control parameter. Suppose that vehicles have passed the open AP $m-1$ times and are going to pass the AP for the mth time. To emulate the mobility of vehicles, we assign a speed and entering time of AP for each vehicle to simulate its mth pass of this AP. The speed is randomly chosen among $[v_{\min}, v_{\max}]$, where $v_{\min}$ and $v_{\max}$ are both parameters. We suppose that vehicles enter the open AP according to the Poisson process with a parameter λ.

We compare the proposed amortized fairness MAC protocol with three existing fairness provisioning schemes, that is, throughput-based fairness, time-based fairness, and speed-based fairness. The former two are widely studied in the WLAN, and the last one is a recent work in the drive-thru Internet.

(i) Throughput-based fairness is naturally provided by the current IEEE 802.11 DCF protocol. It assigns each node, regardless the link quality of nodes, the same probability to access the AP. As a result, different nodes are likely to transmit the same amount of packets in average.

(ii) Time-based fairness grants each node the same amount of the transmission time rather than the probability of the transmission. In this scheme, high quality nodes can transmit more data.

(iii) Speed-based fairness [9] assigns the transmission probability based on the user's speed. A faster vehicle will be granted with high transmission probability because of its shorter resident time, and vice versa for slower vehicles.

5.1. Accuracy of Link Quality Prediction. We first evaluate the accuracy of the link quality prediction algorithm. It is affected by the amount of link quality vectors which have been recorded (i.e., $m-1$). We vary the parameter m from 2 to 20 and adopt the average error to quantify the prediction accuracy which is defined as

$$\sqrt{\frac{\sum_{k=1}^{K_p}\left(\widehat{x}_{m,k}^{(p)} - x_{m,k}^{(p)}\right)^2}{K_p}}. \tag{16}$$

For each m, the simulation is run 100 times and the mean of the average error is depicted in Figure 7. The 90% confidence interval is depicted by an error bar in the figure. When $m = 2$ (i.e., there is only one past pass available for prediction), the inter-AP correlation coefficient cannot be calculated. In this case, we use the static prediction scheme as $\widehat{\vec{x}}_2^{(p)} = \vec{x}_1^{(p)}$.

From Figure 7, we observe that when m is small the prediction error is increasing with the m. This is because that with small m ($m = 3$ and $m = 4$) the inner-AP correlation of sample set may severely deviate from its true value, incurring large prediction errors. As m continuously increases, the errors begin to reduce with better stability. With more than 15 history records, the errors are stabilized to about 0.28. Fewer further improvements are observed with more records. As there is a direct map from the link quality to the corresponding data rate, and the SNR-based data rate adaption techniques are quite mature, we believe the obtained data rate vector will have a similar accuracy.

5.2. Efficiency and Fairness Evaluations. We compare the amortized fairness scheduling protocol with the three existing fairness provisioning schemes, the throughput-based fairness, time-based fairness, and speed-based fairness. Figure 8 shows the individual throughput of each vehicle by

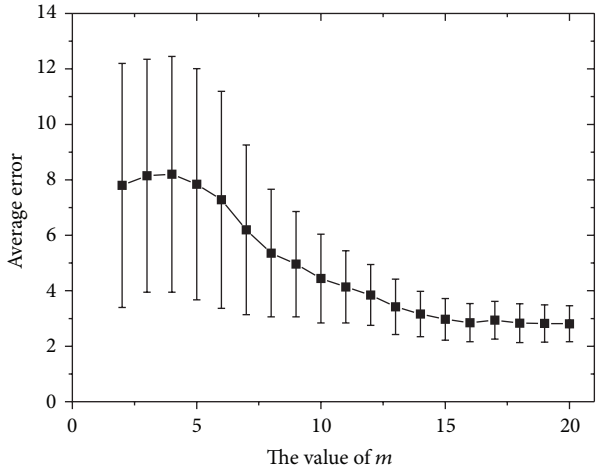


Figure 7: Average error under different m.

these four fairness schemes under different arrival rate and speed of vehicles. The vehicles are sorted by their individual throughput in increasing order. The x-axis is the index of vehicles, and the y-axis is the obtained individual throughput $s(u)$. In general, amortized fairness outperforms all the others in all scenarios.

For the case of the middle vehicle arrival rate $\lambda = 0.5$ and middle vehicle speed $v_{\max} = 20\,\text{m/s} = 72\,\text{km/h}$ (Figure 8(a)), throughput- and speed-based fairness perform similarly. Time-based fairness has about 100% improvements than them, and the amortized fairness has about 100% further improvements. For example, the 10th individual throughput in these four schemes are 1.9 MB, 1.9 MB, 3.7 MB, and 5.5 MB, respectively. When vehicles have the same speed to pass through the APs (Figure 8(b)), speed-based fairness will become the traditional throughput-based fairness. They have the similar performance.

When we reduce the vehicle arrival rate to $\lambda = 0.1$ and increase the range of speed $v_{\min} = 5$ and $v_{\max} = 30$ (Figure 8(c)), the difference between vehicles' residence times increases and the system is under an under-utilized setting. In that scenario, throughput fairness cannot be aware of the huge difference between vehicles' throughput and becomes fairly unfair. The maximal individual throughput is 48 MB, and the minimal is only 1 MB. Other two existing algorithms are similar. To the contrast, amortized fairness can intelligently exploit the link dynamics and achieve a better fairness and efficiency. We further increase the user arrival rate to 1 (Figure 8(d)). In that case, more users contend the AP simultaneously. So, the traffic of users are lower than the other three cases. Again, we observe the significant improvement of amortized fairness compared with others.

Figure 9 shows the average system throughput of these four fairness provisioning schemes under four previous scenarios. In all cases, amortized fairness exhibits significant outperformance. In the scenarios of A, B, and D, amortized fairness improves the total throughput by about 2.5 times compared with throughput- and speed-based fairness and improves over 40% compared with time-based fairness. In the case C, the improvements are 90% and 30%, respectively.

6. Related Work

Many works have demonstrated the feasibility of IEEE 802.11 AP-based drive-thru Internet [3, 4]. The authors of [3, 13] have extensively measured the link quality between the vehicle and the roadside AP. They suggest that the link quality varies when a vehicle passes an AP, and the link quality becomes better in entering phase, while becomes worse in leaving phase. Our experiments get a consistent result about the variance course of link quality. In addition, by comparing a vehicle's link qualities at different passes of an AP, we found that the link qualities of two passes over a same AP are correlated (called inner AP correlation), and the mean of link quality when a vehicle passed some adjacent APs is also correlated (called inter-AP correlation).

For the variance of link qualities in drive-thru Internet, when multiple users share an AP simultaneously, it will lead to a dilemma problem of trade-off between efficiency and fairness. The original IEEE 802.11 DCF achieves the same throughput for nodes with different link qualities. It is notorious for the performance anomaly, because it damages the throughput of high quality users severely. Time-based fairness schemes [5, 6] are proposed to alleviate this anomaly by assigning equal transmission time to each node.

In the drive-thru Internet system, Hadaller et al. [13] first consider performance anomaly in drive-thru Internet and propose a greedy algorithm where only nodes with the best SNR are allowed to transmit. This simple scheme can achieve the maximum system throughput, but it incurs a poor fairness. Luan et al. [8] develop an accurate model to investigate the performance of IEEE 802.11 DCF in the drive-thru Internet. By knowing the node mobility and the link rate previously, they configure the minimum contention window size to maximize the system throughput while guaranteeing a certain lower bound of individual throughput for each vehicle. However, the link rate is very difficult to predict due to high environment dynamic. So, the performance will fail to meet their promise in practice.

The diversity of vehicle speed also leads to fairness problem in the drive-thru Internet. because vehicles with different speeds have their different limited time to communicate with AP. The author in [9] proposed MAC scheme to change the minimum contention window size according to vehicles' speed, so that fast vehicles can transfer the same amount of data as slow vehicles. However it has a low efficiency for the performance anomaly. Furthermore, this scheme also supposes that the link rate is known previously.

In this paper, we aim to an efficient and fair drive-thru MAC scheme, namely, we need to handle the performance anomaly of IEEE 802.11 DCF as well as the diversity of vehicle speed. Contrary to existing theoretical works in the drive-thru networks [8, 9], we exploit link correlations to accurately predict the link rate, instead of assuming that the link rate is known previously. Furthermore, rather than requesting the fairness immediately as all above works do, our amortized

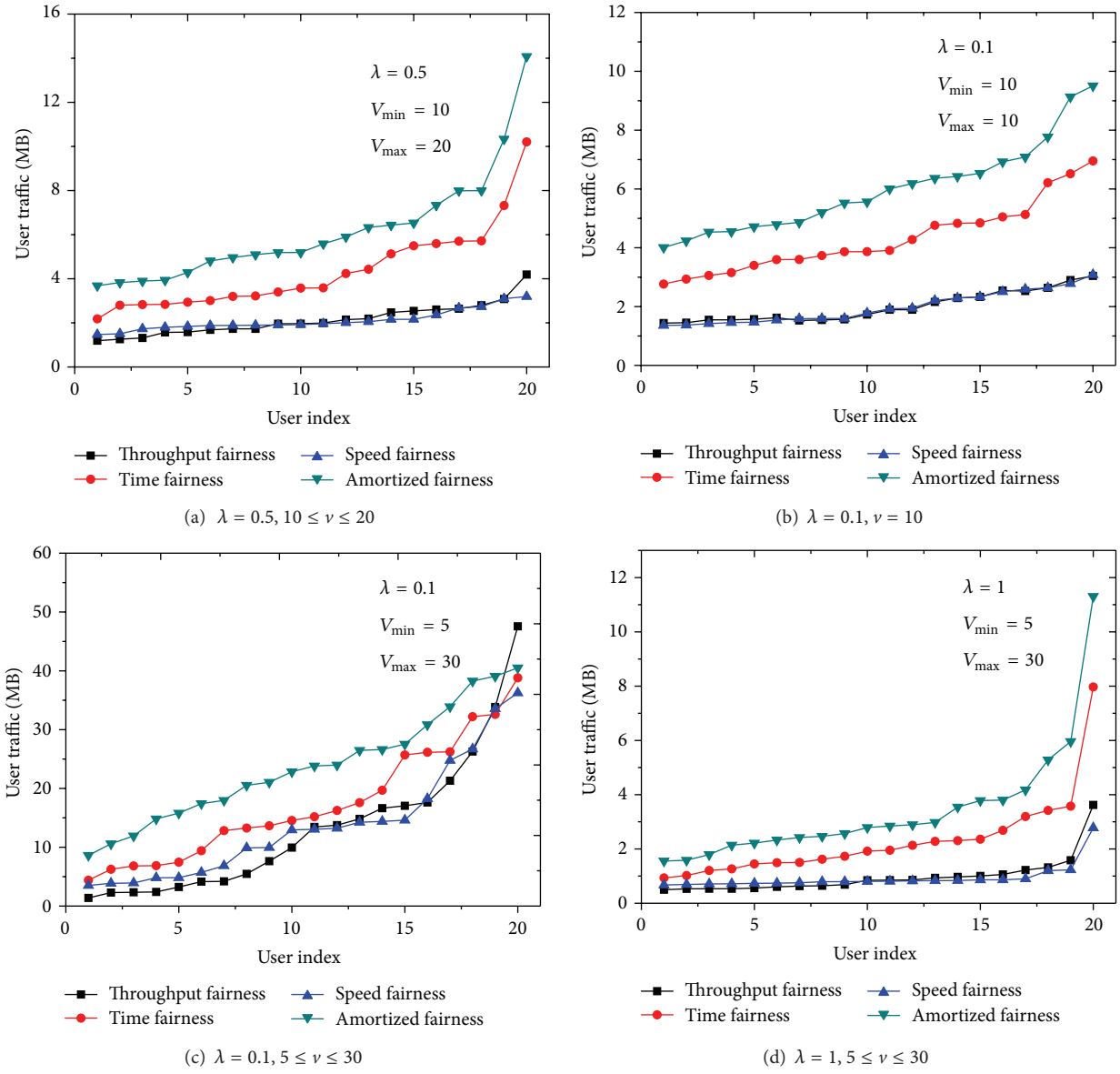


Figure 8: Individual throughput comparison.

fairness protocol defers the fairness requests of low quality vehicles and amortized the loss of fairness over future high link qualities. So, the impacts of low quality links can be largely alleviated and the fairness is guaranteed as well.

7. Conclusion

In this paper, we study the fairness and efficiency issues in the drive-thru Internet networks. Different from the traditional fairness provisioning approaches, in this paper, we propose a novel amortized fairness scheduling protocol, which takes the future opportunity as an advantage. It allows low quality vehicles to defer their fairness requests and claim them back at a better timing when their link become high quality. To exploit such future opportunities well, we investigate the inner and inter-AP correlations between wireless links through extensive field studies. We design a link quality prediction algorithm and an amortized fairness scheduling algorithm. The prediction algorithm is proven to have a bounded performance. We conduct trace-driven simulations for performance evaluations and the results demonstrate supreme performance gains against existing methods in all simulation scenarios.

Appendix

Analysis of the Link Quality Estimation

In this section, we give analysis on the prediction error (measured by the mean-square error (MSE)) of the mean of link quality vector $\overline{x}_m^{(p)}$ and the link quality vector $\vec{x}_m^{(p)}$. We will show that the prediction error of our proposed algorithm

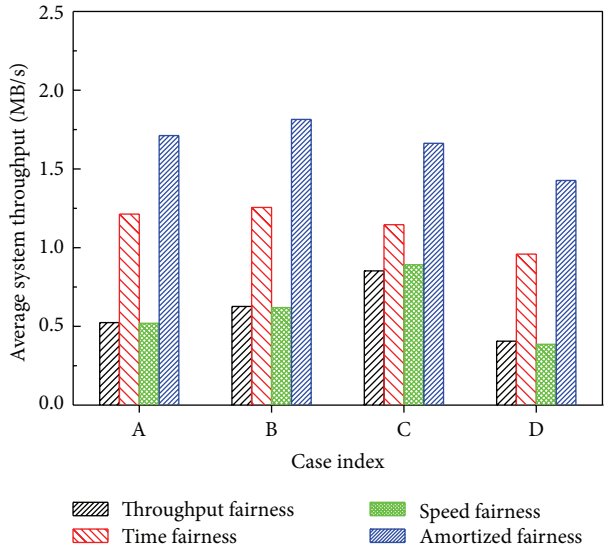


FIGURE 9: The average system throughput of different schemes under four scenarios.

is bounded and much small than average-based prediction scheme. In the average-based prediction scheme, the link qualities are predicted as

$$x_{m,k}^{(p)} = \frac{1}{m-1}\sum_{i=1}^{m-1} x_{i,k}^{(p)}, \quad 1 \le k \le K_p. \tag{A.1}$$

Given a random variable Y, the MSE is defined as

$$\mathrm{MSE}(Y) = E\left[\left(Y - \widehat{Y}\right)^2\right], \tag{A.2}$$

where E is to calculate the expectation, and $\widehat{Y}$ is the prediction value of the random variable Y.

Let P and Q denote the random variables of the link quality vector means when a vehicle passed the AP p and q, respectively. So, the $\overline{x}_m^{(p)}$ is the mth sample of P, and the prediction (6) can be rewritten as $\widehat{P} = b \cdot Q + a$.

Lemma A.1. *The MSE of the link quality vector mean P is*

$$\begin{aligned} MSE(P) = {} & \sigma_p^2\left(1 - \rho_{inter}^2\right) + \sigma_q^2\left(b - \rho_{inter}\sigma_p\sigma_q^{-1}\right)^2 \\ & + \left(\mu_p - b\mu_q - a\right)^2, \end{aligned} \tag{A.3}$$

where $\sigma_p = \sqrt{D(P)}$, $\sigma_q = \sqrt{D(Q)}$, $\mu_p = E(P)$, $\mu_q = E(Q)$, $\rho_{inter} = \rho(P,Q)$. *The operator D is to calculate the square deviation, and operator ρ is to calculate the correlation coefficient.*

Proof. According to the definition, the mean square-error of the prediction is

$$\mathrm{MSE}(P) = E\left[\left(P - \widehat{P}\right)^2\right] \tag{A.4}$$

$$= E\left[(P - (b \cdot Q + a))^2\right]. \tag{A.5}$$

Notice that by the probability theory, we have

$$\begin{aligned} E\left(P^2\right) &= D(P) + (E(P))^2, \\ E\left(Q^2\right) &= D(Q) + (E(Q))^2, \\ E(PQ) &= \rho(P,Q)\sqrt{D(P)}\sqrt{D(Q)} + E(P)E(Q). \end{aligned} \tag{A.6}$$

Expanding the square in (A.5), we obtain

$$\begin{aligned} \mathrm{MSE}(P) = {} & D(P) \\ & + b^2 D(Q) - 2b\rho(P,Q)\sqrt{D(P)}\sqrt{D(Q)} \\ & + (E(P))^2 - 2bE(P)E(Q) + b^2(E(Q))^2 \\ & - 2aE(P) + 2abE(Q) + a^2 \\ = {} & \sigma_p^2\left(1 - \rho_{\mathrm{inter}}^2\right) + \sigma_q^2\left(b - \rho_{\mathrm{inter}}\sigma_p\sigma_q^{-1}\right)^2 \\ & + \left(\mu_p - b\mu_q - a\right)^2. \end{aligned} \tag{A.7}$$

□

Lemma A.2. *When m is becoming infinite, we have*

$$\lim_{m\to\infty} MSE(P) = \sigma_p^2\left(1 - \rho_{inter}^2\right). \tag{A.8}$$

Proof. Notice that $\vec{X}^{(p)}$ and $\vec{X}^{(q)}$ are two sample sets of the random variables P and Q. According to the large number theorem in probability theory, we have

$$\begin{aligned} \lim_{m\to\infty}\sigma^{(p)} &= \sqrt{D(P)} = \sigma_p, \\ \lim_{m\to\infty}\sigma^{(q)} &= \sqrt{D(Q)} = \sigma_q, \\ \lim_{m\to\infty}\overline{X}^{(p)} &= E(P) = \mu_p, \\ \lim_{m\to\infty}\overline{X}^{(q)} &= E(Q) = \mu_q, \\ \lim_{m\to\infty}\rho_{\mathrm{inter}}^{(p,q)} &= \rho(P,Q) = \rho_{\mathrm{inter}}, \end{aligned} \tag{A.9}$$

and thus

$$\begin{aligned} \lim_{m\to\infty}\left(b - \rho_{\mathrm{inter}}\sigma_p\sigma_q^{-1}\right) &= \lim_{m\to\infty}\left(\rho\left(\vec{X}^{(p)}, \vec{X}^{(q)}\right)\frac{\sigma^{(p)}}{\sigma^{(q)}}\right) \\ & \quad - \rho_{\mathrm{inter}}\sigma_p\sigma_q^{-1} \\ &= 0, \\ \lim_{m\to\infty}\left(\mu_p - b\mu_q - a\right) &= \mu_p - b\mu_q - \lim_{m\to\infty}\left(\overline{X}^{(p)} - b\overline{X}^{(q)}\right) \\ &= 0. \end{aligned} \tag{A.10}$$

Therefore, $\lim_{m\to\infty}\mathrm{MSE}(P) = \sigma_p^2(1 - \rho_{\mathrm{inter}}^2)$. □

In the average-based prediction algorithm, the error of the link qualities mean is σ_p^2. It is larger than $\sigma_p^2(1-\rho_{\text{inter}}^2)$ tremendously when ρ_{inter} is close to 1. Due to numerous roadside APs, it is probable to find a large ρ_{inter} for any open AP in practice.

Let M and Π be random variables of the link qualities scanning in the same zone when the vehicle passes the open AP p for the mth and πth time, respectively. Elements in vector $\vec{x}_m^{(p)}$ and $\vec{x}_\pi^{(p)}$ are samples of M and Π, so the prediction (8) of $\vec{x}_\pi^{(p)}$ can be presented as $\widehat{M} = \Pi + \hat{\overline{x}}_m - \overline{x}_\pi$.

Theorem A.3. *The MSE of the predicted link quality M is*

$$MSE(M) = \sigma_m^2 - 2\rho_{inner}\sigma_m\sigma_\pi + \sigma_\pi^2 + \left(\mu_m - \mu_\pi - \hat{\overline{x}}_m + \overline{x}_\pi\right)^2, \tag{A.11}$$

where $\sigma_m = \sqrt{D(M)}$, $\sigma_\pi = \sqrt{D(\Pi)}$, $\mu_m = E(M)$, $\mu_\pi = E(\Pi)$, and $\rho_{inner} = \rho(M,\pi)$.

Proof. According to the definition, the mean square-error of the predicted is

$$\begin{aligned}\text{MSE}(M) &= E\left[\left(M - \left(\Pi + \hat{\overline{x}}_m - \overline{x}_\pi\right)\right)^2\right] \\ &= D\left[\left(M - \Pi - \hat{\overline{x}}_m + \overline{x}_\pi\right)^2\right] \\ &\quad + E^2\left[\left(M - \Pi - \hat{\overline{x}}_m + \overline{x}_\pi\right)\right] \\ &= \sigma_m^2 - 2\rho_{\text{inner}}\sigma_m\sigma_\pi + \sigma_\pi^2 \\ &\quad + \left(\mu_m - \mu_\pi - \hat{\overline{x}}_m + \overline{x}_\pi\right)^2.\end{aligned} \tag{A.12}$$

□

Because $\vec{x}_\pi^{(p)}$ is the sample set of Π by probability theory; we have $\mu_\pi = \vec{\overline{x}}_\pi$ when the amount of element in $\vec{x}_\pi^{(p)}$ becomes infinite. In this case, we have

$$\begin{aligned}\text{MSE}(M) &= \sigma_m^2 - 2\rho_{\text{inner}}\sigma_m\sigma_\pi + \sigma_\pi^2 + \left(\mu_m - \hat{\overline{x}}_m\right)^2 \\ &= \sigma_m^2 - 2\rho_{\text{inner}}\sigma_m\sigma_\pi + \sigma_\pi^2 + \text{MSE}(P).\end{aligned} \tag{A.13}$$

Notice that the $\vec{x}_\pi^{(p)}$ has the closest mean to the $\hat{\overline{x}}_m^{(p)}$. It implies that the ρ_{inner} is close to 1 as well as σ_m^2 and σ_π have similar values. So, the $\sigma_m^2 - 2\rho_{\text{inner}}\sigma_m\sigma_\pi$ is very small usully.

Acknowledgments

This work is supported by the State Key Program of National Natural Science of China (Grant no. 60933011) and the State Key Development Program for Basic Research of China (Grant no. 2011CB302902).

References

[1] B. Hull, V. Bychkovsky, Y. Zhang et al., "Cartel: a distributed mobile sensor computing system," in *Proceedings of the 4th international Conference on Embedded Networked Sensor Systems*, pp. 125–138, ACM, 2006.

[2] J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan, "The Pothole Patrol: using a mobile sensor network for road surface monitoring," in *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys '08)*, pp. 29–39, ACM, June 2008.

[3] J. Ott and D. Kutscher, "Drive-thru internet: IEEE 802.lib for "automobile" users," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, pp. 362–373, March 2004.

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

[5] G. Tan and J. Guttag, "Time-based fairness improves performance in multi-rate wlans," in *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, pp. 23–23, USENIX Association, 2004.

[6] M. Heusse, F. Rousseau, R. Guillier, and A. Duda, "Idle sense: an optimal access method for high throughput and fairness in rate diverse wireless lans," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 121–132, 2005.

[7] L. B. Jiang and S. C. Liew, "Proportional fairness in wireless LANs and ad hoc networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '05)*, vol. 3, pp. 1551–1556, March 2005.

[8] T. Luan, X. Ling, and X. Shen, "Mac in motion: impact of mobility on the mac of drive-thru internet," *IEEE Transactions on Mobile Computing*, vol. 11, no. 2, pp. 305–319, 2012.

[9] E. Karamad and F. Ashtiani, "A modified 802.11-based MAC scheme to assure fair access for vehicle-to-roadside communications," *Computer Communications*, vol. 31, no. 12, pp. 2898–2906, 2008.

[10] P. Deshpande, A. Kashyap, C. Sung, and S. R. Das, "Predictive methods for improved vehicular WiFi access," in *Proceedings of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys '09)*, pp. 263–276, June 2009.

[11] L. Xie, Q. Li, W. Mao, J. Wu, and D. Chen, "Achieving efficiency and fairness for association control in vehicular networks," in *Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP '09)*, pp. 324–333, October 2009.

[12] M. Matosevic, Z. Salcic, and S. Berber, "A comparison of accuracy using a GPS and a low-cost DGPS," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 5, pp. 1677–1683, 2006.

[13] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal, "Vehicular opportunistic communication under the microscope," in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys '07)*, pp. 206–219, ACM, June 2007.

[14] T. Nandagopal, T. E. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 87–98, August 2000.

[15] D. Bertsekas, A. Nedi, and A. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 192675, 9 pages
http://dx.doi.org/10.1155/2013/192675

Research Article

Weight-Based Clustering Decision Fusion Algorithm for Distributed Target Detection in Wireless Sensor Networks

Haiping Huang,[1,2,3] Lei Chen,[1,2] Xiao Cao,[2,4] Ruchuan Wang,[1,2,3,5] and Qianyi Wang[1]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*
[3] *Jiangsu Computer Information Processing Technology Key Laboratory, Suzhou University, Suzhou 215006, China*
[4] *International Business Machine (IBM), Global Business Services, Nanjing 210005, China*
[5] *Key Lab of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, China*

Correspondence should be addressed to Haiping Huang; hhp@njupt.edu.cn

Received 18 December 2012; Revised 17 February 2013; Accepted 21 February 2013

Academic Editor: Jianwei Niu

We use a great deal of wireless sensor nodes to detect target signal that is more accurate than the traditional single radar detection method. Each local sensor detects the target signal in the region of interests and collects relevant data, and then it sends the respective data to the data fusion center (DFC) for aggregation processing and judgment making whether the target signal exists or not. However, the current judgment fusion rules such as Counting Rule (CR) and Clustering-Counting Rule (C-CR) have the characteristics on high energy consumption and low detection precision. Consequently, this paper proposes a novel Weight-based Clustering Decision Fusion Algorithm (W-CDFA) to detect target signal in wireless sensor network. It first introduces the clustering method based on tree structure to establish the precursor-successor relationships among the clusters in the region of interests and then fuses the decision data along the direction from the precursor clusters to the successor clusters gradually, and DFC (i.e., tree root) makes final determination by overall judgment values from subclusters and ordinary nodes. Simulation experiments show that the fusion rule can obtain more satisfactory system level performance at the environment of low signal to noise compared with CR and C-CR methods.

1. Introduction

As we know, we can only use one sensor or radar to detect target signal of one event source in ideal environment. In normal environment, we cannot do it well because of the affection of noise around, which would lead to false detection result. Instead, we can lay many sensors in ROI (Region of Interest), and they cooperate with each other to make final judgment for improving detection accuracy. Actually, wireless sensor network composed of large-scale distributed nodes would resolve the problem of target detection more effectively by determining whether there exists target signal based on sensor nodes' cooperation, and this is one of the most important applications on monitoring objective physical world. In ideal circumstance, how to estimate whether there exists target in ROI can be simplified to judge whether there exists target signal or not. However, practical environment is blended with a lot of random noises, and therefore, how to increase the target detection precision and reduce energy consumption in noisy sensor network environment will be the focus of this paper.

Many scholars have studied distributed target detection technique. Tenney and Sandell apply Bayesian theory to target signal detection [1]. In [2], Chair and Varshney point out that authors do not consider the data fusion among multisensors in [1], and then they set the corresponding weight value for each prior probability of detection on the

basis of Zero-One decision fusion. In [3, 4], considering the criteria of Chair-Varshney, authors excogitate a kind of data fusion rule based on the correlation coefficient.

Niu and Varshney publish an algorithm based on Zero-One decision fusion-Counting Rule [5] in ICASS'05. They set the detection probability and false alarm probability of each sensor node to be the same. And thereby, Chair-Varshney criteria can be simplified to be the summation of judgment values (0 or 1) of all nodes. According to central limit theorem, Niu calculates the value of theory approximation of the detection performance. Katenka et al. put forward the fusion algorithm based on the Zero-One decision [6] of local vote. He takes full advantage of the collaboration features of neighbor nodes in wireless sensor network. Each node amends its judgment according to the judgments made by its neighbor nodes. In [7], Sung et al. put forward a new evaluation guide to describe the performance of target detection. And they contrive a distributed collaboration route method coalescing Kalman's data fusion and shortest path algorithm to convey the data of distributed target detection. In [8, 9], Yang et al. bring forward an energy efficient route algorithm of distributed target detection based on Neyman-Pearson rule. In the process of establishing math model of routing, authors consider the node energy consumption and the detection probability comprehensively, and then they analyze and compare the routing performance. References [10, 11] take advantage of the feature of tree structure and then discuss the performance of the distributed target detection model based on the network topology of the aggregation tree. Reference [12] achieves to the minimum probability of error and the maximum residual energy by multiobjective optimization method in order to optimize the decision threshold of each node under the distributed network topology environment. In [13], based on the mathematical model of multitarget detection, Ermis and Saligrama advance the method of target detection based on the process of Benjamin-Hochberg. And then they analyze the algorithm performance in an ideal environment. In [14], based on the Neyman-Pearson criterion, Aziz proposes a soft decision fusion algorithm differing from the traditional Zero-One decision fusion. He takes the confidence factor of each node as weighted value to fuse the data. In spite of transmitting more bytes of data and consuming more system energy, it can improve the efficiency of system detection significantly. Reference [15] presents a three-tier ocean intrusion detection system by using accelerometer sensors to detect intrusion ships, which exploits the spatial and temporal correlations of the intrusion to increase the detection reliability. Reference [16] proposes a fully distributed cut vertex detection mechanism called CAM, which can be applied in large-scale, highly dynamic overlay networks. In [17], authors considered the problem of adaptive radar detection of distributed targets in the presence of Gaussian noise with unknown covariance matrix. More precisely, they extend the ABORT (adaptive beamformer orthogonal rejection test) idea to the distributed targets resorting to the GLRT (generalized likelihood ratio test) design criterion. Reference [18] advances a PF-DTBD (particle filter based distributed track-before-detect) algorithm via fusing multisensor local estimated conditional PDF (probability density functions). They apply MKDE (multivariate kernel density estimation) technique to estimate sensors' local PDFs on the basis of finite particles set.

Firstly, this paper analyzes the related works about distributed target detection and then puts forward a new method to improve the system detection efficiency on the base of consuming less energy in wireless sensor network. The specific mathematical expression of P_F and T (notation meaning can be seen in Section 3.1.1) is given out. Ultimately we prove the performance of the method that is better than CR and C-CR through the simulation.

The rest of this paper is organized as follows. Section 2 describes the system model, related definitions, and the process of clustering. And then we detail the decision fusion rule. Next we prove that the method is better than others by simulation tests. Finally we summarize the whole paper in Section 5.

2. System Model and Related Definitions

2.1. Signal Attenuation Model. We assume that the signal power emitted by the target or event source declines as the distance from the target grows. Here we choose polynomial decay model as attenuation model in ROI. The amplitude of signal that sensor node v_i received can be expressed by $s_i = D(P_0, v, u_i)$; that is

$$s_i = \frac{P_0}{1 + \left(\left\| v - u_i \right\| / \delta\right)^{\varepsilon}}, \tag{1}$$

where P_0 is the signal power emitted by the target, v is the coordinate of target signal in the region of interest. u_i is the coordinate of local sensor node i, δ is an adjustable constant of the target signal, and ε is the signal decay exponent and takes value between 2 and 3.

2.2. Mathematical Model. The mathematical model of the distributed target detection is based on binary assumptions of probability and statistics theory. Due to the complexity of environment, we assume that the background noise is white Gaussian noise. In a noisy environment, the signal values detected by sensor nodes can be expressed as

$$r(t) = \begin{cases} n(t), & H_0, \\ u(t) + n(t), & H_{1,} \end{cases} \tag{2}$$

where H_0 means there does not exist a target signal; otherwise H_1 means that there exists a target signal.

Definition 1 (detection probability [DP]). Under the condition that there exists target signal, we can use $P(D_1 \mid H_1)$ to express the probability of detection, then $P(D_1 \mid H_1) = \int_{R_1} f_1(y)dy$, where R_1 is the area that there exists target, $f_1(y)$ is probability distribution of target detected.

Definition 2 (false alarm rate [FAR]). Under the condition that there does not exist target signal, but we detect the signal and express as $P(D_1 \mid H_0)$, then $P(D_1 \mid H_0) = \int_{R_1} f_0(y)dy$, and $f_0(y)$ is probability distribution of target undetected.

2.3. Network Model of Clustering. We assume that the local sensors and data fusion center (DFC) are deployed in ROI of wireless sensor network randomly. The processes of clustering can be described as follow.

(1) Network initialization: each node establishes its own neighbor list by broadcasting message. And DFC is the first-level cluster head of the whole network.

(2) In terms of one cluster-head chosen algorithm such as HeeD [19], DFC selects several sensor nodes from its neighbor list as its subcluster (the second-level) cluster heads. The rest of other neighbors of DFC become the normal nodes.

(3) After that, the second-level cluster heads select some relay nodes in the set of their respective neighbor lists simultaneously to the exclusion of those who have become the cluster heads, and then they send subcluster establishment messages to their respective relay nodes.

(4) Thereafter, relay nodes begin to choose the third-level cluster heads delegated from the second-level cluster heads according to the same cluster-head selection algorithm and build the precursor-successor relationship. Those who are neither relay nodes nor cluster heads would be maintained as normal nodes.

(5) The newly chosen cluster heads repeat step (3) and step (4) above until the clustering process of the entire network is finished.

(6) After building the different levels of subclusters, relay nodes are seen as the ordinary nodes, and the precursor-successor relationship between two clusters determines the order of communication of their respective cluster head nodes.

Thus, the process of clustering in ROI is shown in Figure 1, and actually it forms the network topology of tree structure where DFC becomes the root.

The ordinary nodes in precursor cluster transmit their respective judgments to the head of cluster, and then the head of clusters transmit their respective judgments to the head of successor cluster after fusing the judgments from its members. The path of transmitting the judgments data is shown in Figure 2.

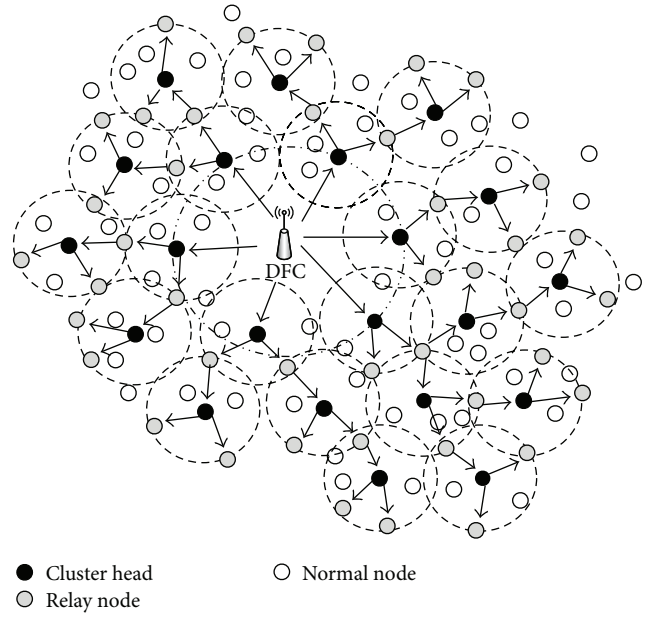


Figure 1: The process of clustering.

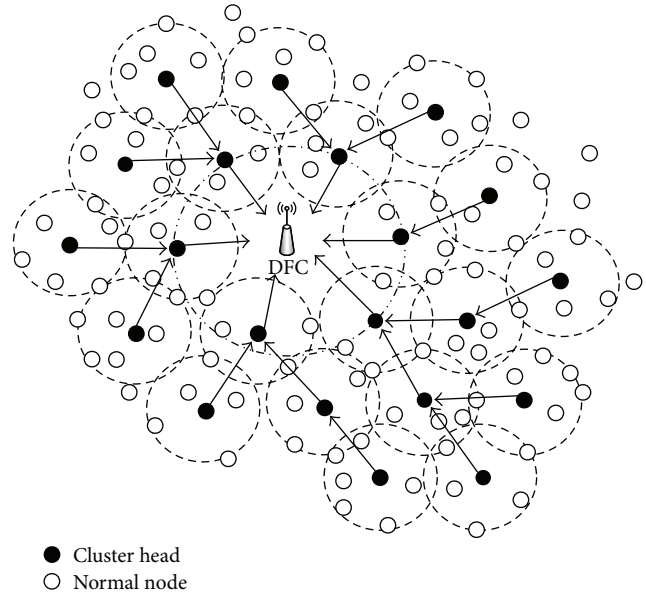


Figure 2: The path of fusing data among the head of cluster.

3. Decision Fusion Rule

3.1. Three Kinds of Clusters. From the previous analysis, we can find that there exists the precursor-successor relationship among clusters. So during the decision fusion, different levels of clusters have different decision fusion methods.

Definition 3 (father and son cluster [FSC]). For any two clusters ϑ_i, ϑ_j if ϑ_j is the next level of cluster ϑ_i, then we can call ϑ_i is the subcluster of ϑ_j.

According to the relationship of precursorsuccessor among clusters, we can divide the clusters into three types, and they are precursor clusters, precursor-successor clusters, and successor clusters separately.

Figure 3(c) depicts the successor cluster whose head is data fusion center, and it has many subcluster head nodes and ordinary nodes. We assume that there are N clusters in the whole network, the precursor cluster set is $\vartheta_{\text{pre}} = \{\dot{\vartheta}_1, \dot{\vartheta}_2, \ldots, \dot{\vartheta}_{N_1}\}$, the precursor-successor cluster set is $\vartheta_{\text{ps}} = \{\ddot{\vartheta}_1, \ddot{\vartheta}_2, \ldots \ddot{\vartheta}_{N_2}\}$, the successor cluster set is $\vartheta_{\text{suc}} = \{\dddot{\vartheta}_1\}$, and $N = N_1 + N_2 + 1$.

W-CDFA (Weight-based Clustering Decision Fusion Algorithm) is a decision fusion algorithm which fuses the weighted data of decision between clusters. We would discuss the decision fusion methods about the different kinds of clusters, described as Figures 3(a) and 3(b), respectively. Because

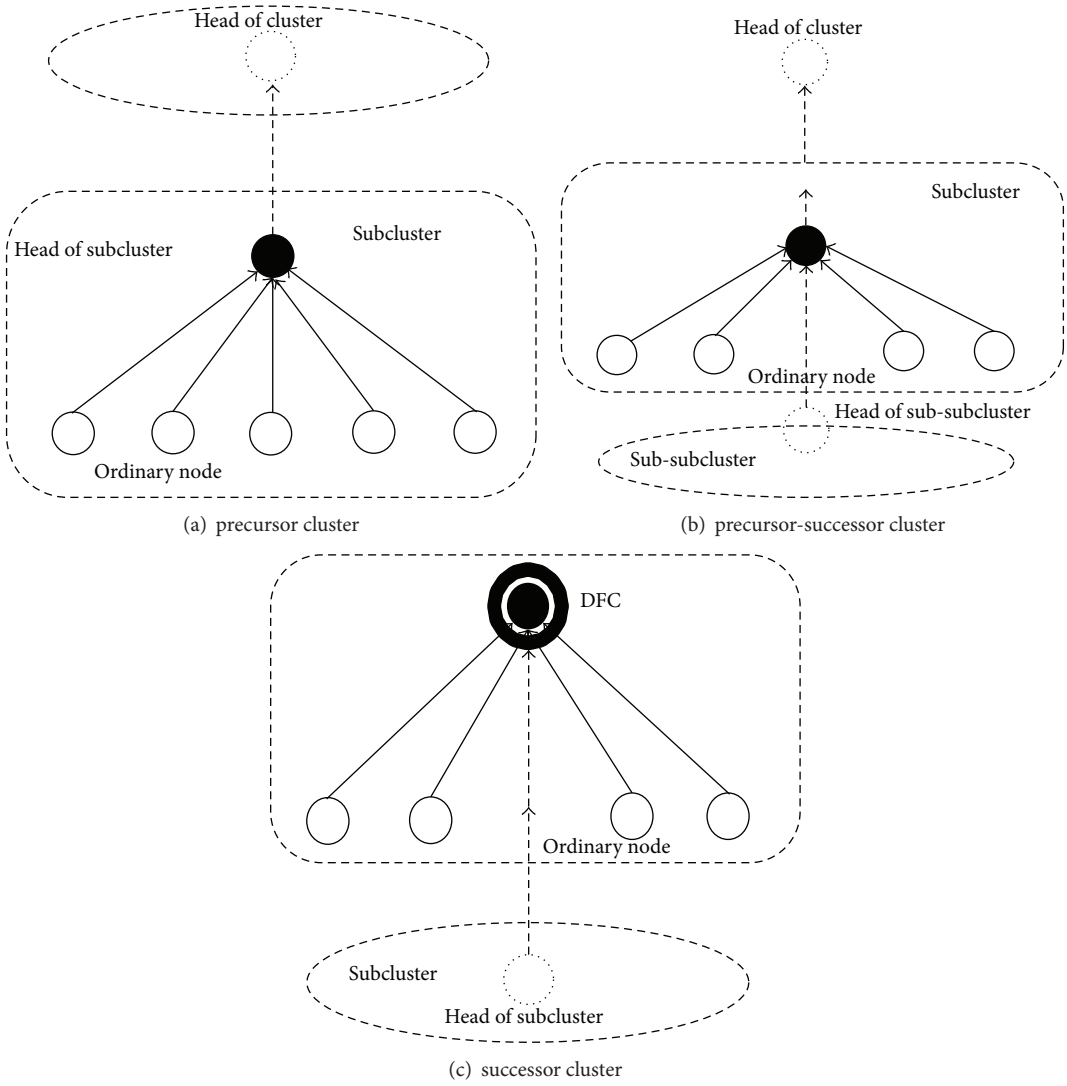


Figure 3: Topology of three types of clusters.

that the successor cluster is the special case of precursor-successor cluster, its method of decision fusion is similar to the precursor-successor cluster.

3.1.1. Decision Fusion of the Precursor Clusters. For the precursor clusters, the method of decision fusion is simple relatively. Since the judging condition of the ordinary nodes in each cluster is equivalent, we set up the weighted factor to the same and to be 1. For each precursor cluster $\dot{\vartheta}_i \in \vartheta_{\text{pre}}$, $i = 1, 2, \ldots, N_1$, the number of nodes is $\dot{n}_i + 1$ including $\dot{n}_i$ ordinary nodes and one head of cluster. The decision of the precursor cluster head is denoted as $\dot{u}_{i0}$, and the decision of the other ordinary nodes are denoted as $\dot{u}_{i1}, \dot{u}_{i2}, \ldots, \dot{u}_{i\dot{n}_i}$. The decision threshold of precursor cluster head is set as $\dot{T}_i$, so the fusion rule at the head of precursor cluster is represented as

$$\dot{u}_{i0} = \begin{cases} 1, & \sum_{j=1}^{\dot{n}_i} \dot{u}_{ij} \geq \dot{T}_i, \\ 0, & \text{otherwise}, \end{cases} \quad i = 1, 2, \ldots, N_1. \tag{3}$$

Then local decision-making fusion rule of precursor cluster head can be simplified as

$$\dot{\Lambda}_i = \sum_{j=1}^{\dot{n}_i} \dot{u}_{ij}, \quad i = 1, 2, \ldots, N_1. \tag{4}$$

3.1.2. Decision Fusion of the Precursor-Successor Clusters. For the precursor-successor clusters (Figure 3(b)), the judgment of the father cluster head not only depends on the judgments of the ordinary nodes in cluster, but also relies on the judgment from its subcluster head nodes, and the decisions uploaded by each cluster members are not the same as the decisions uploaded by subclusters. Since the decision uploaded by the head of subcluster represents the results of all nodes, which would obtain the greater weight value. For any precursorsuccessor $\ddot{\vartheta}_i \in \vartheta_{\text{ps}}$, $i = 1, 2, \ldots, N_2$, the number of ordinary nodes is $\ddot{n}_i$, the judgments are described as $\ddot{u}_{i1}, \ddot{u}_{i2}, \ldots, \ddot{u}_{i\ddot{n}_i}$, the number of subcluster head is $\ddot{m}_i$, their judgments are $\overline{u}_{i1}, \overline{u}_{i2}, \ldots, \overline{u}_{i\ddot{m}_i}$, we assume that the thresholds

of decision are $\overline{T}_{i1}, \overline{T}_{i2}, \ldots, \overline{T}_{i\ddot{m}_i}$, and the judgment of the cluster head nodes is $\ddot{u}_{i0}$.

Since the judgment of subcluster head depends on all judgments of ordinary nodes within subcluster or the judgments of ordinary nodes and its subcluster head nodes, the decision of subcluster head owns a greater correct probability than the ordinary nodes in the father cluster when each head of subcluster conveys its decision to father cluster head. We assume that the weight of the judgment uploaded by each subcluster head is the threshold of subcluster head; therefore, we can conclude that

$$\ddot{u}_{i0} = \begin{cases} 1, & \sum_{j=1}^{\ddot{n}_i} \ddot{u}_{ij} + \sum_{j=1}^{\ddot{m}_i} \overline{T}_{ij}\overline{u}_{ij} \geq \ddot{T}_i, \\ 0, & \text{otherwise}, \end{cases} \quad i = 1, 2, \ldots, N_2. \tag{5}$$

So the decision fusion rule of precursor-successor cluster head is presented as below:

$$\ddot{\Lambda}_i = \sum_{j=1}^{\ddot{n}_i} \ddot{u}_{ij} + \sum_{j=1}^{\ddot{m}_i} \overline{T}_{ij}\overline{u}_{ij}, \quad i = 1, 2, \ldots, N_2. \tag{6}$$

There are only ordinary nodes and DFC in the successor cluster, and it is the exception of the precursor-successor cluster. We assume that the decision threshold of the successor cluster is T_0, and consequently T_0 is the decision threshold of the whole wireless sensor network monitoring system, the method of calculation is as same as that of the precursor-successor, so the decision fusion rule of DFC is $\ddot{\Lambda} = \sum_{i=1}^{\ddot{n}} \ddot{u}_i + \sum_{i=1}^{\ddot{m}} \overline{T}_i\overline{u}_i$, where $\ddot{n}$ is the number of ordinary node in the successor cluster, $\ddot{u}_i$ is the judgment of ordinary node in the successor cluster, $\overline{u}_i$ is the judgment of the successor subcluster head, and $\overline{T}_i$ is the decision threshold of the successor subcluster head.

3.2. Decision Threshold of Different Clusters

3.2.1. Decision Threshold of Precursor Cluster. Through abovementioned description of the method of the decision fusion among three different types of clusters, we assume that the decision threshold of three different clusters are $\dot{T}_i$ (threshold of precursor cluster), $\ddot{T}_j$ (threshold of precursor-successor cluster), and T_0 (threshold of DFC), respectively, $i = 1, 2, \ldots, N_1$, $j = 1, 2, \ldots, N_2$, the specific calculation methods are as follows.

We set the tolerance of false-alarm probability of all clusters in network is α, namely $P_{F_i} = \alpha, i = 1, 2, \ldots, N_1 + N_2 + 1$. For the precursor cluster $\dot{\vartheta}_i$, the false-alarm probability $\dot{P}_{F_i}$ can be expressed as:

$$\begin{aligned} \dot{P}_{F_i} &= p\gamma\left\{\dot{\Lambda}_i \geq \dot{T}_i \mid H_0\right\} \\ &= \sum_{j=\dot{T}_i}^{\dot{n}_i} \binom{\dot{n}_i}{j} p_{\text{fa}}^j (1 - p_{\text{fa}})^{\dot{n}_i - j}, \end{aligned} \tag{7}$$

where $p\gamma\{x\}$ presents the probability value of x, p_{fa} is the false alarm rate of every local sensor.

Obviously, $\dot{\Lambda}_i$ obeys the binominal distribution when the nodes are densely deployed. Based on the central limit theorem, the false-alarm probability $\dot{P}_{F_i}$ can be calculated by using Laplace-DeMoivre approximation as follows

$$\dot{P}_{F_i} \approx Q\left(\frac{\dot{T}_i - \dot{n}_i p_{\text{fa}}}{\sqrt{\dot{n}_i p_{\text{fa}}(1 - p_{\text{fa}})}}\right), \tag{8}$$

where $Q(x) = \int_x^\infty (1/\sqrt{2\pi}) e^{-t^2/2} dt$.

The calculation formula of decision threshold of the precursor cluster $\dot{T}_i$ is

$$\dot{T}_i \approx \sqrt{\dot{n}_i p_{\text{fa}}(1 - p_{\text{fa}})}\, Q^{-1}(\alpha) + \dot{n}_i p_{\text{fa}}. \tag{9}$$

3.2.2. Decision Threshold of the Precursor-Successor Cluster. For the precursor-successor cluster $\ddot{\vartheta}_i$, its false-alarm probability $\ddot{p}_{F_i}$ can be expressed as

$$\ddot{P}_{F_i} = p\gamma\left(\ddot{\Lambda}_i \geq \ddot{T}_i \mid H_0\right). \tag{10}$$

The precursor-successor cluster nodes have two categories: ordinary nodes and subcluster heads; we assume that a of $\ddot{n}_i$ ordinary nodes decides "1" and b of $\ddot{m}_i$ subcluster heads' judgments is "1", so we can conclude the expression $\ddot{T}_i = a + (b/\ddot{m}_i) \sum_{j=1}^{\ddot{m}_i} \overline{T}_{ij}$. Here we assume that the false-alarm tolerance of each cluster is α, so we can get the following expression: $p\gamma\{\overline{u}_{ij} = 1 \mid H_0\} = \alpha$, $p\gamma\{\overline{u}_{ij} = 0 \mid H_0\} = 1 - \alpha$, and (10) can be expressed as

$$\ddot{P}_{F_i} = \sum_{b=1}^{\ddot{m}_i} \sum_{k=a}^{\ddot{n}_i} \binom{\ddot{n}_i}{k} p_{\text{fa}}^k (1 - p_{\text{fa}})^{\ddot{n}_i - k} \sum_{j=b}^{\ddot{m}_i} \binom{\ddot{m}_i}{j} \alpha^j (1 - \alpha)^{\ddot{m}_i - j}. \tag{11}$$

After further simplified calculation, (11) can be described as

$$\ddot{P}_{F_i} = \sum_{b=1}^{\ddot{m}_i} Q\left(\frac{a - \ddot{n}_i p_{\text{fa}}}{\sqrt{\ddot{n}_i p_{\text{fa}}(1 - p_{\text{fa}})}}\right) Q\left(\frac{b - \ddot{m}_i \alpha}{\sqrt{\ddot{m}_i \alpha (1 - \alpha)}}\right). \tag{12}$$

Substituting $\ddot{T}_i = a + (b/\ddot{m}_i) \sum_{j=1}^{\ddot{m}_i} \overline{T}_{ij}$ into (12) we can get

$$\begin{aligned} \ddot{P}_{F_i} = \sum_{b=1}^{\ddot{m}_i} Q\left(\frac{\ddot{T}_i - (b/\ddot{m}_i) \sum_{j=1}^{\ddot{m}_i} \overline{T}_{ij} - \ddot{n}_i p_{\text{fa}}}{\sqrt{\ddot{n}_i p_{\text{fa}}(1 - p_{\text{fa}})}}\right) \\ \times Q\left(\frac{b - \ddot{m}_i \alpha}{\sqrt{\ddot{m}_i \alpha (1 - \alpha)}}\right). \end{aligned} \tag{13}$$

When the number of ordinary nodes in the precursor-successor cluster ($\ddot{n}_i$) and the number of subcluster heads ($\ddot{m}_i$) are given, and decision threshold of each cluster head node corresponding the subcluster $\overline{T}_{ij}$ is known, so (11) is a function of the decision threshold $\ddot{T}_i$ which is the sole

variable, namely, $\ddot{P}_{F_i} = \Psi(\ddot{T}_i)$. For the given tolerance of false-alarm probability α, the decision threshold of cluster $\ddot{T}_i$ can be calculated according to

$$\Psi\left(\ddot{T}_i\right) = \sum_{b=1}^{\ddot{m}_i} Q\left(\frac{\ddot{T}_i - \left(b/\ddot{m}_i\right)\sum_{j=1}^{\ddot{m}_i}\overline{T}_{ij} - \ddot{n}_i p_{\mathrm{fa}}}{\sqrt{\ddot{n}_i p_{\mathrm{fa}}\left(1-p_{\mathrm{fa}}\right)}}\right) \times Q\left(\frac{b-\ddot{m}_i\alpha}{\sqrt{\ddot{m}_i\alpha\left(1-\alpha\right)}}\right) = \alpha. \tag{14}$$

So $\ddot{T}_i = \Psi^{-1}(\alpha)$.

Since the successor cluster is just an exception of the precursor-successor cluster, so for the system decision threshold T_0, the calculation method is similar to that of the precursor-successor cluster.

4. Experiments and Simulations

We assume that the region of interest is a square with length of edge 100, the target signal is in the place with the coordinate (50, 50), the coordinate of data fusion center is (50, 50), and DFC is the head of all clusters. The sensor nodes are deployed densely in ROI which obeys the uniform random distribution. The random noise in ROI is Gaussian white noise, and it follows the standard normal distribution.

Utilizing the Monte Carlo random methods, the experiment scene is shown in Figure 4.

We divide the region of interest into 5∗5 clusters; the length of each cluster edge is 20. In order to do experiment conveniently, we establish the precursor-successor relationship among those clusters based on tree clustering structure in Section 2.3. Shown in Figure 4, the number of the precursor cluster is 12; the number of the precursor-successor cluster (including the successor cluster of data fusion center) is 13. In the precursor-successor cluster, each cluster has only one father cluster, but it can have many subclusters. In all clusters, there is one precursor successor that has four subclusters, named as center cluster, there are two clusters that have three subclusters, and four clusters that have two subclusters, and six clusters that have only one subcluster. Each node closed to the cluster center of square is set as cluster head figured by solid circle. In order to facilitate the description, we set the label as 1-1, 1-2 · · ·1-5, 2-1, 2-2 · · · 2-5 · · ·5-4, and 5-5 in accordance with the order from left to right every line from the beginning of the upper left corner of ROI.

We calculate the decision threshold values of twenty-five clusters and display parts of them in Table 1. We assume that the tolerance of false-alarm probability of each node marked as p_f is equal to 0.2; the tolerance of false-alarm probability of each cluster α is equal to 0.2 and that of the whole system P_F is equal to 0.1.

From Table 1, we can conclude that the threshold of the whole system depends on that of every cluster; namely, the system model does not make final judgment simply based on independent clusters like C-CR. Actually, from Figure 4, it is clear that the decision of every successor cluster relies on ordinary nodes in cluster and its subclusters, and the relationship makes the system as a whole.

Table 1: Threshold of different clusters.

Label	Number of nodes				
	2000	4000	6000	8000	10000
1-1	18	36	51	72	83
1-2	26	50	75	99	127
1-3	18	34	52	71	85
1-4	27	50	76	100	120
1-5	19	37	53	70	84
2-1	19	35	54	71	87
2-2	39	78	112	151	186
2-3	62	136	207	275	343
2-4	39	74	112	152	183
2-5	18	36	51	67	85
3-1	18	37	50	71	86
3-2	27	51	74	100	123
3-3	105	220	330	437	545

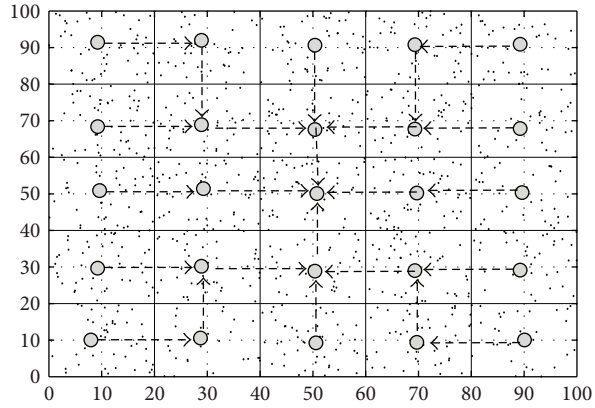


Figure 4: Experiment scene.

For different schemes, T for different values of N is shown in Figure 5. From the previous analysis, we know that the precursor-successor relationship determines that the decision of the successor clusters depends on the decision of their precursor clusters. And this method wipes off the effect of nodes whose judgment is 0, so the threshold of W-CDFA is fewer than that of CR. However, the threshold of C-CR depends on the number of clusters merely. We can conclude that the threshold of C-CR will not go beyond the number of clusters, so it is no worth discussing the threshold of C-CR in practical. In Figure 6, we show the relationship between the system-level threshold T and false alarm rate p_f.

We assume that noises in ROI are i.i.d (independent identical distribution) and follow the standard Gaussian distribution. Here we do 1000 times Monte Carlo experiments under the condition of SNR = 6 and SNR = 3, respectively. We can see the detection performance of CR, C-CR, and W-CDFA in Figures 7(a) and 7(b) and conclude that the detection capability of CR is the best. In [5], authors have derived that the performance of CR would be better with the growth of the number of sensor nodes. However, it is not useful in reality when the length of ROI is very large because

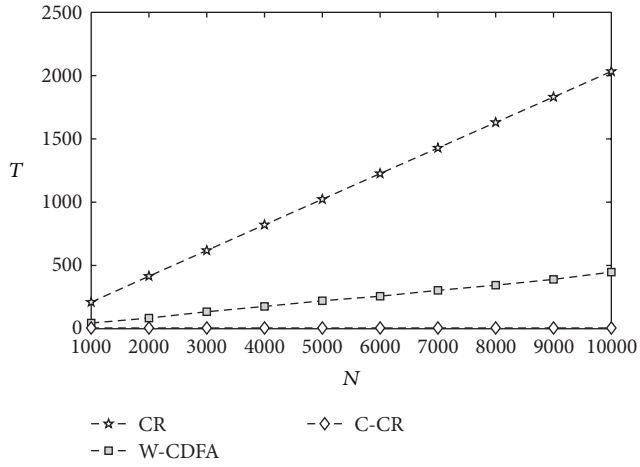


Figure 5: T for different values of N.

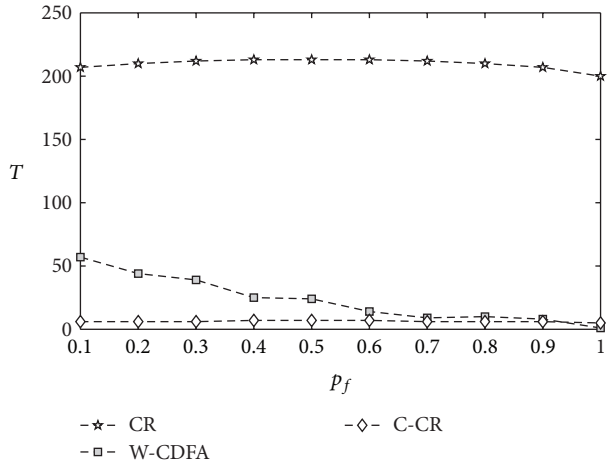
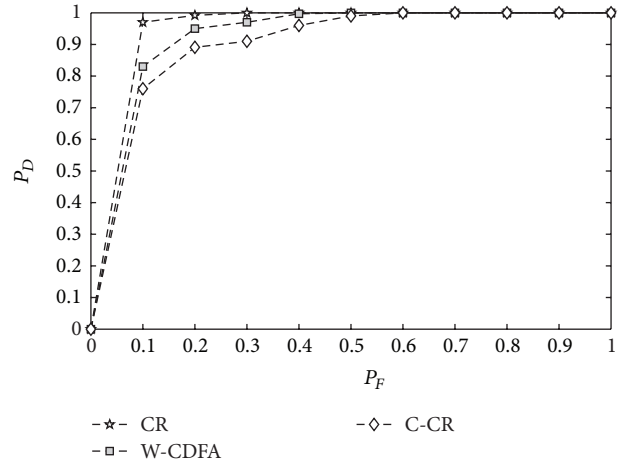
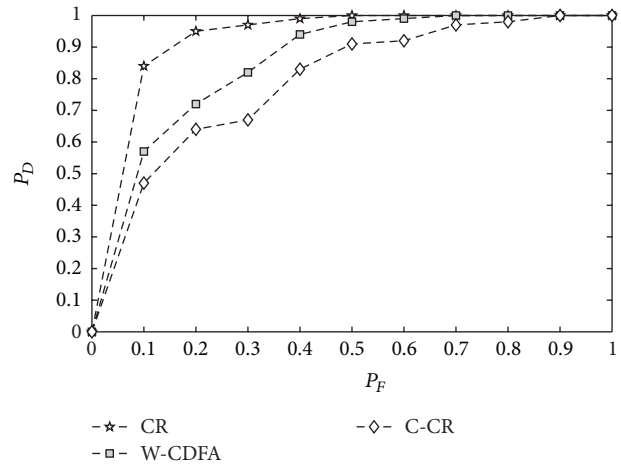


Figure 6: T for different values of p_f.



(a) SNR = 6



(b) SNR = 3

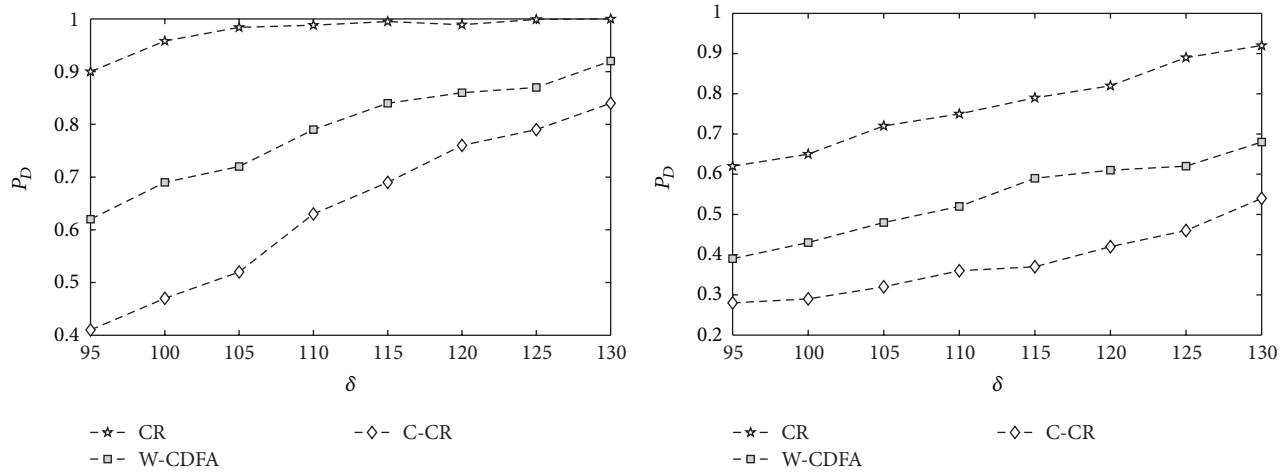
Figure 7: ROC of system through Monte Carlo ($N = 1000$; $\delta = 120$; $p_f = 0.2$; $\alpha = 0.2$; $\varepsilon = 3$).

that the energy consumption grows fast. The performance of W-CDFA proposed in this paper is better than C-CR. Because C-CR loses lots of useful information in its first fusion step, and this method does not utilize the relativity relationship among sensor nodes.

P_D as a function of δ is shown in Figures 8(a) and 8(b). As δ increases, the detection probability P_D increases synchronistically. This is because the larger the number of sensor nodes is, the higher the requirement of collaboration among sensor nodes is. However, the number of sensor nodes within the effective communication region of target signal would reduce when target signal attenuates quickly. So we can understand easily that the larger the signal attenuation factor δ is, the larger the detection probability P_D is. And from the Figure 8, we can find that changing of parameter δ will not have effect on the curve relation among CR, C-CR, and W-CDFA because that changing of parameter δ only has impact on the range value of signal detected.

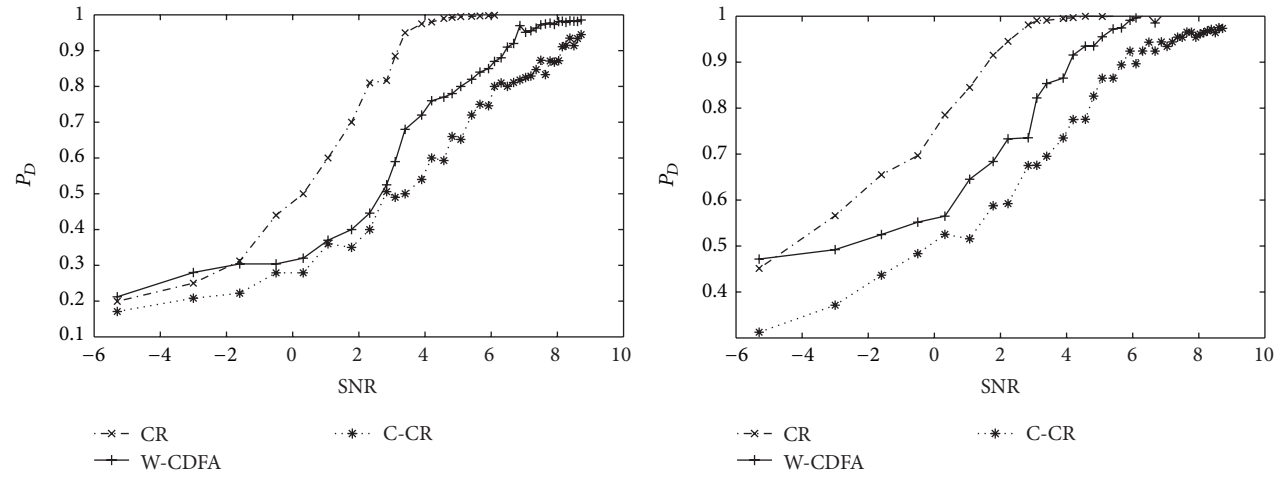
P_D as function of SNR is plotted in Figures 9(a) and 9(b). From the simulation results we conclude that the SNR of the larger the target signal is, the larger the detection probability of target signal is. Actually we can detect the target signal easily if signal power is large enough or noise interference is small enough. Though the detection probability of W-CDFA is lower than CR when SNR is large, we also can find that of W-CDFA is higher than CR and C-CR when SNR is small which may be more useful in real network environment. Meanwhile the superiority of W-CDFA also appears in the aspect of energy consumption.

Under the same condition of network, we assume that the total energy of network is 1 J. Figure 10 shows the change of remaining energy with the frequency of experiment testing. We can find that W-CDFA's energy consumption of all nodes is slower than CR and C-CR. It is easy to understand that the way of single hop communication will cost more energy than the way of multihop communication. This also is why the curve trend difference between CR and C-CR is remarkable in Figure 10. We enlarge one part of curve of C-CR and W-CDFA, and we can find there exists a tiny discrimination between them because the heads of clusters transmit their judgments to DFC directly in C-CR; however, the heads of



(a) SNR = 6 (b) SNR = 3

Figure 8: P_D for different values of δ ($N = 1000$; $P_F = 0.1$; $p_f = 0.2$; $\alpha = 0.2$; $\varepsilon = 3$).



(a) $P_F = 0.1$ (b) $P_F = 0.3$

Figure 9: P_D for different values of SNR ($N = 1000$; $\delta = 120$; $p_f = 0.2$; $\alpha = 0.2$; $\varepsilon = 3$).

cluster transmit their judgments to the successors and finally to DFC by the way of multihop communication in the W-CDFA rule.

5. Conclusion and Discussion

In this paper, we propose a Weight-based Clustering Decision Fusion Algorithm based on the large-scale wireless sensor network to achieve more effective distributed target signal detection. Compared with other typical schemes, the most significant feature of this proposed method is the use of tree structure-based clustering algorithm to create the precursor-successor relationship among clusters and meanwhile derives the decision fusion criterion on signal judgment based on these relationships.

We can analyze and calculate the system-level false alarm probability from the beginning of the precursor clusters. Thereby we can obtain the system-level judgment threshold. In order to demonstrate that W-CDFA can realize better performance than C-CR, we set the simulation scenarios and experiments that take relationships among clusters into account. Simulation results show that the fusion rule can get satisfactory system-level performance at a low signal to noise ratio. Due to the exclusion of those useless data before making system level judgment, the system threshold in W-CDFA is lower than C-CR which is an important system performance guideline.

It is difficult to detect target signal in the noisy environment. As we know, the most important factor that affects detection performance is the signal to noise ratio (SNR). In Figure 9, we can find the performance of W-CDFA is better than CR, and C-CR with lower SNR and harsher condition ($P_F = 0.1$). So W-CDFA obtains better approbation when background noise is blatant relatively. In Figure 8, we can

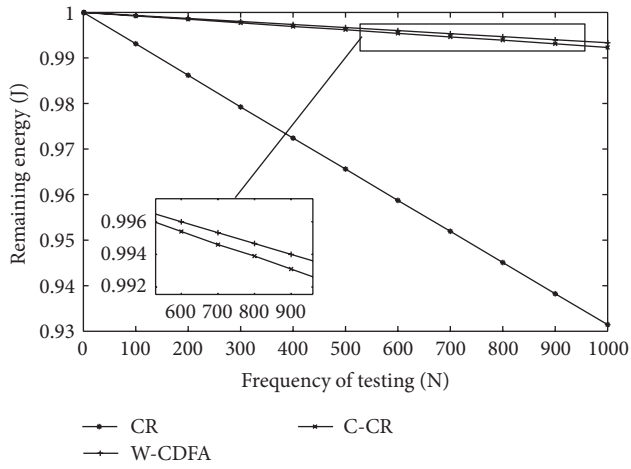


Figure 10: Comparison of the remaining energy.

conclude that the parameter (δ) would not have effect on the relation of the curve trends of W-CDFA, CR and C-CR because that the parameter (δ) only changes the value of received signal but not changes the signal to noise ratio (SNR).

For the purpose of convenience of simulation experiments, we even manual intervene the establishment of the precursor-successor relationships among clusters, but this intervention should be alleviated for the validity. In the future, those works will be further investigated.

Acknowledgments

The subject is sponsored by the National Natural Science Foundation of China (no. 61170065, 61003039), Scientific and Technological Support Project (Industry) of Jiangsu Province (no. BE2012183), Natural Science Key Fund for Colleges and Universities in Jiangsu Province (12KJA520002), Postdoctoral Foundation (2012M511753, 1101011B), Science and Technology Innovation Fund for higher education institutions of Jiangsu Province (CXZZ11-0409), Fund of Jiangsu Provincial Key Laboratory for Computer Information Processing Technology (KJS1022) and Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (yx002001).

References

[1] R. R. Tenney and N. R. Sandell, "Detection with distributed sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 17, no. 4, pp. 501–510, 1981.

[2] Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22, no. 1, pp. 98–101, 1986.

[3] E. Drakopoulos and C. C. Lee, "Optimum multisensor fusion of correlated local decisions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 27, no. 4, pp. 593–606, 1991.

[4] M. Kam, Q. Zhu, and W. S. Gray, "Optimal data fusion of correlated local decisions in multiple sensor detection systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 916–920, 1992.

[5] R. Niu and P. K. Varshney, "Decision fusion in a wireless sensor network with a random number of sensors," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 4, pp. 861–864, March 2005.

[6] N. Katenka, E. Levina, and G. Michailidis, "Local vote decision fusion for target detection in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 329–338, 2008.

[7] Y. Sung, S. Misra, L. Tong, and A. Ephremides, "Cooperative routing for distributed detection in large sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 2, pp. 471–483, 2007.

[8] Y. Yang, R. S. Blum, and B. M. Sadler, "Energy-efficient routing for signal detection in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2050–2063, 2009.

[9] Y. Yang, R. S. Blum, and B. M. Sadler, "A distributed and energy-efficient framework for Neyman-Pearson detection of fluctuating signals in large-scale sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1149–1158, 2010.

[10] F. Wuhib, M. Dam, and R. Stadler, "Decentralized detection of global threshold crossings using aggregation trees," *Computer Networks*, vol. 52, no. 9, pp. 1745–1761, 2008.

[11] W. P. Tay, J. N. Tsitsiklis, and M. Z. Win, "Bayesian detection in bounded height tree networks," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, pp. 4042–4051, 2009.

[12] E. Masazade, R. Rajagopalan, P. K. Varshney, C. K. Mohan, G. K. Sendur, and M. Keskinoz, "A multi-objective optimization approach to obtain decision thresholds for distributed detection in wireless sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 2, pp. 444–457, 2010.

[13] E. B. Ermis and V. Saligrama, "Distributed detection in sensor networks with limited range multimodal sensors," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 843–858, 2010.

[14] A. Aziz, "A soft-decision fusion approach for multiple-sensor distributed binary detection systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 2208–2216, 2011.

[15] H. Luo and Z. Guo, "Ship detection with wireless sensor networks," *IEEE Transaction on Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1336–1343, 2012.

[16] X. Liu, "A fully distributed method to detect and reduce cut vertices in large-scale overlay networks," *IEEE Transaction on Computers*, vol. 61, no. 7, pp. 969–985, 2012.

[17] C. Hao, J. Yang, and X. Ma, "Adaptive detection of distributed targets with orthogonal rejection," *IET Radar, Sonar and Navigation*, vol. 6, no. 6, pp. 483–493, 2012.

[18] Y. Gong, H. Yang, W. Hu, and W. Yu, "An efficient particle filter based distributed track-before-detect algorithm for weak targets," in *Proceedings of the International Radar Conference (IET '09)*, April 2009.

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

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 690974, 10 pages
http://dx.doi.org/10.1155/2013/690974

Research Article

RoadGate: Mobility-Centric Roadside Units Deployment for Vehicular Networks

Yongping Xiong,[1] Jian Ma,[1] Wendong Wang,[1] and Dengbiao Tu[2]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China*
[2] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*

Correspondence should be addressed to Dengbiao Tu; tudengbiao@163.com

Received 31 December 2012; Accepted 12 February 2013

Academic Editor: Jianwei Niu

With the increase of the storage capacity, computing, and wireless networking of the vehicular embedded devices, the vehicular networks bring a potential to enable new applications for drivers and passengers in the vehicles. Due to the prohibitive cost of deployment and management of a roadside unit (RSU), it is difficult to cover roads with a large number of RSUs so that every vehicle can always keep a connection with the nearby RSU. In this paper, we study the problem of deploying the RSUs to provide the desired connectivity performance while minimizing the number of the deployed RSUs. The key idea of our solution is to exploit the time-stable mobility pattern to find the optimal deployment places. We analyze a realistic vehicle trace, observe the mobility pattern, and propose a graph model to characterize it. Based on the graph model, we transform the gateway deployment problem into a vertex selection problem in a graph. By reducing it into the minimum vertex coverage problem, we show that the RSU deployment problem is NP-complete. Then, a heuristic algorithm *RoadGate* is proposed to search greedily the optimal positions. Extensive simulations based on the synthetic and realistic scenarios are carried out to evaluate the performance. The results show that *RoadGate* outperforms other approaches in terms of the number of required RSUs and the actual achieved coverage performance.

1. Introduction

Today, a growing number of vehicles are equipped with the embedded communication devices that can facilitate vehicle-to-vehicle and vehicle-to-infrastructure communication. Increased storage capacity, computing, and communications power, coupled with the advanced wireless networking technology, bring a potential to enable new applications for drivers and passengers in the vehicles. Therefore, vehicular ad hoc networks (VANETs) recently have started to attract attention from many researchers in both industry and academia. The Federal Communications Commission (FCC) in the United States has allocated specifically 5.850–5.925 GHz band and enacted the dedicated short range communications (DSRCs) standard using that band [1]. DSRC is designed to support an intelligent transportation system (ITS) with public safety and private operations for vehicle-to-roadside units (RSUs) and intervehicle communications.

In the typical vehicular networks applications, the vehicles can be equipped with various sensors to collect the traffic and environmental data such as air pollution level, pavement condition, driving habits, and road congestion. The data are reported to the backend servers via the wireless interface embedded in the vehicles. Besides, those Internet services including email, news, entertainment, and location-based services such as ads and navigation are also be provided by the vehicle-to-RSU communication paradigm.

However, a number of technical challenges should be solved before vehicular networks become a reality. In vehicle-roadside communication, RSUs act as RSUs to the Internet and to the infrastructure of other systems such as an ITS, vehicles transmit their gathered data and Internet access requests to RSUs. RSUs send responses to the Internet queries and road information to vehicles. Due to the prohibitive cost of deployment and management of an RSU (typical cost is 3000$/node [2]), it is difficult to cover roads with a large number of RSUs so that every vehicle on road can always be connected to at least one nearby RSU. The solution that can leverage intermittent connectivity provided by RSUs is more

scalable and competitive. The experiments in various controlled environments have confirmed the feasibility of RSU-based vehicular Internet access for noninteractive applications. However, solutions based on intermittent connectivity of RSUs can provide opportunistic services without any worst case service guarantees, which poses great difficulty to its application.

In this paper, we study the problem of RSUs placement that required the minimum number of RSUs, with the vehicle-to-RSU contact probability guarantee given that an intermittent single-hop connectivity exists between vehicles and RSUs in a road region. We firstly analyze a realistic vehicles trace and observed that there is time-stable statistical mobility pattern in the realistic trace. Then, we propose a graph model to characterize this pattern and show that the RSUs deployment problem is NP-complete by reducing it into the minimum vertex coverage problem. Finally, we propose a heuristic greedy algorithm *RoadGate* to find the optimal locations. Extensive experiments in the synthetic and realistic scenarios are carried out to evaluate the performance of our solution. The results show that our solution achieves the desired coverage performance and minimize the number of the required RSUs.

We make the following contributions in this paper.

(1) We disclose the time-stable statistical mobility pattern existing in the realistic vehicles and develop a graph model to characterize it.

(2) We show that the RSU deployment problem is NP-complete and propose a heuristic algorithm to find the optimal places.

The rest of this paper is organized as follows. In Section 2, we briefly review the related works. We firstly describe the system model in Section 3, then analyze a realistic vehicle trace to verify its time-stable mobility pattern, and propose a graph model to characterize it in Section 4. We formulate the RSU deployment problem and prove it is NP-complete, then develop a heuristic algorithm in Section 5. In Section 6, we evaluate the solution performance in two scenarios. Finally, we make conclusions in Section 7.

2. Related Works

Wireless AP or Base Station placement is a well-known research topic in the cellular, wireless sensor networks, and mesh networks; however, most of the works that have addressed this problem so far consider a continuous infrastructure radio coverage. Here, we just describe those most relevant RSUs deployment works in vehicular networks.

There are some works studying the feasibility of leveraging the RSUs in the vehicular networks. Drive-thru Internet [3] was first introduced in the paper, which shows that a vehicle moving with the velocity of 180 km/h can access internet data via a roadside AP. Refrences [4, 5] confirm the feasibility of WiFi-based vehicular Internet access for noninteractive applications. Cabernet [6] aims to deliver data to and from moving vehicles by using WiFi access points. It only provides the intermittent network connectivity with the current deployed APs. Cartel [7] is a mobile sensor computing system which collect, process, and deliver data from vehicular sensors to the server located in Internet by opportunistically using the roadside APs. All these works are assumed to utilize the unplanned deployment APs without the service guarantee. The feasibility of information dissemination using stationary supporting units (SSUs) is investigated in [8] mainly based on computer simulations. However, the deployment issues have not been carefully studied in these works.

Thus, the dedicated RSUs are proposed to be integrated into the vehicular networks to achieve the system scalability, and various RSUs deployment strategies are developed. Banerjee et al. [2] consider a simple nonuniform strategy that places more stationary nodes in the network core. However, it was completely based on intuition without providing any performance guarantees. Alpha coverage [9] provides the intermittent coverage and guarantees the number of contact between vehicles and RSUs. The authors further present an efficient deployment method that maximizes the worst case contact opportunity under a budget constraint [10]. Li et al. [11] consider the optimal placement of RSUs to minimize the average number of hops from APs to RSUs. Lee and Kim [12] seek optimal placement of RSUs by analyzing the number of the reported locations per minute by taxis to telematics system. Lochert et al. [13] use genetic algorithm for optimal placement of RSUs for a VANET traffic information system. The optimal placement is aimed at minimizing the travel time based on aggregated sharing of traffic information. A centrality-based AP deployment scheme was proposed to optimize the end-to-end delay in [14]. Trullols et al. [15] consider that a given number of RSUs have to be deployed for disseminating information to vehicles in an urban area. They formulate it as a maximum coverage problem (MCP) and seek to maximize the number of vehicles that get in contact with the RSUs over the considered area. The deployment scheme proposed in [16] can guarantee that vehicles at any place could communicate with RSUs in certain driving time by proving it equivalent to the set-covering problem. In [17], the deployment optimization objective guarantees a required maximum vehicle-to-RSU data packet delivery delay with a certain predetermined delay violation probability. Besides, [18] uses the game theory to model the RSUs deployment when multiple operators perform their deployment decisions concurrently.

3. System Model

In this section, we firstly present the system model and then give the problem description.

As depicted in Figure 1, a typical VANET consists of three entities in city scenarios: the top server, the fixed RSUs or gateway along the roadside, and the mobile OBUs (on-board units) equipped on the running vehicles. The servers depend on the specific application, and we will not give detailed description. All RSUs are planned to provide communication services. The RSUs and OBUs are equipped with short-range radio interfaces such as 802.11 b/g/p, and they can exchange

data when entering into their mutual transmission ranges. The RSUs have the powerful storage space to cache the data reported by vehicle or the disseminated data from the server. Moreover, RSUs may connect to the server in the Internet to download or upload data.

Considering a limited geographical region, vehicles enter and leave, autonomously and continuously, the region. All RSUs can be installed at any place in the region. When a vehicle moves into the radio range of any RSU, it may use the opportunity to establish connectivity with the RSU and then send or receive data from it. The capacity of the communication system gets fairly important when the amount of transferred data becomes large. The capacity of the intermittently connected network relies on a few factors such as the vehicle speed, the radio range, and the data rate. For example, the higher the data rate is, the larger the throughput is. However, in this paper, we focus on the impact of *meeting probability* which indicates the possibility that a vehicle can access the deployed RSU when it goes through the area.

Our goal is to guarantee the meeting probability specified by users while minimizing the deployment cost, say, the number of the required RSUs. Specifically, the meeting probability is the probability with which any vehicle can enter the communication range of at least on installed RSU after it moves within a given distance from entering the region. This enables our solution work at the situation that the vehicles sojourn or stay within the area. Note that we do not take the possible available open access points into account, but our work provides a lower bound of system performance when that way is allowed.

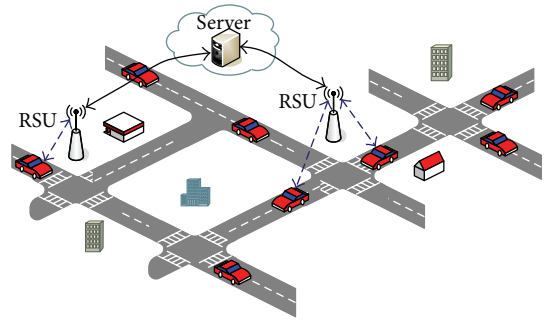


Figure 1: System model for vehicle-to-roadside communication.

Figure 2: Road network and traffic in MMTS.

4. Mobility Graph

In this section, we firstly analyze a realistic vehicle trace and observe the time-stable statistical mobility pattern. Then, we give the definition of the graph model to characterize the mobility pattern.

4.1. Analysis on Realistic Vehicle Trace. Some existing works also study the performance enhancement in mobile networks by deploying stationary nodes. But these works basically assume the nodes move according to the simplistic random models. These models are usually easy to implement in simulations and allow statistical analysis of large-scale protocols and systems. However, they do not capture the characteristic of people move in realistic environment.

Recent that studies [19] on some realistic traces of moving users show that nodes within a social environment do not move completely randomly. Instead, they usually move around a set of landmarks such as home, office, and park. Specifically, nodes show preference for a small number of landmarks and would move less often to the neighborhood of other landmarks. The second observation is that in some social environments the node trajectory in time is almost deterministic [20]. This means a node has its own mobility schedule and it generally moves between landmarks according to that schedule, subject to few random deviations.

Above observations disclose the long-term mobility pattern of people in reality. Inspired by above observations from the study on the realistic people traces, Piorkowski also [21] analyzed a realistic GPS-based mobility traces of taxi in San Francisco, USA. This dataset contains the GPS coordinates of 665 taxis collected over 30 days in the Bay Area. His work verifies that the spatially heterogeneous mobility pattern appears to be stable in time.

As the taxi trace is only a sample of the urban mobility pattern, we use a realistic vehicle trace containing different type mobile nodes (buses, taxis, pedestrians, etc.). This trace is generated by MMTS (multiagent microscopic traffic simulator) [22] which accurately models the public and private traffic over real regional road maps of Switzerland with a high level of realism [23]. Figure 2 shows the simulation involving around 260000 vehicles in an area of around 250 km × 260 km in the canton of Zurich, the largest city in Switzerland.

For the purpose of our analysis, we extract the traffic data over around 18 hours in an area of around 3000 m × 3000 m in the central city of Zurich. The whole area is divided into a set of nonoverlapped uniform zones. Each zone has an area of 200 m × 200 m, 300 m × 300 m, and 600 m × 600 m, respectively, in three experiments. With the time unit of 20 minutes, we count the number N_i of vehicles passing zone i and the number N_{ij} of vehicles entering zone j after leaving zone i for each time unit. Then, we use N_{ij}/N_i as the estimation of transition probability. In each experiment, we record and plot the transition probability of five pairs of zones shown in Figure 3.

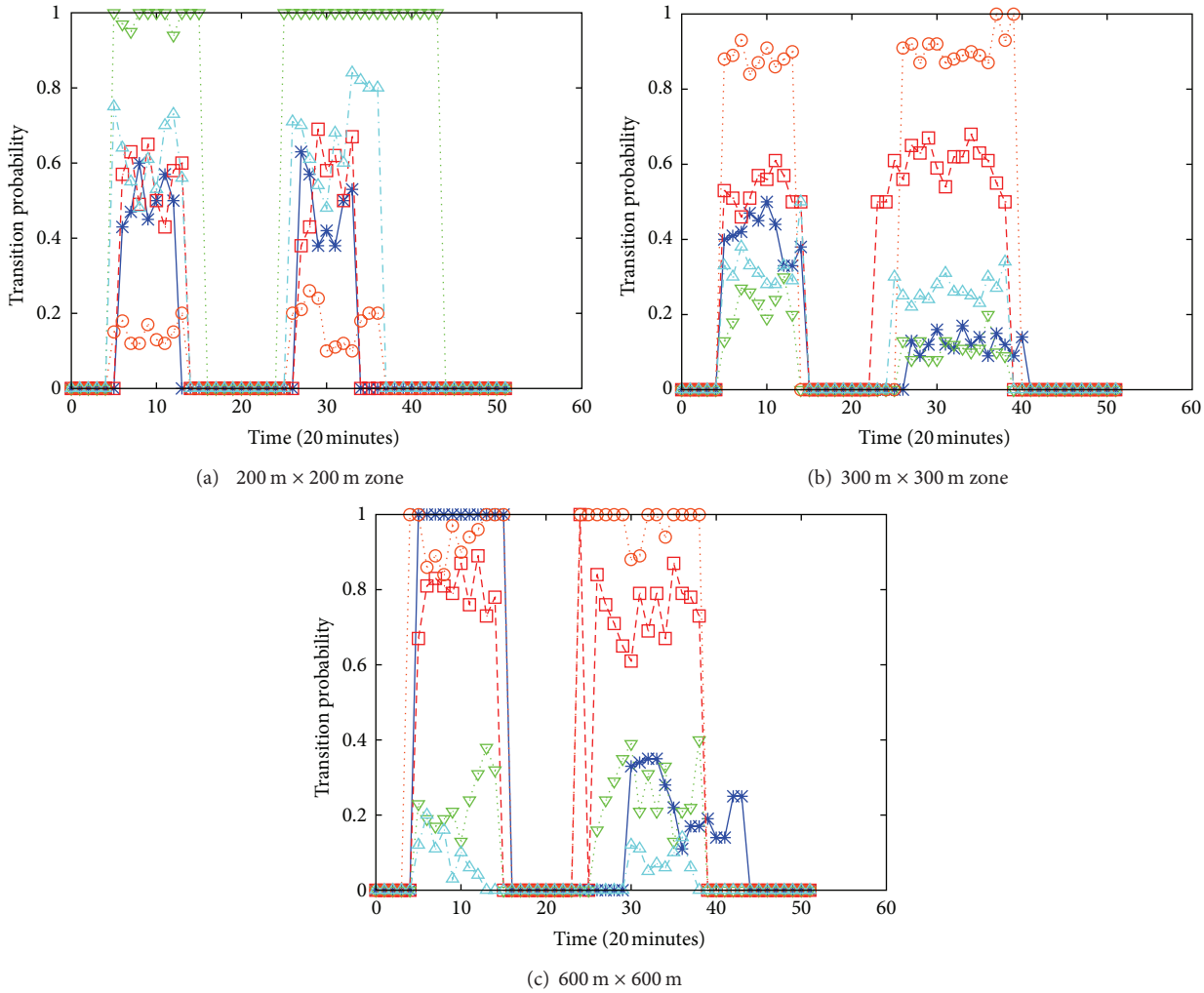


Figure 3: Transition probability for zone of different area.

Some key characteristics are observed from the above statistical results. (1) The transition probabilities between two zones stay approximately stable which fluctuate around a mean value within two traffic peak time. (2) The transition probability between two same zones is different in two peak time, as depicted by blue lines in Figures 3(b) and 3(c). The possible reason is the regular mobility behaviors of people. For example, people usually move from their home to office in the morning and return along the inverse routine in the evening. (3) There is no obvious impact of the size of zone on the time stability of transition probability. However, the transition probability may vary with time when the zone is very small. We do not consider the case because our zone division is big enough.

4.2. Definition. Based on the above observations, we define a *mobility graph* to characterize the time-stable statistical mobility pattern in a region.

Let us consider a connected and bounded geographical area A, we divide it into s nonoverlapped uniform zones. A Mobility Graph is a directed graph G, whose vertex set $\mathscr{V}(G)$ corresponds to the set of zones. Its edge set $\mathscr{L}(G)$ corresponds to the set of mobility links between zones on which vehicles travel. There exists a mobility link between two neighboring zones i and j only if a vehicle leaves zone i and then enters zone j immediately. Each edge is associated with a transition probability which indicates the probability that a random node moves from i to j. We use the similar approach used in Section 4.1 to compute the transition probability. Let T denote the time unit, the transition probability is computed as follows:

$$P^T(i,j) = \frac{\left|\left\{\mathscr{N}_i(T) \cap \mathscr{N}_j(T)\right\}\right|}{\left|\mathscr{N}_i(T)\right|}, \quad (1)$$

where $\mathscr{N}_i(T)$ and $\mathscr{N}_j(T)$ are, respectively, the set of vehicles located in zones i and j within a time unit T. As stated in Section 4.1, transition probability between two zones is basically time stable for a long period of time and changes for another duration in a day. Then, the average of all time units in the total statistical time is computed as the final weight of the corresponding edge.

We also explore the mobility process of all vehicles passing the boundary of the area A. We introduce a virtual vertex U to represent the exterior zone beyond A. If there are the vehicles entering A from the bounding zone i, then an edge exists between vertex U and vertex i. The corresponding transition probability is computed in the similar way to the ordinary edges. An example is shown in Figure 4. Its left part shows that the geographical area is divided into 6 zones, say, zone from 1 to 6, and the corresponding mobility graph is depicted in right side. Here, we find that all the vehicles only move into the area from 1 and 4 zones with the probability of 0.8 and 0.2, respectively. Meanwhile, they only leave the area from zones 2 and 6 with the probability of 0.3 and 0.6, respectively. The red curves represent, respectively, the physical path 1-3-4-6 in the geographical area and that on the Mobility Graph.

Now, we discuss some methods to optimize the constructed mobility graph. First, we delete those vertices corresponding to the zones which do not contain roads because all vehicles cannot move into them. Additionally, we also delete those vertices that contain only a road segment because they have no chance of moving to other zones but the two fixed neighbors. Finally, those edges associated with the transition probability smaller than a given threshold should be removed because they nearly have no impact on the final computation.

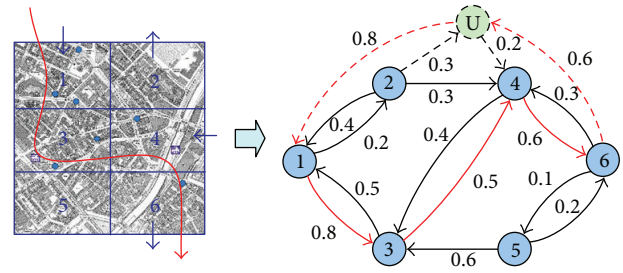


Figure 4: A geographical area and the corresponding mobility graph.

Table 1: Summary of major notations.

A	Target geographical area
G	Mobility graph
$\mathscr{V}(G)$	Vertex set
$\mathscr{L}(G)$	Edge set
$\mathscr{M}(G)$	Transition matrix of graph
$P(i, j)$	Transition probability from vertex i to j
π_{ij}^{δ}	Transition probability from i to j in at most δ hops
$\lambda_{i\mathscr{R}}$	Visiting probability from vertex i to group $\mathscr{R}$
$\widehat{\mathscr{V}}$	Optimal vertex set being searching
δ	Maximum hops
P_u	Meeting probability threshold specified by users

5. RSU Deployment Problem

In this section, we firstly present the several definitions and assumptions, then formally define the Minimum RSUs deployment problem (MRDP). Finally, we prove it is NP-hard and develop a greedy heuristic algorithm to find the optimal deployment solution.

5.1. Assumptions and Definitions. For the sake of convenience to make our idea clear, we make the following assumptions. We assume that each zone can be covered by the communication range of an RSU. That is to say, the vehicles can exchange data with the installed RSU once they enter the zone. We assume that the transmission ranges of all RSUs are fixed, so we have to adjust the size of zones to meet that assumption. Let us take the RSU integrating the 802.11 g interface as an example. The outdoor standard transmission distance of 802.11 g is around 300 m. Thus, the area of a zone may be selected as $400 \times 400\ \mathrm{m}^2$ if the RSU is placed at the center of the zone. The size of a zone should be set smaller while considering the signal decay caused by the buildings. In the following, we introduce several key definitions. The symbols and notations used in the paper are summarized in Table 1.

Definition 1 (transition matrix). Since there are the time-stable transition probabilities between all zones, the statistical mobility pattern can be represented by a time homogeneous Markov chain. Its state space is exactly corresponding to the vertex set, say, all zones. Therefore, the transition probability distribution between the state space can be represented by the transition matrix $\mathscr{M}(G)$. A journey of a vehicle passing the area A can be denoted as a path in the Markov chain.

Definition 2 (vertex visiting probability (VVP)). π_{ij} is the probability of a node that moves from vertex i to j within a given maximal length δ. It is computed as

$$\pi_{ij}^{\delta} = \sum_{h=1}^{\delta} \pi_{ij}(h), \tag{2}$$

where $\pi_{ij}(h)$ is the probability of a node move from vertex i to j within the exact length h. Let $\mathscr{M}$ denote the transition matrix of mobility graph, then we have

$$\pi_{ij}(h) = \mathscr{M}^{h}[i][j], \tag{3}$$

where $\mathscr{M}^{h}$ is the h-step transition probability, which can be computed as the hth power of the transition matrix $\mathscr{M}$.

Definition 3 (set visiting probability (SVP)). $\lambda_{i\mathscr{R}}$ is defined as the probability with which a node moves from vertex i to at least one vertex $j \in \mathscr{R}$. It is derived as follows:

$$\lambda_{i\mathscr{R}} = 1 - \prod_{j \in \mathscr{R}} \left(1 - \pi_{ij}^{\delta}\right). \tag{4}$$

5.2. MRDP and Its Complexity. Based on the above definitions, we transform the minimum RSU deployment problem to a problem of selecting vertex subset. The formulation of MRDP is formulated as follows. Given a mobility graph G

Input: G—Mobility Graph; δ—maximal path length; P_u—meeting probability specified by user
Output: $\widehat{\mathscr{V}}$—result set being searched
(1) $\mathscr{M} \leftarrow$ transition matrix of G
(2) compute the 1-step, 2-step, ..., δ-step transition matrix of $\mathscr{M}$
(3) compute the VVP π_{ij}^{δ} between all nodes according to (2)
(4) $\widehat{\mathscr{V}} \leftarrow \emptyset$
(5) $\mathscr{S} \leftarrow \emptyset$
(6) **while** $|\mathscr{S}| < |\mathscr{V}(G)|$ **do**
(7) $\quad j \leftarrow \arg\max_{x \in \mathscr{V} - \widehat{\mathscr{V}}}(|\{k \mid k \in \mathscr{V} - \mathscr{S}, \lambda_{k(\widehat{\mathscr{V}} \bigcup \{x\})} \geq P_u\}|)$
(8) $\quad \widehat{\mathscr{V}} \leftarrow \widehat{\mathscr{V}} \bigcup \{j\}$
(9) $\quad \mathscr{S} \leftarrow \mathscr{S} \bigcup \{k \mid \lambda_{k\widehat{\mathscr{V}}}\} \geq P_u\}$
(10) **end while**

Algorithm 1: *RoadGate* deployment algorithm.

modeling the statistical mobility pattern over the area A and the meeting probability threshold P_u specified by users, the objective of MRDP is to find the smallest subset $\widehat{\mathscr{V}} \subseteq \mathscr{V}(G)$ such that the SVP from any vertex to $\widehat{\mathscr{V}}$ is not smaller than the probability specified by user, say,

$$\begin{aligned} &\text{minimize} \quad \left|\widehat{\mathscr{V}}\right| \\ &\text{s.t.} \quad \widehat{\mathscr{V}} \subseteq \mathscr{V}(G), \quad \forall i \in \mathscr{V}(G), \quad \lambda_{i\widehat{\mathscr{V}}} \geq P_u. \end{aligned} \tag{5}$$

We have the following theorem regarding the complexity of the MRDP.

Theorem 4. *The MRDP problem is NP-complete.*

Proof. The MRDP problem can be reduced to the classical *minimum vertex cover* problem which is a well-known NP-complete problem. First, for each vertex $i \in \mathscr{V}(G)$, we compute its VVS to all vertex j, π_{ij}^{δ} according to (2). Then, we find a vertex subset $\mathscr{X}_i = \{j \mid j \in \mathscr{V}(G) \text{ and } \pi_{ij}^{\delta} \geq P_u\}$. It contains all reachable vertices from vertex i within the constraint of given path length and the visiting probability specified by user. By repeating the process, we can compute the above set for each starting vertex i, say, $\mathscr{X}_i \mid i \in \mathscr{V}(G)$. Then, we construct a set for each vertex i, $Y_i = \{j \mid i \in \mathscr{X}_j\}$, containing those starting vertices from which a node can visit the vertex i within the constraint of the given path length and meeting probability threshold. Finally, the MRDP is equivalent to the problem of finding a subset $\widehat{\mathscr{V}}$,

$$\begin{aligned} &\text{minimize} \quad \left|\widehat{\mathscr{V}}\right| \\ &\text{s.t.} \quad \bigcup_{i \in \widehat{\mathscr{V}}} Y_i \supseteq \mathscr{V}(G). \end{aligned} \tag{6}$$

Obviously, the formulation is the classical *minimum vertex cover problem*, which has been shown as NP-complete. Note that we just consider the visiting probability from all vertices to a single vertex in the $\widehat{\mathscr{V}}$ instead of the SVP, which usually is greater than the former. However, this point cannot affect the correctness of the proving procedure because it just equivalent to that the meeting probability specified by user P_u is set to a smaller value. Consequently, MRDP is still an NP-complete problem. □

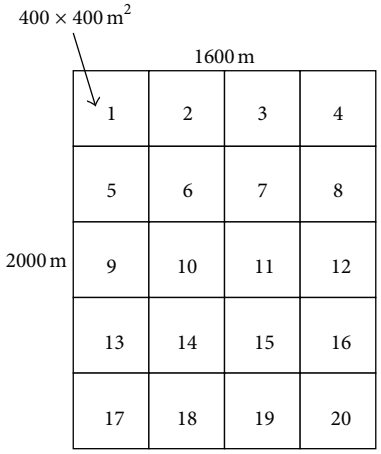
5.3. RoadGate Algorithm. In order to solve the MRDP problem, we develop a heuristic algorithm *RoadGate* which uses the greedy strategy to search optimal RSU deployment. The details of the algorithm are shown in Algorithm 1.

In this algorithm, the first 3 lines are responsible for computing the VVP π_{ij}^{δ} from any vertex i to other vertex j. The vertex set being searched is initialized in line 4. In line 5, the algorithm initializes a set $\mathscr{S}$ from which the SVP to the set being searched $\widehat{\mathscr{V}}$ is not smaller than the probability specified by user. Line 6 shows the following procedure terminates until all SVP from all vertices to the result set $\widehat{\mathscr{V}}$ are not smaller than predefined probability. Line 7, the key idea of *RoadGate*, searches greedily the vertex which can maximize the number of vertices whose SVP to the result set is not smaller than the predefined threshold. The found vertex is added the result vertex set in line 8. Finally the set $\mathscr{S}$ is updated after the new vertex is added. Clearly, the time complexity of this algorithm is $O(|\mathscr{V}|^3)$.

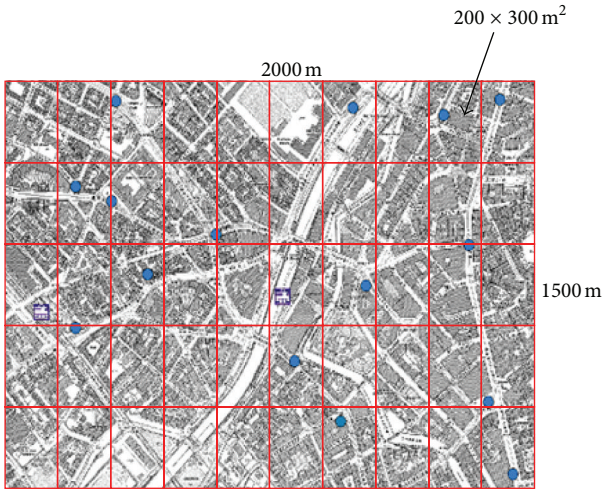
6. Performance Evaluation

6.1. Methodology. In this section, we evaluate the performance of *RoadGate* algorithm in two different scenarios. The first is a synthetic scenario shown in the left side of Figure 5. A $1600 * 2000\ \text{m}^2$ area is divided into 20 zones, each with $400 * 400\ \text{m}^2$. Thus, the corresponding mobility graph contains 21 vertices. The largest degree of each vertex is 4 because the vehicles only move from current zone to 4 neighboring zones. The average path length is $\delta = 5$ zones when a vehicle passes the whole area. The experiment is conducted for 200 runs. Each of those generates randomly the mobility graph with different random seeds.

Another experiment scenario is the realistic vehicle trace used in Section 4.1. We choose a central city area of $2000 \times 1500\ \text{m}^2$ in Zurich. It is divided into 50 zones each with $200 \times 300\ \text{m}^2$. Then, we use the approach proposed in Section 4.1 to compute the statistical transition probability and then

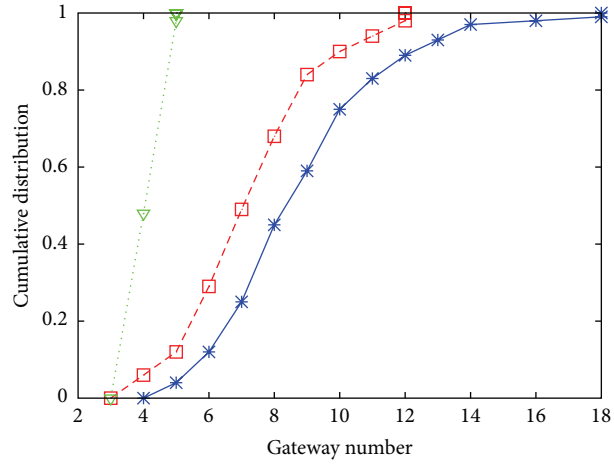


(a) The synthetic scenario

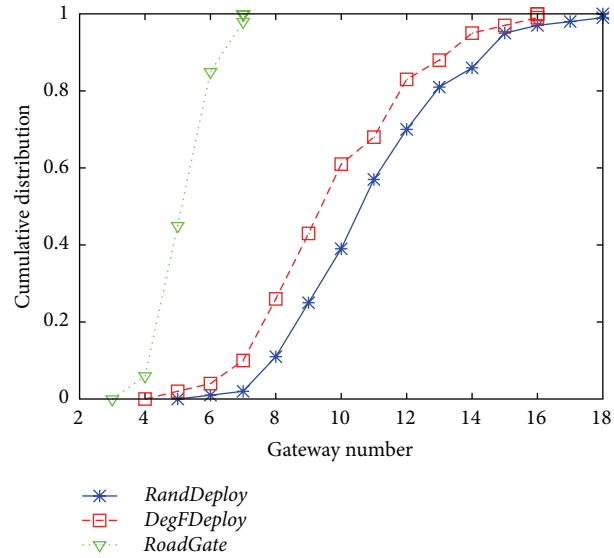


(b) The realistic scenario

FIGURE 5: Two scenarios for performance evaluation.



(a) P_u = 70%



(b) P_u = 90%

FIGURE 6: The cumulative distribution of required RSUs.

construct the corresponding mobility graph. Moreover, we also choose a larger area including both two scenarios to simulate the vehicles that enter and leave the scenarios.

We compare *RoadGate* with other two baseline algorithms. The first is the *Random Deployment (RandDeploy)*, which selects randomly a vertex to be added the result set until the SVP of all vertices to the result set is not smaller than the predefined threshold. The second is the *Degree First Deployment (DegFDeploy)*. Contrary to the *RandDeploy*, it chooses greedily the vertex with the largest degree to be added to the result set. The higher the degree of a vertex is, the more popular the corresponding zone is. Thus, *DegFDeploy* captures the stationary statistical pattern of the mobility in the target area in contrast to the *RoadGate*.

The following performance metrics are evaluated. (1) The number of required RSUs indicates the deployment cost which is the key metric of our system. (2) Actual meeting probability. It represents the achieved coverage performance when the vehicles go through the above scenarios. We implement the deployment solution in the simulator ONE (opportunistic network environment) [24] and drive the nodes to move to count the realistic meeting probability.

6.2. The Number of Required RSUs. We firstly compute the number of required RSUs when the meeting probability specified by user is 70% and 90% in the synthetic scenario. Figure 6 shows the cumulative distribution of 200 experiment results. It can be seen that *RoadGate* requires much less RSUs than other algorithms. When the expected meeting probability of users is 70%, 5 RSUs are needed for *RoadGate*, but *RandDeploy* and *DegFDeploy* need nearly 8 RSUs in most cases. Similarly, *RoadGate* also outperforms the other two algorithms when the meeting probability specified by user is 90%. It can be explained as follows. *RandDeploy* blindly

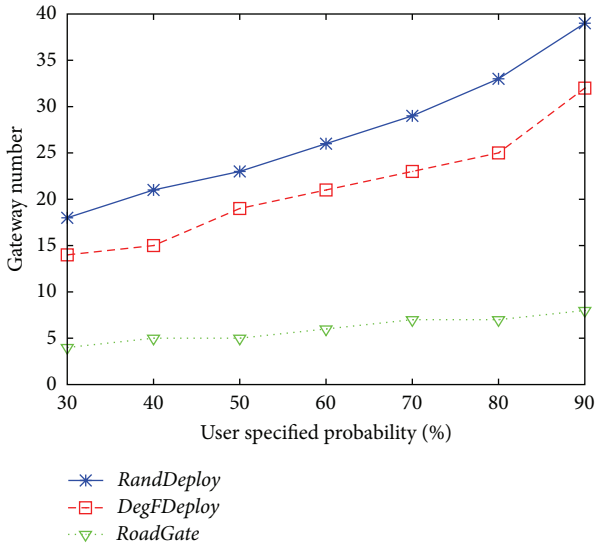


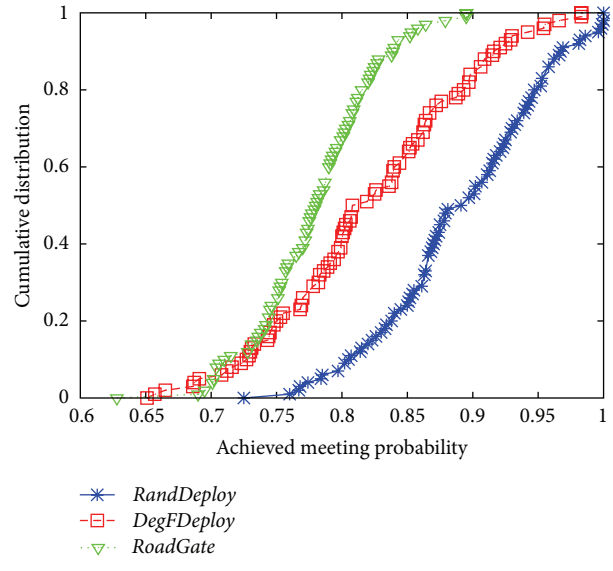
Figure 7: Number of required RSUs in the realistic scenario.

chooses the placement zones thus achieves and the worst performance. *DegFDeploy* only uses the coarse statistical information of the mobility pattern in the area. Thus, it has the poor performance in the random mobility graph. *RoadGate* utilizes the fine-grained statistical characteristic of mobility and can select the optimal places to install RSUs.

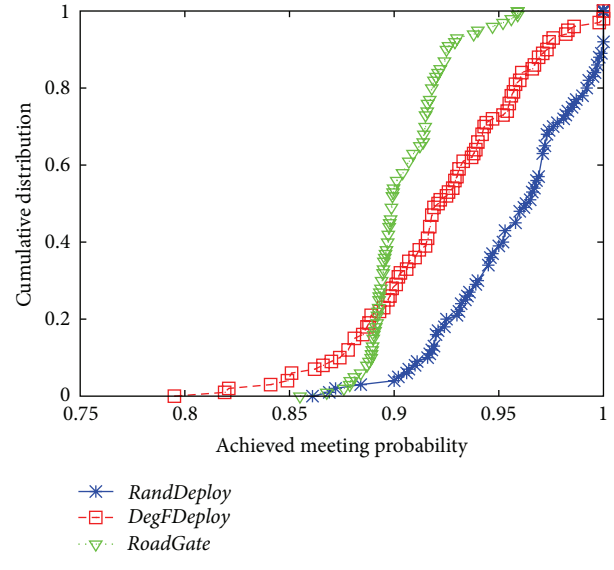
Only one deterministic mobility graph can be generated from the realistic vehicle traces. Therefore, we vary the meeting probability specified by user to observe its impact on the required number of RSUs in the three algorithms. As shown in Figure 7, the higher the meeting probability specified by users, the larger the number of the required RSUs. Moreover, our algorithm *RoadGate* always outperforms the two baseline algorithms. Since *RoadGate* can take full advantage of the coverage capability of each added RSU, it needs to add a few RSUs to fulfil the increase of users' expected meeting probability.

6.3. Actual Meeting Probability. We also evaluate the achieved meeting probability when vehicles go through the experiment scenarios. In the synthetic scenario, we compute the RSU deployment solution, respectively, by using, three algorithms for the meeting probability specified by users, which is 70% and 90%. In the 200 experiments in the synthetic scenario, we place 1000 vehicles in the experiment area and let them move according to the generated mobility graph. The RSUs are deployed according to the result computed by the three algorithms. When a vehicle moves into the zone with RSU, it succeeds in communication with the RSU. The actual meeting probability is computed as the ratio of the number of the vehicles meeting RSUs to total vehicle number. The cumulative distribution of 200 experiment results is shown in Figure 8.

As can be seen, all three algorithms succeed in meeting the meeting probability requirement of users. However, in contrast to *RandDeploy* and *DegFDeploy*, the real meeting



(a) $P_u = 70\%$

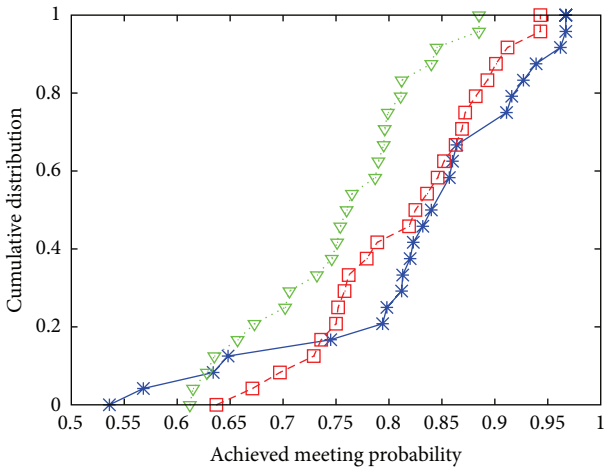


(b) $P_u = 90\%$

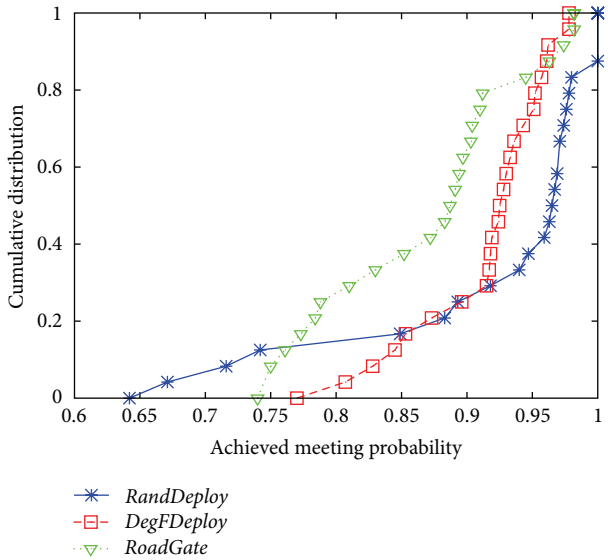
Figure 8: The cumulative distribution of actual meeting probability in synthetic scenario.

probability achieved by our *RoadGate* fairly matches the expected probability. Most of its results fall in the interval of 70%–80% when the specified probability is 70%. It shows that *RoadGate* is capable of selecting accurately the deployment places to meet the coverage requirement.

We measure the actual meeting probability in the realistic scenario. Similarly, we compute the deployment solution by using the three algorithms for the user's specified probability, which is 70% and 90%. Then, we make use of the solution to place the RSU in the scenario in the ONE simulator. These RSUs broadcast a beacon packet periodically. The experiments are carried out for 24 runs. Each run uses the extracted realistic vehicles trace lasting for 20 minutes from the total 8 hours peak time. These traces are fed into the



(a) $P_u = 70\%$



(b) $P_u = 90\%$

FIGURE 9: The cumulative distribution of actual meeting probability in realistic scenario.

simulator to drive the vehicles move. The actual meeting probability is calculated as the ratio of the number of the vehicles receiving the beacon to the total vehicle number in current run. The cumulative distributions of actual meeting probability are shown in Figure 9.

It can be observed from the above figures that the actual meeting probability achieved by the three algorithms basically varies in a large range and failes to meet the users' specified performance threshold. For example, when the expected meeting probability is 90%, around 40% probability achieved by *RoadGate* is smaller than the threshold. However, our *RoadGate* still outperforms *RandDeploy* and *DegFDeploy*. The reasons are described as follows. The analysis result of Section 4.1 shows that the transition probability between a same pair of zones is probably different for different time periods in a day. However, the mobility graph is constructed by using the average transition probability, causing probably a considerable deviation. Thus, the three algorithms running on the graph are hard to fulfill accurately the users' performance requirement. We also consider some possible strategies to relieve the problem. For example, we build an individual mobility graph for each possible transition probability of an edge. Then, we run the algorithm in each graph to get a deployment solution and then combine them as the final solution. Its essence is to meet the expected coverage performance by adding more RSUs.

7. Conclusions

In this paper, we study the problem of deploying RSUs for mobile vehicles in the vehicle-to-roadside communication system. Due to the limited transmission range and high deployment and maintenance cost, it is difficult to decide how many and where the RSUs should be placed. The objective of our study is to satisfy the connectivity requirement for all vehicles passing the coverage region. At the same times, the deployment cost such as the number of RSUs must be minimized. Our solution *RoadGate* uses the time-stable statistical mobility pattern observed in the realistic vehicle traces to find the optimal installation places. We propose a graph model to characterize this pattern and show that the RSU deployment problem is NP-complete by reducing it the minimum vertex coverage problem. Finally, we propose a heuristic greedy algorithm *RoadGate* to find the optimal installation places. Extensive experiments in the synthetic and realistic scenarios are carried out to evaluate the performance of our solution. The results show that our solution achieves the desired coverage performance and minimizes the number of the required RSUs.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grants no. 61202436 and 61271041) and the National Basic Research Program of China (973Program) (Grant no. 2009CB320504).

References

[1] J. Zhu and S. Roy, "MAC for dedicated short range communications in intelligent transport system," *IEEE Communications Magazine*, vol. 41, no. 12, pp. 60–67, 2003.

[2] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, "Relays, base stations, and meshes: enhancing mobile networks with infrastructure," in *14th Annual International Conference on Mobile Computing and Networking (MobiCom '08)*, pp. 81–91, September 2008.

[3] J. Ott and D. Kutscher, "Drive-thru internet: IEEE 802.lib for "automobile" users," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 362–373, March 2004.

[4] V. Navda, A. P. Subramanian, K. Dhanasekaran, A. Timm-Giel, and S. Das, "MobiSteer: using steerable beam directional

antenna for vehicular network access," in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services (MobiSys '07)*, pp. 192–205, June 2007.

[5] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan, "Interactive wifi connectivity for moving vehicles," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 427–438, 2008.

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

[7] B. Hull, V. Bychkovsky, Y. Zhang et al., "CarTel: a distributed mobile sensor computing system," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys '06)*, pp. 125–138, November 2006.

[8] C. Lochert, B. Scheuermann, M. Caliskan, and M. Mauve, "The feasibility of information dissemination in vehicular ad-hoc networks," in *Proceedings of the 4th Annual Conference on Wireless on Demand Network Systems and Services (WONS '07)*, pp. 92–99, January 2007.

[9] Z. Zheng, P. Sinha, and S. Kumar, "Alpha coverage: Bounding the interconnection gap for vehicular internet access," in *Proceedings of the 28th Conference on Computer Communications (INFOCOM '09)*, pp. 2831–2835, April 2009.

[10] Z. Zheng, Z. Lu, P. Sinha, and S. Kumar, "Maximizing the contact opportunity for vehicular internet access," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, March 2010.

[11] P. Li, X. Huang, Y. Fang, and P. Lin, "Optimal placement of gateways in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6 I, pp. 3421–3430, 2007.

[12] J. Lee and C. M. Kim, "A roadside unit placement scheme for vehicular telematics networks," in *Advances in Computer Science and Information Technology*, vol. 6059 of *Lecture Notes in Computer Science*, pp. 196–202, Springer, 2010.

[13] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, and M. Mauve, "Data aggregation and roadside unit placement for a vanet traffic information system," in *Proceedings of the 5th ACM International Workshop on VehiculAr Inter-NETworking (VANET '08)*, pp. 58–65, September 2008.

[14] A. Kchiche and F. Kamoun, "Centrality-based Access-Points deployment for vehicular networks," in *Proceedings of the 17th International Conference on Telecommunications (ICT '10)*, pp. 700–706, April 2010.

[15] O. Trullols, M. Fiore, C. Casetti, C. F. Chiasserini, and J. M. Barcelo Ordinas, "Planning roadside infrastructure for information dissemination in intelligent transportation systems," *Computer Communications*, vol. 33, no. 4, pp. 432–442, 2010.

[16] Y. Sun, X. Lin, R. Lu, X. Shen, and J. Su, "Roadside units deployment for efficient short-time certificate updating in VANETs," in *Proceedings of IEEE International Conference on Communications (ICC '10)*, pp. 1–5, May 2010.

[17] A. Abdrabou and W. Zhuang, "Probabilistic delay control and road side unit placement for vehicular ad hoc networks with disrupted connectivity," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 1, pp. 129–139, 2011.

[18] I. Filippini, F. Malandrino, G. Dan, M. Cesana, C. Casetti, and I. Marsh, "Non-cooperative RSU deployment in vehicular networks," in *Proceedings of the 9th Annual Conference on Wireless On-Demand Network Systems and Services (WONS '12)*, pp. 79–82, January 2012.

[19] J. Ghosh, S. J. Philip, and C. Qiao, "Sociological orbit aware location approximation and routing in MANET," in *Proceedings of the 2nd International Conference on Broadband Networks (BROADNETS '05)*, pp. 688–697, October 2005.

[20] C. Liu and J. Wu, "Routing in a cyclic mobispace," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 351–360, May 2008.

[21] M. Piorkowski, "Mobility-centric geocasting for mobile partitioned networks," in *Proceedings of the 16th IEEE International Conference on Network Protocols (ICNP '08)*, pp. 228–237, October 2008.

[22] V. Naumov, R. Baumann, and T. Gross, "An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces," in *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '06)*, pp. 108–119, May 2006.

[23] B. Raney, A. Voellmy, N. Cetin, M. Vrtic, and K. Nagel, "Towards a microscopic traffic simulation of all of switzerland," in *Proceedings of the International Conference on Computational Science (ICCS '02)*, vol. 2329 of *Lecture Notes in Computer Science*, pp. 371–380, Springer, London, UK, 2002.

[24] A. Keranen, J. Ott, and T. Karkkainen, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques (ICST '09)*, pp. 1–10, May, 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 518518, 6 pages
http://dx.doi.org/10.1155/2013/518518

Research Article

An Energy-Efficient Multisite Offloading Algorithm for Mobile Devices

Ruifang Niu,[1] Wenfang Song,[2] and Yong Liu[1]

[1] *School of Electronic and Information Engineering, Henan University of Science and Technology, Luoyang 471023, Henan, China*
[2] *School of Computer Science and Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Wenfang Song; iewenfangsong@163.com

Received 10 January 2013; Accepted 20 February 2013

Academic Editor: Jianwei Niu

Computation offloading is a popular approach for reducing energy consumption of mobile devices by offloading computation to remote servers. Most of the prior work focuses on a limited form of offloading part of computation from a mobile device to a single server. However, with the advent and development of cloud computing, it is more promising for the mobile device to reduce energy consumption by offloading part of computation to multiple remote servers/sites. This paper proposes an Energy-Efficient Multisite Offloading (EMSO) algorithm, which formulates the multiway partitioning problem as the 0-1 Integer Linear Programming (ILP) problem. Moreover, our proposed EMSO algorithm adopts the multi-way graph partitioning based algorithm to solve it. Experimental results demonstrate that our algorithm can significantly reduce more energy consumption as well as execution time and better adapt to the unreliability of wireless networks (such as the network bandwidth changes), compared with the existing algorithms.

1. Introduction

With the development of cloud computing, mobile devices have the potential to become powerful tools for information access and mobile application. Nowadays, it has become the primary computing platform for many users who expect their mobile devices, such as smart phones, to run sophisticated applications. However, the limited battery life is still a big obstacle for the further growth of mobile devices [1]. Several known power-conservation techniques [2, 3] include turning off the mobile computing devices screen when not used, optimizing I/O, and slowing down the CPU, among others. Although exponential improvements have occurred in hardware components, such developments have not come up in battery technology, and we cannot anticipate any significant changes in this field in the near future. Therefore, prolonging the battery life of mobile devices has become one of the top challenges.

One popular technique to reduce the energy consumption for mobile devices is computation offloading [4] or cyber forging [5], which means that parts of an application execute on the remote servers, with results communicated back to the local device. Most of existing work is often limited and restricted to the form of offloading computation from a mobile device to a single server [4–9]. Such scheme cannot adapt well to the cloud environment [10]. Since it is difficult to directly program the cloud-enabled application, an alternate scheme is to write a monolithic application and then automatically partition it between the mobile device and the remote sites [11, 12]. Therefore, in this paper we propose an Energy-Efficient Multisite Offloading (EMSO) algorithm, which focuses on offloading parts of an application from a mobile device to multiple remote sites. Figure 1(a) shows a monolithic mobile application totally running on the resource-limited mobile device without computation offloading, and Figure 1(b) shows the case of distributed application execution by offloading parts of the codes from the mobile device to remote servers.

Computation offloading is confronted with several key challenges. Firstly, in a multisite offloading scenario, the unreliability of the wireless network (e.g., bandwidth often changes dynamically) affects the feasibility and efficiency of computation offloading for mobile devices. Existing work relies on programmers to modify the program to deal

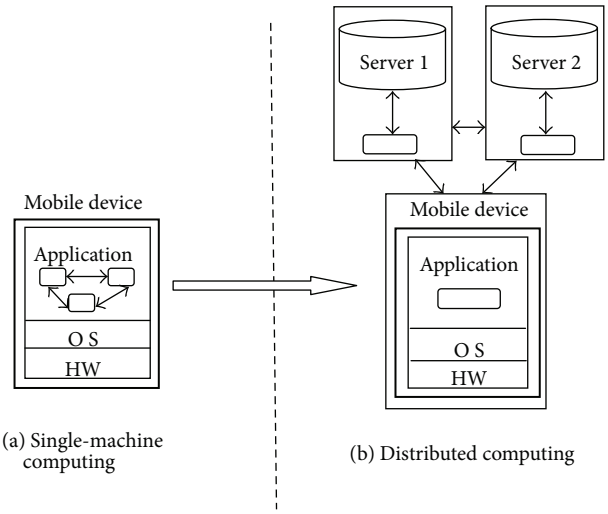


Figure 1: (a) Monolithic mobile application running on a mobile device. (b) Distributed execution by computation offloading between a mobile device and two sites.

with partitioning, state migration, and even the changes in network conditions [4, 5]. Although it can save more energy for mobile devices, it increases the additional burden for programmers and cannot adapt well to network environment changes. Secondly, the granularity of offloading must be chosen appropriately. Most prior work performs offloading at the class level [6, 9, 11], which does not allow objects of the same class to be offloaded to different servers, resulting in poor partitioning performance.

This paper describes a novel approach motivated by the idea in [8] to tackle these challenges. In this paper, we introduce a multiway partition algorithm, which models the bandwidth as a random variable to better adapt to the bandwidth changes of wireless networks and allows a program to be partitioned between multiple sites. Moreover, based on the Weight Object Relation Graph (WORG) constructed by using static analysis and dynamic profiling techniques [7], we accomplish the computation offloading for a given monolithic application at the object level to perform more efficient partitioning than that at the class level. Experimental results demonstrate that our algorithm can significantly reduce energy consumption with automatic adjustment to different network conditions.

The rest of this paper is organized as follows. Section 2 provides a detailed description of our multisite offloading scheme. Experiment and analysis are presented in Section 3, followed by some concluding remarks in Section 4.

2. Energy-Efficient Multisite Offloading Algorithm

Figure 1 in Section 1 shows the multisite offloading model. It shows that, with computation offloading, a distributed application execution will be partitioned between the mobile device which must contain at least one execution module such as the user interface and one or more servers which can be used for computation offloading in order to improve the execution or reduce energy consumption for the mobile device.

Normally, determining which portions of a computation to offload is cast as a graph partitioning problem. Our proposed Energy-Efficient multisite Offloading (EMSO) models the program to be partitioned as a Weight Object Relation Graph (WORG), with nodes representing the computation module (a run time object of the application), and edges representing the interaction between modules (e.g., invocations between one object and another). In a WORG, the weight of an edge indicates communication costs (in power) of the interaction between two modules, while the weight of a node represents the computation power consumption of the object module. The goal of this paper is to minimize the energy/power consumption by computation offloading.

The total costs of the partitioning can be calculated by considering both the weights of edges for communication and the weights of nodes for computation to get the best tradeoff. The optimal partitioning scheme means the optimal choice of modules to offload [13]. The next section formalizes such problems, giving an Integer Linear Programming (ILP) formulation of the multi-site offloading problem.

2.1. Graph Construction. As for the aforementioned weights of nodes and edges of WORG, they can be estimated by either static analysis or profiling of the program. EMSO first apply constructs the initial ORG of the application by using the Soot analysis framework [14] to perform the static points to analysis. And then offline profiling [7] is performed to assign weights to the nodes and edges of the ORG to construct the WORG. Figure 2 shows a WORG which we construct by both the static analysis and offline profiling methods for an application.

2.2. Problem Formulation. Our goal is to partition a graph WORG $= (V, E)$, with vertices set V and edges set $E \in V \times V$, and a set of $k + 1$ partitions denoted as $P = \{p_0, p_1, \ldots, p_k\}$ (p_0 represents the mobile device, and $p_1, \ldots, p_k$ represent the offloading sites, k is the number of offloading sites). As shown in Figure 2, the weight of the vertex v is described as a 2-tuple $\langle t_c(v), t_s(v)\rangle$, where $t_c(v)$ indicates the CPU execution time for each object running on the client, and $t_s(v)$ is that for each object running on the server. $t_s(v)$ can be calculated by $t_c(i)/k$, where k indicates that the server is k times faster than the mobile device. Each edge $e_{v1,v2}$ is associated with a weight $\langle s(v1, v2)\rangle$ indicating the amount of the total data that need to be transmitted between two nodes. EMSO collects $\langle t_c(v), t_s(v)\rangle$ and $\langle s(v1, v2)\rangle$ metrics by offline Profiling [7] during the WORG construction described in Section 2.1.

We can formulate the multiway partitioning problem as the 0-1 ILP problem. Our goal is to minimize the energy consumption, that is, the value of the following objective function:

$$\text{Energy (WORG)} = \sum_{v \in V} \left(E\left(v_l\right) \cdot x_l + E\left(v_s\right) \cdot x_s\right) + \sum_{v1 \in V,\, v2 \in V} \left|x_l - x_s\right| \cdot E\left(e_{v1,v2}\right), \quad (1)$$

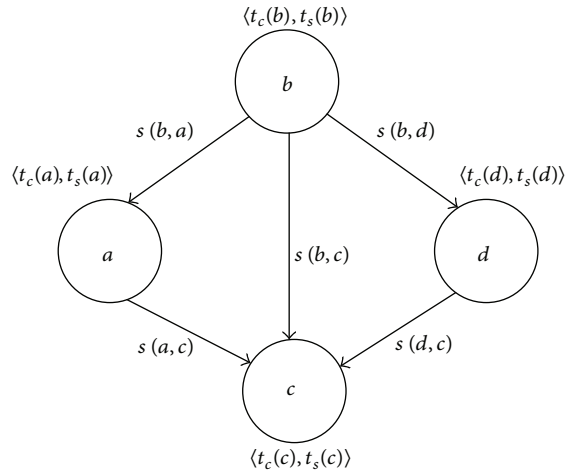


Figure 2: WORG of an application.

Input: $WORG = (V, E)$, B, b, N_L, a
Output: *Xmin*-the optimal partitioning scheme,
MinEnergy-the minimal energy consumption
(1) Compute the minimum energy consumption when bandwidth = b using the Stoer-Wagner algorithm, noted as *minE*
(2) $minE = min\ E^*(1 + a)$
(3) For v_i in V
(4) If v_i in N_L
(5) $X[i] = 1$; // vertexes running on the client
(6) Else
(7) $X[i] = -1$; //vertexes to be partitioned
(8) End if
(9) End for
(10) DFSearch(1, *minE*, *WORG*, *X*, *Xmin*, *MinEnergy*)
(11) Return {*Xmin*, *MinEnergy*}

Algorithm 1: Graph partitioning based algorithm.

where x_l and x_s indicate the assignment of each node: $x_l = 1$, $x_s = 0$ if vertex v_l is assigned to the client and v_s is assigned to servers, $x_l = 0$, $x_s = 1$ otherwise. Equation (1) is subject to the following constraint:

$$\forall v \in V : x_l + x_s = 1. \tag{2}$$

$E(v_l)$ and $E(v_s)$ are the energy consumption of vertex v running on the client and the server, respectively. They can be computed through the following (3):

$$\begin{aligned} E(v_l) &= P_c \times t_c(v), \quad v \in P_0, \\ E(v_s) &= P_s \times t_s(v), \quad v \in P_1 \ldots P_k, \end{aligned} \tag{3}$$

where $t_c(v)$ and $t_s(v)$ are the weights of vertex v when running on the client and on the servers, respectively. P_c and P_s are the power CPU of the client or the servers.

$E(e_{v1,v2})$ is the energy consumption for data transmission between vertex $v1$ and vertex $v2$ when they are not running on the same site, for example, one running on the client and the other on the servers. $E(e_{v1,v2})$ is computed by (4):

$$E(e_{v1,v2}) = \frac{s(v1, v2)}{b} \times P_{\text{wi-fi}}, \tag{4}$$

where $s(v1, v2)$ is the weight of the edge between vertex $v1$ and $v2$. b indicates the network bandwidth and $P_{\text{wi-fi}}$ is the power of the wireless Wi-Fi network interface.

To minimize the value of (1), the key is to determine the value of x_l and x_s, that is, 0 or 1. As remote servers usually executed much faster than mobile devices with powerful configuration, it can save energy and improve execution to offload part of computation to servers. However, when vertexes are assigned to different sites, the interaction between them leads to communication cost. Therefore, our problem formulation aims at the optimal assignment of vertexes for graph partitioning and computation offloading by trading off computation costs and communication costs.

2.3. Partitioning/Offloading Algorithm. We perform a multiway graph partitioning based algorithm to solve the ILP problem. First, we transform the WORG to a Directed Acyclic Graph (DAG) and perform the topologic sort. Then, we use the depth-first search to traverse the search tree and compute the $Energy(G)_B$ and $Energy(G)_b$ for each encountered nodes, where B is the current bandwidth, and b is the critical bandwidth that meets $P\{B \geq b\} > P_c$. P_c is the guaranteed probability, and $Energy(G)_b$ represents the energy consumption of the particular partitioning scheme when bandwidth is b. During the search, if $Energy(G)_b$ does not fulfill the constraints or $Energy(G)_B$ is larger than the current minimal energy (*MinEnergy*), that is, $Energy(G)_B > MinEnergy$, the subtree of the node will be cut and it traverses back to the parent node to continue searching. After traversing the whole tree of the DAG, we will get the optimal partitioning that fulfills the given constraints. The partitioning algorithm is shown as Algorithm 1, where N_L is the node collection which runs locally on the client, and a is an empirical constant to set the constraint of the minimal energy. The DFsearch function is the depth-first search algorithm described as above shown.

3. Evaluation

3.1. Experiment Setup. This section presents the experimental results of our EMSO algorithm. We evaluate the performance of EMSO by comparing it with the No Application Partitioning (NAP) case and a Static Application Partitioning (SAP) algorithm [15]. The evaluation metrics are energy consumption and execution time. We perform the comparisons on three random graphs generated by certain schemes to simulate the real-world scenarios involving a client device and two remote servers. The dataset used in experiments are listed in Table 1 as follows.

We suppose that the application is initially located on a mobile device and the bandwidth varies between the value of 10 kb/s and 100 kb/s. Other parameters, such as power consumption rates, are set as $P_i = 1.7W, P_c = 2.6W$, and

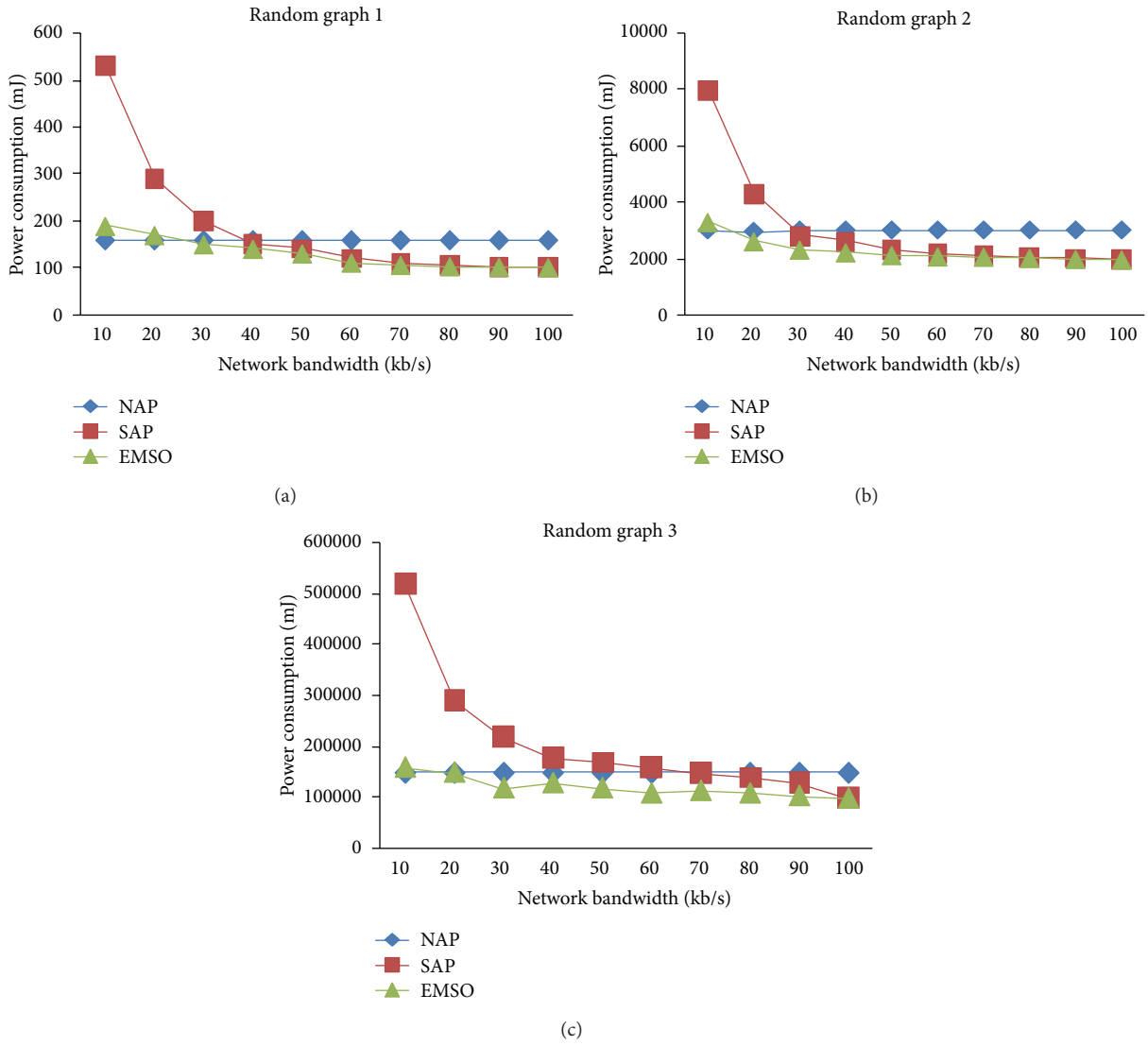


Figure 3: Energy (power) consumption comparisons of different algorithms with network bandwidth variation.

Table 1: The size of graphs.

Random Graph	Number of nodes	Number of edges
Graph 1	15	60
Graph 2	30	350
Graph 3	100	3238

$P_{\text{wi-fi}} = 2.3W$. $k = 5$ indicates that servers execute 5 times faster than the mobile device.

3.2. Energy Consumption Evaluation. From (1), we can see that the communication cost (i.e., $E(e_{v1,v2})$) is critical to partitioning decision, and it is directly related with the network bandwidth. However, in most cases, the network bandwidth changes dynamically, especially in wireless networks of mobile devices. The bandwidth is considered as a variable to improve the dynamic of partitioning in our EMSO.

To evaluate the adaption of EMSO to bandwidth changes, we compare the energy consumption of three algorithms with bandwidth changes. The experimental results with bandwidth varying in steps of 10 kb/s are presented in Figure 3. As shown in Figure 3, from Random Graph 1 with fewer nodes and edges to Random Graph 3 with the most nodes and edges, the energy consumptions of all three approaches increase, because the computation become larger and more complex as the nodes and edges grow. NAP consumes the constant energy with increasing bandwidth because the whole application keeps running on the mobile device without offloading and without energy costs of communication. As the bandwidth changes between 10 kb/s and 100 kb/s, the energy consumption of SAP varies more severely than that of EMSO. When the bandwidth becomes lower, SAP still maintains the former partitioning scheme, resulting in great increase of communication costs. However, our EMSO algorithm can find the better partitioning assignment when network bandwidth changes. Particulary, when bandwidth >20 kb/s,

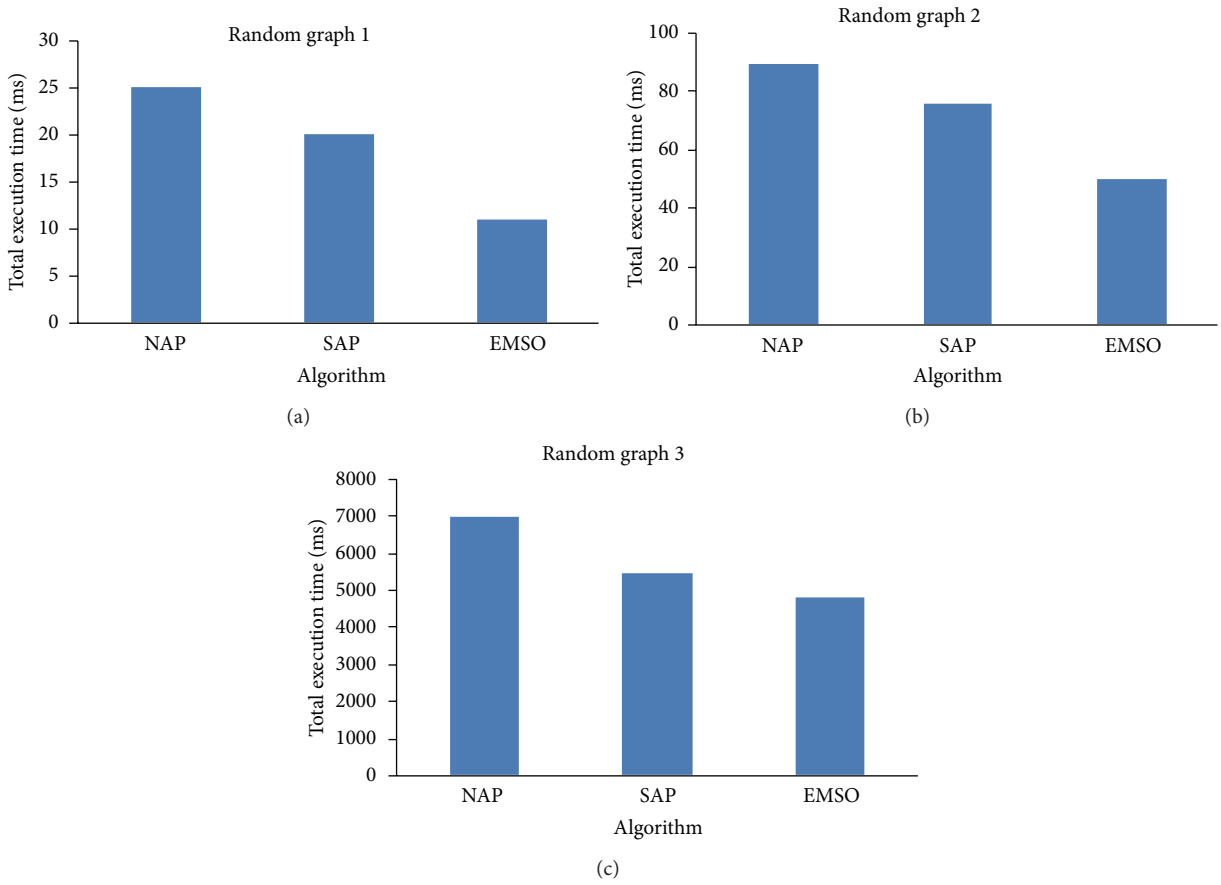


Figure 4: Total execution time comparisons of different algorithms.

EMSO algorithm saves about 25% energy compared to SAP. Meanwhile, when bandwidth >20 kb/s, our proposed EMSO algorithm also outperforms NAP due to partitioning approaches. The results demonstrate that EMSO is effective and beneficial to perform the partitioning for mobile devices.

3.3. Execution Time Evaluation. To further evaluate the performance of our EMSO algorithm, we estimate the total execution time of different algorithms as another evaluation metric to perform the partitioning in Figure 4. If it takes less time to execute a mobile application, it is beneficial for energy conversation of mobile devices and also improves the user experience with high-efficiency execution. As shown in Figure 4, we can see that our EMSO executes the computation offloading much faster than NAP without code offloading and faster than SAP with static partitioning, which demonstrates that, compared to NAP and SAP, our proposed EMSO can significantly improve execution time and reduce energy consumption for resource-restricted mobile devices. Besides, the result that execution time for Random Graph 3 is much larger than the other two (Random Graph 1 and Random Graph 2) also meets our expectations because of its more nodes and edges compared with the other two.

As a conclusion, as shown in Figure 3 on energy consumption and Figure 4 on total execution time, it is obvious that our proposed EMSO algorithm effectively saves the most energy and takes the least execution time to perform the partitioning, which is significantly beneficial for energy conservation of mobile devices.

4. Conclusion

This paper proposes an Energy-Efficient Multisite Offloading (EMSO) algorithm for computation offloading to save energy of mobile devices. This is a multi-site partitioning approach, which supports multiple differentiated offloading sites and assigns appropriate objects between mobile devices and servers dynamically to minimize energy consumption as the network bandwidth changes. EMSO models the application partitioning as a 0-1 ILP problem by using the multiway graph partitioning based algorithm to get the best tradeoff between computation costs and communication costs. With the constructed Weight Object Relation Graph (WORG), EMSO performs partitioning at the object level to achieve more precise offloading. Our evaluation demonstrates that EMSO is efficient in computation partitioning/offloading for mobile devices and outperforms the static algorithm in prior work with respect to both energy consumption and execution time.

Acknowledgments

This work was supported by the Research Fund of the State Key Laboratory of Software Development Environment

under Grant no. BUAA SKLSDE-2012ZX-17, the National Natural Science Foundation of China under Grant no. 61170296 and 61190120, and the Program for New Century Excellent Talents in University under Grant no. NECT-09-0028.

References

[1] D. Rakhmatov and S. Vrudhula, "Energy management for battery-powered embedded systems," *ACM Transactions on Embedded Computing Systems*, vol. 2, no. 3, pp. 277–324, 2003.

[2] K. Lahiri, S. Dey, D. Panigrahi et al., "Battery-driven system design: a new frontier in low power design," in *Proceedings of the Asia and South Pacific Design Automation Conference*, pp. 261–267, Bangalore, India, 2002.

[3] R. Rao, S. Vrudhula, and D. N. Rakhmatov, "Battery modeling for energy-aware system design," *IEEE Computer*, vol. 36, no. 12, pp. 77–87, 2003.

[4] Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: a partition scheme," in *Proceedings of the 4th ACM international Conference on Compilers, Architecture and Synthesis for Embedded systems (CASES '01)*, pp. 16–19, Atlanta, Ga, USA, 2001.

[5] R. Balan and J. Flinn, "The case for cyber foraging," in *Proceedings of the 10th ACM SIGOPS European European Workshop (Sigcom '02)*, pp. 160–165, Saint-Emilion, France, July 2002.

[6] N. Geoffray, G. Thomas, and G. Folliot, "Transparent and Dynamic Code Offloading for Java Applications," in *Proceedings of the OTM Confederated International Conferences CoopIS, DOA, GADA, and ODBASE, CO*, pp. 57–66, 2006.

[7] L. Wang and M. Franz, "Automatic partitioning of object-oriented programs for resource-constrained mobile devices with multiple distribution objectives," in *Proceedings of the 14th IEEE International Conference on Parallel and Distributed Systems (ICPADS '08)*, pp. 369–376, Melbourne, Australia, December 2008.

[8] E. Cuervoy, A. Balasubramanian, D. K. Cho et al., "MAUI: making smartphones last longer with code offload," in *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '10)*, pp. 49–62, San Francisco, Calif, USA, June 2010.

[9] K. Yang, S. Ou, and H. H. Chen, "On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 56–63, 2008.

[10] K. Kumar and Y. H. Lu, "Cloud computing for mobile users: can offloading computation save energy?" *Computer*, vol. 43, no. 4, Article ID 5445167, pp. 51–56, 2010.

[11] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[12] B. G. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution," in *Proceeding of the 8th Workshop on Hot Topics in Operating Systems (HotOs '09)*, pp. 100–109, Monte Verità, Switzerland, 2009.

[13] S. Han, S. Zhang, Y. Zhang, and J. Cao, "Dynamic software allocation algorithm for saving power in pervasive computing," *Journal of Southeast University*, vol. 23, no. 2, pp. 216–220, 2007.

[14] Soot, November 2012, http://www.sable.mcgill.ca/soot/.

[15] S. Ou, K. Yang, and A. Liotta, "An adaptive multi-constraint partitioning algorithm for offloading in pervasive systems," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '06)*, pp. 116–125, Pisa, Italy, March 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 762503, 7 pages
http://dx.doi.org/10.1155/2013/762503

Research Article

Pervasive Urban Sensing with Large-Scale Mobile Probe Vehicles

Yanmin Zhu,[1,2] Xuemei Liu,[1] and Yin Wang[3]

[1] *Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China*
[2] *Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai 200240, China*
[3] *HP Labs, Palo Alto, CA 94304, USA*

Correspondence should be addressed to Yanmin Zhu; yzhu@cs.sjtu.edu.cn

Received 26 October 2012; Accepted 9 January 2013

Academic Editor: Yan Zhang

With the advance of embedded sensing devices, Pervasive Urban Sensing (PUS) with probe vehicles is becoming increasingly practical. A probe vehicle is equipped with onboard sensing devices that detect urban information as the probe vehicle drive across the road network. For example, GPS sensors can detect real-time vehicle status including instant speed and physical position. PUS can provide the general public valuable urban sensing information, such as frequently updated digital maps, and real-time traffic light states. In this paper, we first present the framework of Pervasive Urban Sensing with probe vehicles. Next, we present two cases of urban sensing with probe vehicles. As case one, we discuss the design of a sensing algorithm for detecting the instant state of traffic lights. As case two, we discuss the design of sensing algorithms for recognizing roads by using the vehicular footprints. Some preliminary results of these two cases of urban sensing are presented and discussed.

1. Introduction

Governments and organizations have been engaged in providing convenient traveling experiences for citizens and drivers [1–5]. Besides convenience, reliability and security should also be guaranteed by Pervasive Urban Sensing (PUS). As a result, organizations and researches are creating and developing services such as digital map construction and update [6–9], traffic light optimization [1, 2], traffic flow detection [10], and so forth. Digital map construction and update are the foundation for most other services. The conventional method of digital map construction is based on geological survey, which is a time-consuming and expensive process. What is more, for the areas where new road networks are created or existed roads are reconstructed or closed, these digital maps may become outdated and not applicable for other services in PUS. Traffic light optimization is another important task in PUS, as drivers and passengers spend a large proportion of time waiting red lights. Thus, the switching history of traffic lights red-green status should be collected and optimization systems should be designed. Current way of collecting light history status is by field survey, which is laborious and not applicable to large-scale metropolis. Besides, traffic flow detection is the key component in navigation systems to provide reliable and optimal route scheduling for drivers.

Nowadays, an increasing number of vehicles are deployed with GPS devices for location detection and other status measurements. Taxis companies equip their taxis [11] with GPS devices for the requirement of supervisory control and scheduling. Apart from that, systems for collecting real-time taxis GPS traces and visualization projects should also be developed. For civilian vehicles, GPS devices are usually deployed together with the navigation systems. We collected GPS traces of about 4,000 taxis in Shanghai from March 2006 to May 2007. The GPS traces are coarse grained in terms of large sampling interval and inaccurate data sensory data.

It is probe vehicles equipped with GPS devices that make PUS possible and thriving. It is feasible to realize PUS with probe vehicles as urban activities or events especially those related to transportation will have direct effect on the movement status of probe vehicle. Through methodologies such as data analysis and data mining, these activities or events can be extracted from the GPS traces reported by probe vehicles.

There are several considerable advantages of urban sensing with probe vehicles. First, the expense will be reduced as the necessary elements of sensing system already exited, including the probe vehicles, and the data transmission and collection system. Second, the coverage of the sensing area is guaranteed to be large as the probe vehicles will traverse nearly all the road segments in the city. Third, the GPS traces reported by probe vehicles can be utilized in many sensing researches, including road map sensing, traffic flow sensing, traffic light status sensing, emergent accidents sensing, and others related to transportation.

This paper presents an overview to Pervasive Urban Sensing with probe vehicles, accompanied with two concrete examples, that is, traffic light sensing and road map sensing. Instead of presenting complete details, this paper gives a general introduction. In [6–9, 12, 13], map building using offline GPS trajectories with low error and high sampling frequencies have been studied. Gravitation and repulsion force in physical theory are utilized to do map detection in [12]. The authors in [6, 8] developed algorithms to detect new roads. In [10], the authors applied compressed sensing theory to reconstruct the citywide traffic flow status with limited GPS trajectories. Traffic light optimization algorithms and systems are developed in [1, 2, 4] with the help of optimization theory and heuristic algorithms.

The rest of the paper is organized as follows. Section 2 presents the main framework of PUS system and the characters and GPS traces generated by probe vehicles. In Sections 3 and 4, we introduce two urban sensing cases, traffic light sensing and road map sensing. The problem formulation and challenges are given together with the sensing algorithms. Section 5 concludes the paper with a discussion.

2. Preliminary

In this section we will present the general framework for PUS and analyze the characteristics of GPS traces generated by probe vehicles.

2.1. Urban Sensing Framework. The main framework of PUS is shown in Figure 1. Suppose there are N probe vehicles in the system, and each of them moves freely in the city and generates GPS report periodically. The generated GPS report at time t by vehicle i is a five-tuple, $r_i(t) : \langle id, v, a, p, s, t\rangle$, representing vehicle identification, instant velocity, headway direction, position (*longitude* & *latitude*), vacant or not, and timestamp when the data is reported. Suppose we collect the data from $T_{\min}$ to $T_{\max}$ and denote the set of GPS reports collected as $\Omega(T_{\min}, T_{\max}) = \sum r_i(t)$, $i \in \{1, 2, \ldots, N\}$, $T_{\min} \leq t \leq T_{\max}$.

The GPS reports are delivered to the PUS server via data channel of GSM/GPRS. Note that the quality of the data delivery channel will affect the quantity of GPS reports stored in the PUS sensing server. It is the PUS server that collects the real-time GPS reports and executes the Pervasive Urban Sensing tasks. After the real-time and reliable sensing results are produced, they will be made available to public through Internet.

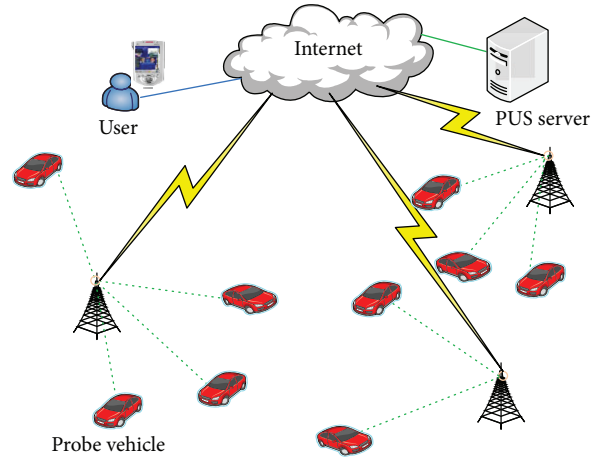


Figure 1: System architecture.

2.2. Analysis of Real Vehicular GPS Traces. The fleet of probe vehicles we utilize in the PUS system are taxis from several taxis companies, and the sampling duration is from March 2006 to May 2007. Before conducting urban sensing, it is necessary to analyze the GPS reports in order to get the overall distribution of the GPS reports and be clear about what characteristics of the data may lead to challenges in the process of urban sensing. Thus, we conduct statistical experiments with one week GPS data in 2007 and the results are shown below.

2.2.1. Sampling Interval. The purpose of equipping taxis with GPS devices is for supervisory control and scheduling, so it is not necessary for the sampling interval of taxis to be as frequent as 1 Hz, which is the usual sampling interval used in many traffic studies [6, 7]. Among all taxis that report nonstationary GPS coordinates, 1,855 sample at 16 sec when vacant and 61 sec when occupied (corresponding to SH-B group in [14]); 430 sample at a fixed interval 60-61 sec; the rest mostly sample by the distance traveled.

2.2.2. Speed. The speed of a vehicle is influenced by a number of factors. Three conclusions are got through analysis to the GPS data. First, there is a large probability for a vehicle to travel with low speed when facing traffic lights. Second, the average speed of a taxi when it is vacant is much slower than that when carry passengers as it usually just travels around slowly to catch passengers when vacant. Third, the average speed of a vehicle in peak hours is much slower than that in normal hours as the traffic condition is much better and traffic is more fluent in normal hours.

2.2.3. Resolution. The GPS coordinates reported include four fraction digits, that is, 0.0001 degree, which is 8.5 or 11.1 meters along latitudinal or longitudinal line in Shanghai, respectively. Resolution limit aside, GPS measurement noise can be modeled as a Gaussian distribution [15]. Different roads have different number of length. Finally, tunnels and high-rise buildings are dense in downtown Shanghai, which

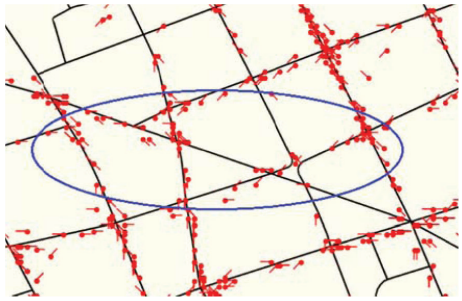


Figure 2: GPS reports and their heading directions on OpenStreetMap.

can result in significant noise [14]. Nevertheless, we found the GPS measurement accurate enough for road recognition. For example, 94% of records are within 100 meters to some road, among which 95% are within 38 meters to the nearest road [14]. Figure 2 shows GPS samples as red dots on the map. Most of these dots are located near roads.

2.2.4. Spatial Distribution. By mapping GPS records onto digital map, we find that the spatial distribution is uneven in the urban area. Hot spots effect is very evident. The frequency of collecting GPS reports in areas where big malls or subways exist is much larger than other places. This suggests that for the areas where a relatively small amount of taxis pass by, the time duration used for collecting GPS reports should be longer to guarantee the quality of PUS services.

3. Traffic Light Sensing

3.1. Background. The objective of traffic light sensing is to detect the states of traffic lights, which is very important in many researches and applications, such as traffic lights optimization [1, 4, 5], traffic management [2], and real-time vehicle navigation. It is required to have traffic light state in order to optimize traffic management [3]. A number of research projects [1, 2] about traffic light optimization are being carried out all around the world. To perform traffic lights optimization, the information of traffic light state is very important.

Vehicle networks [16, 17] have attracted more and more attentions as they are providing intelligence transportation services and Internet access [18]. Traffic light state information is very important in designing vehicle routing protocols for its impact on the mobility of vehicle. Moreover, efficient data delivery approaches can be discovered when traffic light state is available since a red traffic light may pause a traffic flow and create good connectivity for a certain duration [17, 19–21]. However, traffic light state has not been considered in existing vehicle mobility models due to the lack of state information of traffic lights.

Thus, traffic light sensing is very important since it is fundamental for many exciting applications. Few researches are related to traffic light sensing yet. One candidate approach to traffic light sensing is to deploy cameras at intersections and perform image processing to detect the states of traffic lights. However, the expense of this approach is unbearable thus the coverage will be limited. Furthermore, this approach is vulnerable to bad whether such as fog or rain.

In this section, we introduce the approach of traffic light sensing with probe vehicles, which achieves several advantages such as large coverage and low cost.

Several challenges are faced in traffic light sensing with probe vehicles. First, the periods of a traffic light are not fixed, but adaptive to the current traffic condition of the road segments attached to the intersection. To detect road traffic, loop detectors [22] are deployed beneath the road surface. Second, GPS reports for a traffic light are temporal discrete but the objective of traffic light sensing is to detect the state of a traffic light at any time. Third, the distribution of GPS reports in the city is uneven. Thus, the effect of traffic light sensing with probe vehicles on all the lights in the city should be investigated.

3.2. Problem Description. A traffic light is changing its state over time and the interval of state may be uncertain. We denote $\mathscr{M}$ as the set of traffic lights we are interested in, and $s_l(t) \in \{\text{red}, \text{green}\}$ as the real state of the traffic light l at time t. The objective of traffic light sensing is to estimate traffic light state over time: $\widehat{s}_l(t)$, for all $l \in \mathscr{M}, T_{\min} \le t \le T_{\max}$ with the GPS reports set $\Omega(T_{\min}, T_{\max})$.

Suppose we get a state estimate $\widehat{s}_l(t)$ for traffic light l at time t, then the estimation error of the estimate is as follows:

$$\xi_l(t) \triangleq \left| s_l(t) - \widehat{s}_l(t) \right|. \tag{1}$$

Then, the problem of traffic light sensing is to estimate traffic light states with the objective of minimizing the average estimation error rate ξ as follows:

$$\min \xi \triangleq \frac{1}{(T_{\max} - T_{\min}) \cdot |\mathscr{M}|} \sum_{l \in \mathscr{M}} \int_{T_{\min}}^{T_{\max}} \xi_l(t) \times dt \tag{2}$$

utilizing the GPS reports set $\Omega(T_{\min}, T_{\max})$.

We have the intuition that there is strong correlation between traffic lights and probe vehicles movements. The movements of a vehicle running in the city is regulated by traffic lights. We can easily find that the speed of a vehicle facing a red light is much smaller than that facing a green light. This is apparent since vehicles have to stop and wait for a red traffic light to turn green.

However, we also find that when the light is in red, there is a considerable percentage of nonzero vehicle speeds, and the percentage of zero speed is nonnegligible when the light is in green. This is understandable because immediately after the light turns from red to green, the vehicles have to spend a certain time to slow down before they fully stop. The reason is similar for the phenomenon of zero speed when the light is in green.

Moreover, when a vehicle is further from the traffic light, for example, 150 m away from the light, the mobility of the vehicle is relatively less related to traffic light state. This suggests that GPS reports generated far away from the lights should be neglected in traffic light sensing.

3.3. Detecting Light States. In this section we propose a novel algorithm for traffic light sensing. Two steps are executed in the algorithm, Snapshot State Estimation and Panoramic Static Estimation. The Snapshot State Estimation is to estimate the status of traffic lights one vehicle is facing at the moments GPS reports are generated. Panoramic Static Estimation is to estimate the continuous status of traffic lights citywide between $T_{\min}$ and $T_{\max}$. Traffic light sensing is formalized as an optimization problem and heuristic algorithms can be utilized to get the optimal result.

In Snapshot State Estimation, clustering models in machine learning are utilized to estimate the state of the traffic light $\widehat{s}_l(t)$ at the time instant when a GPS report was generated. In order to generate a clustering model, a sample set $\{r_i(t) : \langle id_i, v_i, p_i, t, l_i, s_l(t)\rangle\}$ should be available and we got it by field study. There are some clustering models that can be applied here, MAP [23], SVM [24], and so forth.

For a traffic light l, the subset of sensory reports related to this light is denoted by Ω_l. As a result, we can obtain a set $\mathcal{K}_l = \{(\widehat{s}_l(t),\ c(\widehat{s}_l(t)), t),\ \text{for all } r_i(t) \in \Omega_l\},\ \text{for all } l \in \mathcal{M}$, where $c(\widehat{s}_l(t))$ is the confidence of traffic light l with state $\widehat{s}_l(t)$.

In Panoramic Static Estimation, the problem of traffic light sensing for light l is transformed into finding a series of boundary time instants $[T_0, T_1, \ldots, T_{q-1}]$, at which the light changes its state, where $T_0 = T_{\min}$ and $T_{q-1} = T_{\max}$. Since we consider only two traffic light states, the light state of duration $[T_{i-1}, T_i]$ is opposite to that of $[T_i, T_{i+1}]$. It is hence sufficient to determine the state of the first duration $[T_0, T_1]$, denoted by s_0, and then the light states of other durations naturally follow. Thus, PSE is an optimization problem with $q - 1$ variables.

We design two other objectives, for example, *Violation minimization* and *Conformability maximization*. *Violation minimization* represents that the violation between $\widehat{s}_l(t)$ in $\mathcal{K}_l$ and the continuous traffic light state estimated should be minimized. *Conformability maximization* represents that durations of red interval and green interval should approximate those in reality. Many adaptive heuristic algorithms can be used to solve this problem, such as genetic algorithm, ant colony algorithm, and so forth.

4. Road Map Sensing

4.1. Background. Road map sensing aims to construct and update of digital road map with probe vehicle. Road map construction and update are the foundations of most systems that provide traffic services, such as navigation systems, online-traffic condition system, and so forth. Traditionary approaches of road map construction and update are based on geological survey. However, for areas where new road networks are created or existed roads are reconstructed or closed, these approaches cannot provide timely accurate road map and leads to errors within the services based on road map. For example, temporary roads have caused fatal accidents with even experienced drivers [25].

With more and more vehicles are equipped with GPS tracking devices, for example, taxis [11], buses [26], commercial and utility vehicles, it is possible to conduct road map sensing using GPS reports generated by these vehicles [7, 9, 12, 13, 27]. Thus, most existing map sensing approaches adopt the same strategy of clustering GPS reports that are likely on the same road segment and calculate the road centerline for each cluster. The clustering is conducted either based on an existed map [7] or high-sampling-rate GPS trajectories [12], typically at 1 Hz.

In this section, we introduce the problem of road map sensing with probe vehicles which generate coarse-grained GPS reports. It is a very challenging task due to the lack of existed map and inaccuracy of GPS reports.

Map building using offline GPS trajectories with high sampling frequencies have been studied too [6–9, 12, 13]. B-spline fitting is a popular method for approximating highways from GPS data [6, 8]. When the actual drive path of the GPS trace is unknown, data clustering is needed to group together traces that are likely from the same road [7, 9, 12, 13]. This problem is very challenging even with high-sampling-rate data. In [7, 13], the clustering is assisted by a base map. In [12], gravitational and attraction forces are simulated to cluster GPS traces. With a sampling interval of 15 seconds or higher, clustering GPS traces is extremely difficult. In our GPS traces, it is not uncommon that a taxi generates no more than one sample per road segment. The trajectory bears little or no similarity with the true road geometry. Highly accurate GPS locations benefit the data clustering significantly. In [12], the standard deviation of the GPS Gaussian noise is estimated to be 4.07 meters. With highly accurate GPS devices, traces from the same road but opposite heading directions are clearly distinguishable even visually [9, 12]. With a resolution limit already at around 10 meters, we have never observed any clear separation of GPS traces heading opposite directions.

In our algorithm, heading directions of vehicles are utilized to assist the data clustering. This information has been used to separate traces of opposite driving directions [9] and has been used to coarsely split traces in the preprocessing step [13]. Different GPS traces are then grouped together based on trajectory similarity. Some existing work go beyond road recognition and infer intersection and lane structures [7, 27], which we do not address in this paper.

4.2. Problem Description. Road map sensing aims to construct and update maps from coarse-grained GPS records reported by probe vehicles. A complete routable map used in navigation devices would contain geometry, lane configuration, speed limit, turn restriction, road type information, and so forth. In this section we focus on the recognition of road geometry, which is of first priority in building routable map.

The data set utilized for road map sensing is $\Omega(T_{\min}, T_{\max})$. Several major performance metrics are considered. First, high coverage is desirable. For given $\Omega(T_0, T_{\text{now}})$, we want to recognize as many roads as possible. Second, low false negative rate should be achieved. As the GPS records are coarse-grained, there may be roads that do not exist in reality but recognized. Thus, one objective is to gain as low false negative rate as possible. Third, it is also desirable to gain high accuracy. Three aspects are defined to measure accuracy, for example, horizontal and vertical shift, and

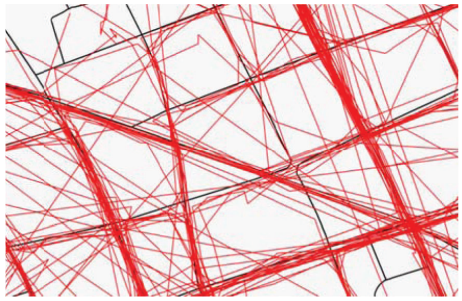


Figure 3: Vehicle trajectories by connecting consecutive GPS reports of every vehicle.

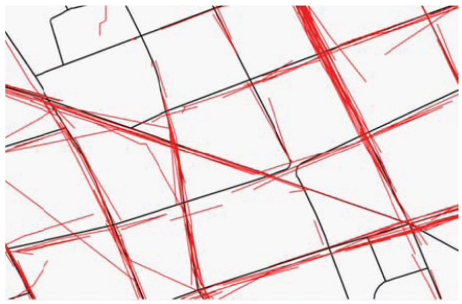


Figure 4: Result of trajectories pruning. Useless components are discarded.

separation distance between the recognized roads and roads in OpenStreetMap, and roads in truth in addition.

There are several challenges in road map sensing with probe vehicles. First, errors exist in the GPS reports generated by probe vehicles. As shown in Figure 2, we can see that most GPS reports are away from the centerline of road segments, and heading directions of reports on the same road segments are not uniform. Second, the distribution of GPS reports on road segments is uneven. Some road segments have only few attaching GPS reports, for example, the circles one in Figure 2, and the frameworks of these road segments are not clear just with GPS reports. Third, we can connect consecutive footprints of a vehicle and get a continuous trajectory, as shown in Figure 3. Although such trajectories can provide sufficient information for detecting roads with few footprints, they also bring negative influence. In Figure 3, we can see that a considerable number of the raw trajectories are messy and provide no useful information for road map sensing. This is because these raw trajectories are not the actual drive paths of the vehicles.

4.3. Detecting Roads. In this section we propose the algorithm for road map sensing. The basic idea is to first aggregate the GPS trajectories that are likely on the same road segments into one cluster, and then apply fitting algorithms to obtain a polyline representing the road centerline for each cluster. However, in the very beginning, useless components in the GPS trajectories should be discarded.

As shown in Figure 4, we discard the part of GPS trajectories that cannot agree with real travelling route of

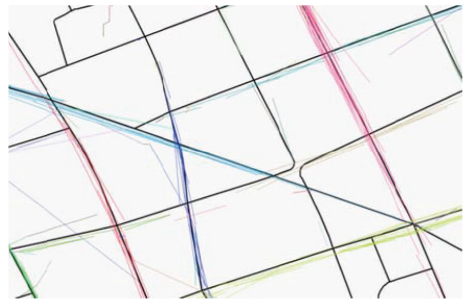


Figure 5: Result of trajectories clustering. Different clusters are in different colors.

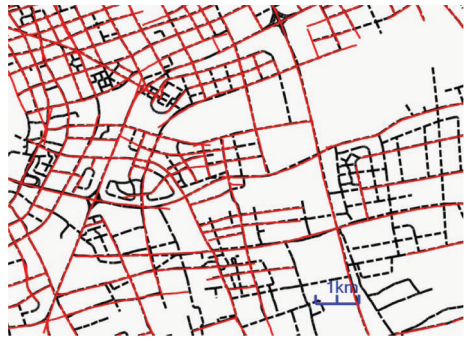


Figure 6: Road map sensing result: dashed lines are roads in OpenStreetMap, and red ones are those recognized.

vehicles. The criterion used is that the heading directions of two consecutive GPS reports together with the orientation of connected line segment in the trajectory should be the same for one vehicle.

As shown in Figure 5, trajectories clusters are plotted in different colors. The basic idea is to allocate trajectories generated by vehicles likely travelling on the same road segments to one cluster.

For one GPS trajectory cluster shown in Figure 5, we want to generate one road segment utilizing cure fitting algorithms. Many fitting algorithms could be applied, including polynomial fitting, Weibull fitting, and so forth. We considered and tried nearly all fitting algorithms but the results are not good, as the background road segment of one trajectories cluster can be various types, for example, straight line, arches, or even curve. Finally, we found spline fitting [6, 8] is suitable and modify it to make it adaptive. The fitting result of an area selected is shown in Figure 6.

In the trajectories clustering step, the problem shown in Figure 7 comes up. One possible approach to solve the problem is as follows. Firstly, find one polyline as the backbone. Secondly, GPS reports near the backbone are allocated into one new cluster and removed from the original cluster. Repeat these two steps till no more GPS reports exist in the cluster.

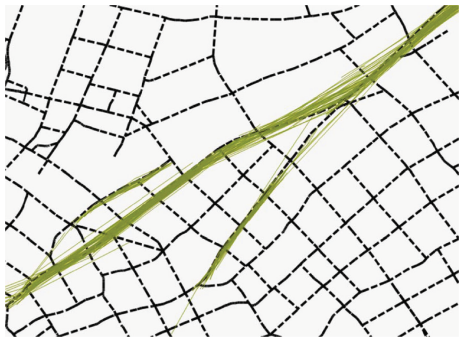


Figure 7: The problem: trajectories at road splits with small separation angles are allocated into one cluster.

5. Conclusion

This paper has presented the concept of Pervasive Urban Sensing (PUS) with probe vehicles. The general framework of PUS is presented. In addition, we also present some analysis on a dataset of real vehicular traces that have been collected from taxis operational in Shanghai, China. Two cases of PUS, that is, traffic light sensing and road sensing with probe vehicles are discussed. The problems of the two sensing cases are described. In addition, the basic algorithms for resolving the two sensing problems are presented.

We believe that PUB with probe vehicles will become increasingly practical and it will benefit the people living in cities by providing with valuable real-time urban information. However, many challenging issues remain untouched and a lot of research efforts are still required.

Acknowledgments

This research is supported in part by Shanghai Pujiang Talents Program (10PJ1405800), Shanghai Chen Guang Program (10CG11), NSFC (no. 61170238, 60903190, 61027009, 61202375, 61170237), MIIT of China (2009ZX03006-001-01), Doctoral Fund of Ministry of Education of China (20100073120021), National 863 Program (2009AA012201 and 2011AA010500), HP IRP (CW267311), SJTU SMC Project (201120), STCSM (08dz1501600), Singapore NRF (CREATE E2S2), and Program for Changjiang Scholars and Innovative Research Team in Universities of China (IRT1158, PCSIRT).

References

[1] "Portland traffic signals optimization," http://www.climatetrust.org/.

[2] "Statistics researchers predict road traffic conditions," http://domino.watson.ibm.com/comm/research.nsf/pages/r.statistics.innovation.traffic.html..

[3] J. Vrancken and O. Kruse, "Intelligent control in networks: the case of road traffic management," in *Proceedings of the IEEE International Conference in Networking, Sensing and Control (ICNSC '06)*, pp. 308–311, 2006.

[4] "Signal optimization in seattle," http://www.cityofseattle.net/transportation/.

[5] B. A. Toledo, V. Muñoz, J. Rogan, C. Tenreiro, and J. A. Valdivia, "Modeling traffic through a sequence of traffic lights," *Physical Review E*, vol. 70, no. 1, Article ID 016107, 2004.

[6] D. Ben-Arieh, S. Chang, M. Rys, and G. Zhang, "Geometric modeling of highways using global positioning system data and b-spline approximation," *Journal of Transportation Engineering*, vol. 130, no. 5, pp. 632–636, 2004.

[7] S. Schroedl, K. Wagstaff, S. Rogers, P. Langley, and C. Wilson, "Mining GPS traces for map refinement," *Data Mining and Knowledge Discovery*, vol. 9, no. 1, pp. 59–87, 2004.

[8] M. Castro, L. Iglesias, R. Rodríuez-Solano, and J. A. Sánchez, "Geometric modelling of highways using global positioning system (gps) data modelling of highways using global positioning system (gps) data and spline approximation," *Transportation Research Part C*, vol. 14, no. 4, pp. 233–243, 2006.

[9] S. Worrall and E. Nebot, "Automated process for generating digitised maps through GPS data compression," in *Proceedings of the Australasian Conference on Robotics and Automation*, 2007.

[10] Z. Li, Y. Zhu, H. Zhu, and M. Li, "Compressive sensing approach to urban traffic sensing," in *Proceedings of the 31st IEEE International Conference in Distributed Computing Systems (ICDCS '11)*, pp. 889–898, 2011.

[11] K. Liu, T. Yamamoto, and T. Morikawa, "Feasibility of using taxi dispatch system as probes for collecting traffic Information," *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, vol. 13, no. 1, pp. 16–27, 2009.

[12] L. Cao and J. Krumm, "From GPS traces to a routable road map," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '09)*, pp. 3–12, November 2009.

[13] L. Zhang, F. Thiemann, and M. Sester, "Integration of gps traces with road map," in *Proceedings of the 2nd International Workshop on Computational Transportation Science*, 2010.

[14] Y. Wang, Y. Zhu, Z. He, Y. Yue, and Q. Li, "Challenges and opportunities in exploiting large-scale GPS probe data," Tech. Rep. HPL-2011-109, HP Labs, 2011.

[15] F. van Diggelen, "GNSS accuracy: lies, damn lies, and statistics," *GPS World*, vol. 18, no. 1, pp. 26–32, 2007.

[16] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan, "Interactive wifi connectivity for moving vehicles," in *Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM '08)*, pp. 427–438, ACM, August 2008.

[17] X. Zhang, J. K. Kurose, B. N. Levine, D. Towsley, and H. Zhang, "Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, pp. 195–206, ACM, September 2007.

[18] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden, "A measurement study of vehicular internet access using in situ Wi-Fi networks," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 50–61, ACM, September 2006.

[19] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "Tbd: Trajectory-based data forwarding for light-traffic vehicular networks," in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems*, pp. 231–238, IEEE, 2009.

[20] S. Nelson, M. Bakht, R. Kravets, and A. Harris III, "Encounter: based routing in dtns," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 13, no. 1, pp. 56–59, 2009.

[21] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, vol. 6, April 2006.

[22] B. Coifman, "Improved velocity estimation using single loop detectors," *Transportation Research A*, vol. 35, no. 10, pp. 863–880, 2001.

[23] W. Chou, "Maximum a posterior linear regression with elliptically symmetric matrix variate priors," in *Proceedings of the 6th European Conference on Speech Communication and Technology*, 1999.

[24] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[25] "More than 50 crashes on bay bridge curve," USA Today, 2009, http:// www.usatoday.com/news/nation/2009-11-18-bay-bridge N.htm.

[26] R. L. Bertini and S. Tantiyanugulchai, "Transit buses as traffic probes: use of geolocation data for empirical evaluation," *Transportation Research Record*, vol. 1870, pp. 35–45, 2004.

[27] Y. Chen and J. Krumm, "Probabilistic modeling of traffic lanes from GPS traces," in *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '10)*, pp. 81–88, November 2010.