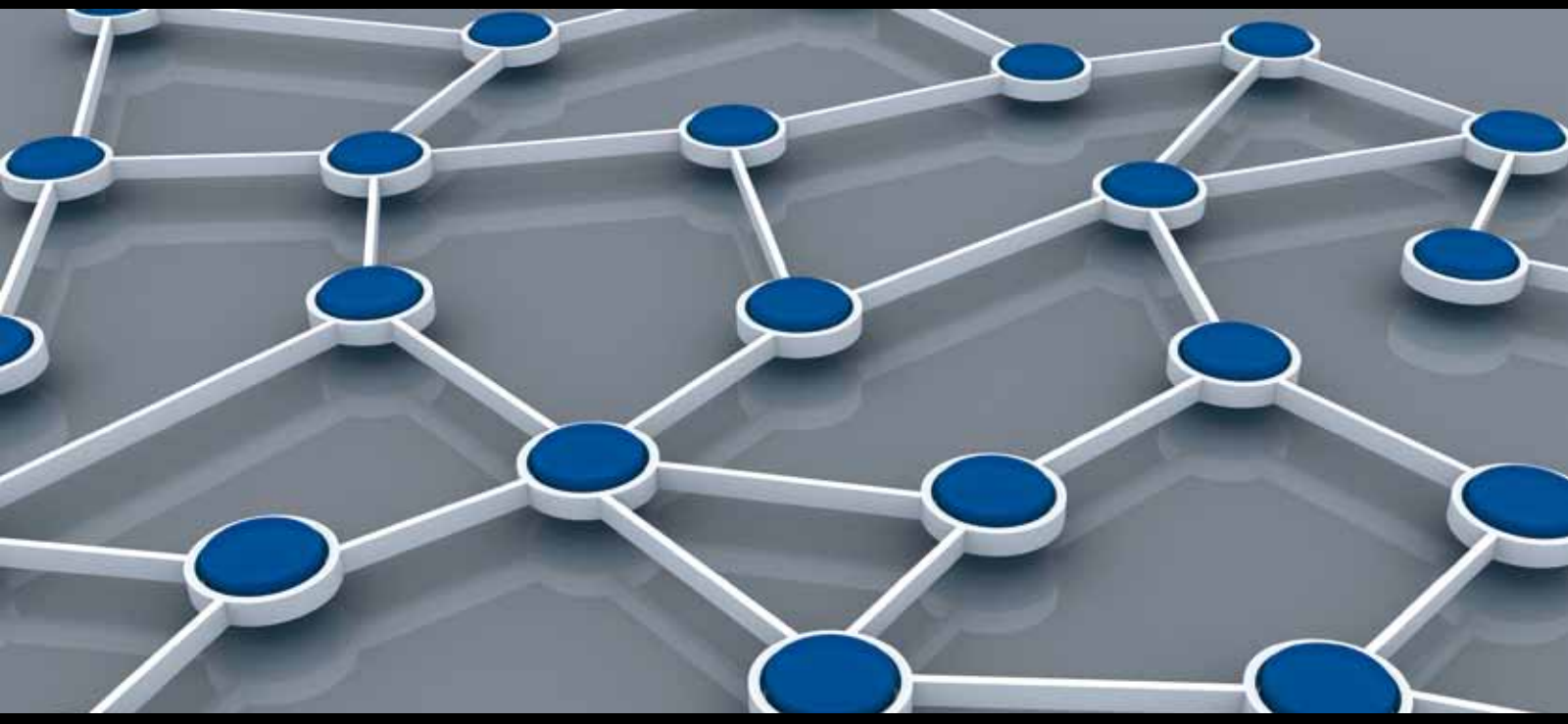
International Journal of Distributed Sensor Networks

New Technologies and Research Trends for Mobile Wireless Sensor Networks

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



New Technologies and Research Trends for Mobile Wireless Sensor Networks

International Journal of Distributed Sensor Networks

New Technologies and Research Trends for Mobile Wireless Sensor Networks

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

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 929121, 2 pages
http://dx.doi.org/10.1155/2014/929121

Editorial

New Technologies and Research Trends for Mobile Wireless Sensor Networks

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

[1] *Computer Science and Technology Institute, Soochow University, Suzhou, Jiangsu 215006, China*
[2] *School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 215011, China*
[3] *Department of Computer and Information Sciences, Temple University, 1039 Wachman Hall, 1805 N. Broad Street, Philadelphia, PA 19122, USA*

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

Received 24 December 2013; Accepted 24 December 2013; Published 29 April 2014

Mobile wireless sensor network (MWSN) has recently attracted considerable attention. It can simply be defined as WSN in which mobility plays a key role in the execution of the application. Both academia and industries are now focusing on developing MWSNs not only for the Internet, but also for other applications, such as target tracking, traffic monitoring, urban microclimate, and disaster monitoring. The challenges for such networks not only encompass a broad spectrum of research topics but also involve envisioning new multidisciplinary applications that will change the way in which we live and work. The authors have focused on relay cooperation models based on the network coding, node localization, spectrum sensing, neighbor discovery, congestion control, and clock synchronization, and so forth.

The paper "*Data dissemination in mobile wireless sensor network using trajectory-based network coding*", by L. Li et al., investigates the mobile nodes to forward data referring to their trajectory information. It is designed according to the characteristics of MWSN and appropriate for the mobile nodes that share anonymously its prepath for the higher bandwidth. Network coding is used to adapt for the dynamics velocity of mobile nodes. The method improves the reliability and scalability of MWSN in any scenario and benefits its deployment.

The paper "*A routing algorithm based on dynamic forecast of vehicle speed and position in VANET*", by H. Huang and S. Zhang, proposes the concept of circle changing trends angle in vehicle speed fluctuation curve and the movement domain. It designs an SWF routing algorithm based on the vehicle speed point forecasted and the changing trends time computation. The algorithm has a certain degree of improvement in routing hops, the packet delivery ratio, delay, and link stability.

In the paper "*Research on vehicle automatically tracking mechanism in VANET*" by L. Wang et al., the authors make some useful explorations in the fields of intelligent vehicle control technology and obstacle avoidance. The deep research is carried out about the distance of vehicle, vehicle tracking, vehicle lane changing and intersections obstacle avoidance and communication protocols. Some innovative ideas are proposed during the research.

In the paper "*Congestion control based on consensus in the wireless sensor network*" by X. Yang et al., the authors novel introduce a congestion control algorithm (CC-CA). NS simulation results indicate that the proposed algorithm restrains the congestion over the wireless sensor network, maintains a high throughput and a low delay time, and also improves the quality of service for the whole network.

In the paper "*EasiND: neighbor discovery in duty-cycled asynchronous multichannel mobile WSNs*" by T. Huang et al., the authors present an EasiND protocol for MWSNs. The proposed neighbor discovery system based on quorum system can bound the discovery latency in multichannel scenarios with low power consumptions. The proposed optimal asynchronous neighbor discovery system can minimize the power consumption with bounded discovery latency.

The paper "*Spectrum sensing for cognitive coexistent heterogeneous networks*", by B. Zhao et al., proposes a spectrum

sensing scheme for the cognitive coexistent heterogeneous networks. In this scheme, the power decomposition is formulated into a problem of solving a nonhomogeneous linear equation matrix. Both the analysis and the simulation results show the feasibility and efficiency of the proposed scheme.

The paper "*A type of localization method using mobile beacons based on spiral-like moving path for wireless sensor networks*", by C. Sha and R.-C. Wang, proposed a type of energy optimization localization method for MWSN. Traverse point is marked with the help of the optimum deployment model. According to the moving path and the localization time, energy consumption of the network could be estimated, and the sleep scheduling strategy for the node is localized.

The paper "*A comparison of clock synchronization in wireless sensor networks*", by S. Youn, examines the clock synchronization issues in WSN. A comparison of different clock synchronization algorithms in wireless sensor networks with a main focus on energy efficiency, scalability, and precision properties of them are provided in the paper.

In the paper "*LIRT: A lightweight scheme for indistinguishability, reachability, and timeliness in wireless sensor control networks*" by W. Ren et al., the authors make the first attempt to specify the security requirements for WSCNs. In addition, several new attacks in WSCN are pointed out at the first time. A lightweight scheme LIRT is proposed with tailored design to guarantee the indistinguishability, reachability, and timeliness inWSCNs.

The paper "*Face recognition in mobile wireless sensor networks*", by Q.-M. Lin et al., presents a new wireless sensing network paradigm for face recognition applications. In addition to the flexibility the face recognition system gains by integrating into a wireless sensor network, they take it further by introducing mobility into the network to improve the sensing coverage area and cost efficiency. A multilayered network structure and Gauss-Markov mobility model are proposed.

Yong Sun
Shukui Zhang
Hongli Xu
Shan Lin

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 427679, 9 pages
http://dx.doi.org/10.1155/2014/427679

Research Article

A Multicast Algorithm for Wireless Sensor Networks Based on Network Coding

Zhi-jie Han,[1,2] Ru-chuan Wang,[2] and Fu Xiao[2]

[1] *College of Computer & Information Engineering, Henan University, Kaifeng, Henan 475001, China*
[2] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China*

Correspondence should be addressed to Zhi-jie Han; hanzhijie@qq.com

Received 6 June 2013; Accepted 8 September 2013; Published 20 January 2014

Academic Editor: Shukui Zhang

We propose a set of distributed algorithms for improving the multicast throughput in wireless sensor networks. To this end, network coding is applied when exploiting path diversity with two disjoint paths to each multicast group receiver. We depart from the traditional wisdom that the multicast topology from source to receivers needs to be a tree and propose a novel and distributed algorithm to construct a 2-redundant multicast graph (a directed acyclic graph) as the multicast topology, on which network coding is applied. We conduct both analytical and simulation-based studies to evaluate the effectiveness and performance of our algorithm.

1. Introduction

Multicast mechanism is mainly used in sink nodes to send control messages to the sensor nodes in wireless sensor networks (WSNs), or in a sensor node to send data to multiple sink nodes. Multicast routing algorithm plays a vital role in WSN regarding the survival time and transmission efficiency of the WSNs.

Ahlswede et al. proposed network coding [1–3] in 2000. This method is useful for greatly improving the network throughput and reliability. Literatures [4, 5] have proved that in each multicast diagram, corresponding linear coding can be found to achieve minimum cut-maximum flow. Zhu et al. [6] found that the network coding has good performance and advantage in multicast network, but the network topology has different effect on the throughput and bandwidth of multicast; redundant multicast figure presented better efficiency and performance than the traditional multicast tree.

Jiang et al. [7] proposed a multicast tree algorithm through network coding, determined the relationship between cluster heads, and analyzed the algorithm performance using network calculus. Yuan et al. [8] analyzed the performance of network coding by the stochastic process. Li et al. [9] analyzed the end-to-end route delay performance based on the network coding using network calculus.

According to the WSN characteristic, considering the geographic and energy-aware routing (GEAR) [10], we constructed k-redundant multicast graph in the form of overlay network in this paper using the multicast algorithm based on network coding (MABNC). We improved the multicast performance by reducing the energy consumption.

2. Algorithm Description

The proposed algorithm is for sensor networks. Based on the geographic routing algorithms, k-redundant multicast graph was constructed. The network throughput and bandwidth utilization were improved, the effect of minimum cut to maximum flow was determined, and the network multicast routing performance was enhanced by utilizing the network coding mechanism.

Definition 1. Redundancy multicast figure: redundancy multicast figure is a directed acyclic graph with a single multicast source node and has the following two properties.

(1) All nodes belong to A, including the source node s, the relay node A_1, and the receiving node A_t. s is expressed as indegree$(s) = 0$ and outdegree$(s) > 0$; the forward nodes are $u \in A_I$, indegree$(u) \leq k$, and outdegree$(u) > 0$; the receiving

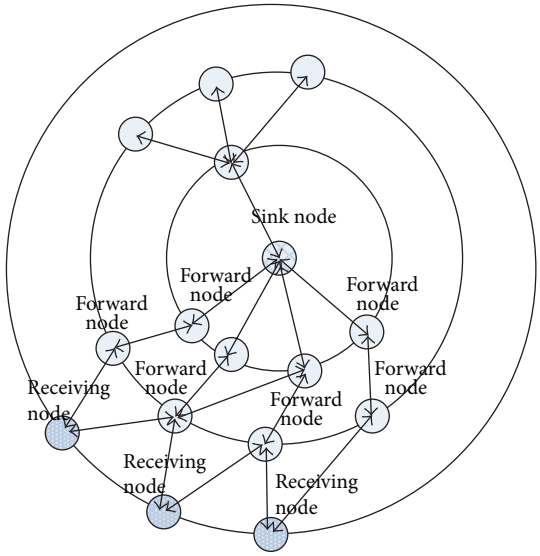


Figure 1: WSN topology.

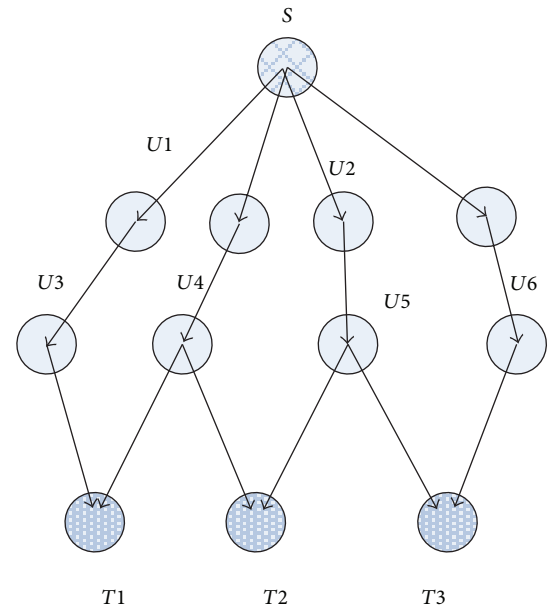


Figure 2: 2-redundant network diagram.

node is expressed as $T_i \in A_T$, $1 \le i \le T$, indegree$(T_i) = k$, and outdegree$(T_i) \ge 0$.

(2) Each side has a unit Bandwidth; k in-degree for the receiving node T_i, whose independently maximum flow is k. The independent maximum flow refers to the maximum flow through the source node s to the receiving node T_i.

Figure 1 shows that among the WSN nodes, the gathering and sensor nodes formed wireless sensor multiple hops self-organizing network through hierarchical forwarding and were connected; the first layer comprises the gathering node, the second layer comprises the cluster head node, and the third and fourth layers comprise the common sensor nodes. Figure 1 can be further decomposed into k-redundant multicast figure. Figure 2 shows that the gathering node can be the multicast source node s, the second and third layer nodes can be the multicast transit nodes, and the ordinary sensor nodes can be the receiving nodes. In this paper, $k = 2$. For the 2-redundant multicast figure, the key is how to build k-redundant multicast figure and how network coding can be reasonable.

2.1. Building and Maintaining k-Redundant Multicast Figure. Building k-redundant multicast figure needs to satisfy the following conditions to ensure the performance of multicast routing:

(1) ensure minimum energy consumption;

(2) build k-redundant multicast path from the source node s to the receiving node t;

(3) guarantee the minimum amount of transit nodes built from the source node to all of the receiving nodes;

(4) maintain the least link load stressed to ensure the minimum times being forwarded on the same physical link of the same packet.

The whole process can be divided into three steps as follows: (1) build a preliminary figure that can contain all nodes; (2) simplify the preliminary figure to k-redundant multicast figure; (3) combine the receiving node t.

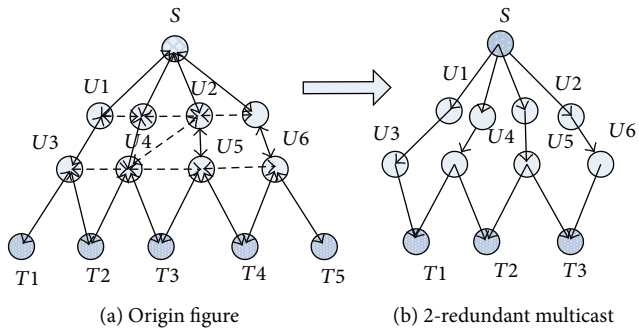


Figure 3: Conversion of the origin figure to 2-redundant multicast figure.

2.1.1. Building Preliminary Figure. In building a preliminary figure, when a new node joins a preliminary diagram, GEAR mechanism is used. The nearest preliminary node will be chosen to lead the joining of preliminary to be transshipment node and form the preliminary figure of each node, as shown in Figure 3(a).

2.1.2. Building k-Redundant Multicast Figure. Figure 3 shows that for each receiving node, two unmutual crossed best route paths need to be established from the source node in a preliminary figure. The pros and cons of a path are mainly evaluated through a tree tuple.

Definition 2. Path evaluation: the quality of the link from the source node s to the receiving node t must be evaluated. If the link quality is a triple, the edge of the preliminary figure

(or the link between two nodes i, j) is set to e; that is, $\text{path}_{s\to t}$ goes through m transshipment nodes. The quality evaluation function of side e_{ij} is

$$w\left(e_{ij}\right) = \alpha \times D_{ij} + \beta \times \text{Bandwidth}\left(e_{ij}\right) + \lambda \times \text{Latency}\left(e_{ij}\right), \tag{1}$$

where α, β, and λ are the normalized parameters, D_{ij} is the distance between two nodes, Bandwidth is the direct bandwidth of the two nodes, and Latency refers to the time delay between the two nodes. The quality assessment of the entire link is

$$\begin{aligned} w\left(\text{path}_{s\to t}\right) &= \left(D, \text{Bandwidth}\left(\text{path}_{s\to t}\right), \right. \\ &\qquad \left. \text{Latency}\left(\text{path}_{s\to t}\right)\right) \\ D &= D_{t1} + D_{12} + \cdots + D_{ms} \\ \text{Bandwidth}&\left(\text{path}_{s\to t}\right) \\ &= \min\left(\text{Bandwidth}\left(e_1\right), \text{Bandwidth}\left(e_2\right), \right. \\ &\qquad \left. \ldots, \text{Bandwidth}\left(\text{e}_m\right)\right) \\ \text{Latency}&\left(\text{path}_{s\to t}\right) = \sum_{i=1}^{m} \text{Latency}\left(e_i\right). \end{aligned} \tag{2}$$

In Formula (2), the minimum Bandwidth side should be the evaluation value when evaluating the path Bandwidth because in the path, only the flow of the minimum Bandwidth edge can be the maximum flow of the entire path.

In WSNs, as the sensor nodes join or quit dynamically, the network topology changes. The preliminary figure also changes. The path quality evaluation and calibration can be performed periodically. When a better path occurs, the existing path must be replaced with the better path to guarantee at least two or more unmutual crossed best route paths from the receiving node t to the source node s.

To transform the distributed graph into 2-redundant multicast figure, according to the definition requirements of 2-redundant multicast figure, the following requirements must be satisfied.

(1) In multicast figure, the degrees (i.e., the sum of indegree and outdegree) of all nodes, including the source node s, the transmitted node u, and the receiving node t, should not be greater than Δ, where Δ is the number of the multicast figure layer and is the height of the tree. As shown in Figure 3(b), the highest layer is 4.

(2) The source node s should has k child node, and $2 \le k \le \Delta - 1$.

(3) $\text{degree}(u) \le \Delta$, and $1 \le \text{indegree}(u) \le k$.

(4) If the child node of the transfer u is the receiving node t, then the transfer u child node's number is not greater than $\Delta - 1 - \text{indegree}(u)$.

The fourth rule is used to ensure that at least one outdegree's value of the transit nodes is vacant for adding new transit nodes and for realizing the extensibility and scalability of the multicast figure. When adding a new receiving node to the multicast figure, we can determine the new transit node rapidly to receive the join requests and improve the search speed of convergence.

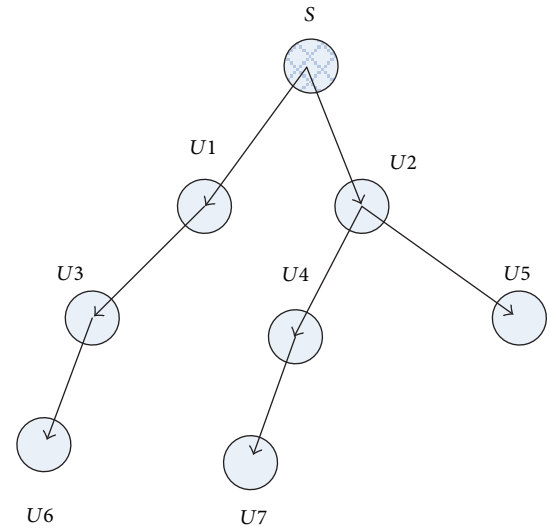


Figure 4: Tree consisting the transiting node.

2.1.3. Receiving Node to Join. Given the low energy of node and the complex network environment of WSNs, the node failure or sleep scheduling causes rapid changes in the network topology. Therefore, the flexible mechanism of the receiving node is needed to join.

The node that can receive the joining node is defined as the transiting node of leave.

Definition 3. The transiting node of leave denotes that its child nodes are not the leave nodes in the multicast figure. As shown in Figure 4, *U3*, *U4*, *U5*, and *U6* are all the transiting nodes of leave.

Definition 4. Saturated node: in the transiting node of leave, if the degree is $\text{degree}(u) = \Delta - 1$, then the node is a saturated node, such as *U4* and *U5* nodes in Figure 3. Otherwise, it is an unsaturated node, such as *U3* and *U6* nodes.

The receiving node t to join the process of multicast is also added to the tree that consists of the transfer nodes in Figure 4. The process can be divided into two steps as follows: (1) finding the proper unsaturated node to join in the tree; (2) finding k that does not cross paths connected to the source node s.

Figure 5 shows the joining of the node T in the multicast diagram. The node T determines first whether a node of multicast diagram exists in the neighbor nodes. The following three situations exist.

(i) Two or more than two nodes exist. According to the function of the link quality evaluation in Formula (1), two nodes are chosen from many eligible nodes and

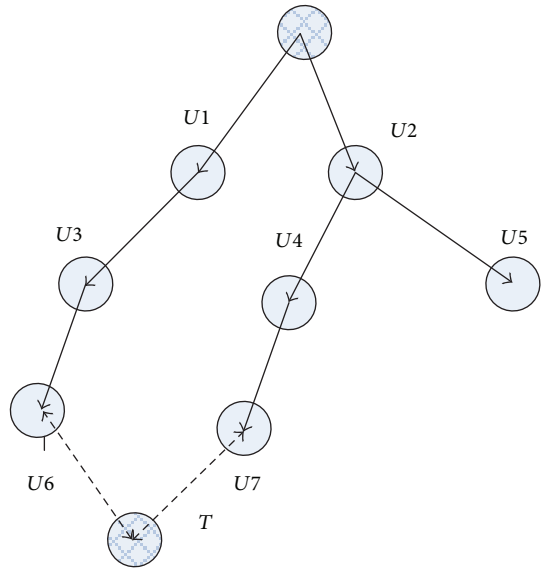


Figure 5: Node to join.

the request of access is initiated. The node T has two neighbor nodes (i.e., $U6$ and $U7$), which are the nodes of diagram in Figure 5. They are requested to join in the multicast diagram.

(ii) If only one node exists, links to join and begins flooding search until another multicast diagram node is found.

(iii) Requests are flooded until two multicast diagram nodes are found if the neighbor node does not exist in the multicast diagram nodes.

Two mutually cross paths (*Pf* and *Ps*) must be built when the node t connects to the multicast diagram. Let *uf* be the father node of t in the path of *Pf*. Specific algorithms are presented in Algorithm 1.

Pf is the path of the receiving node to the source node. The following setup of another path *Ps* with *Pf* is not the path of the cross, as shown in Algorithm 2.

2.2. k-Redundant MABNC. We established k redundant multicast diagram with a special topology based on the above work. The basic characteristic is that every receiving node has k paths. According to this characteristic, we designed a distributed networking coding mechanism to optimize the performance of the wireless sensor multicast. The maximum multicast traffic can be obtained under the same conditions.

In network coding, each transiting node assigns the appropriate coding vector according to certain rules to encode the forwarding data. The receiving node decodes the original data when it collects enough data and coding vector. Data are then received. The core key of this process is to distribute suitable coding vectors to each relay node.

In the tree of k-redundant multicast diagram, for example, 2-redundant multicast, each receiving node has two mutually crossed paths to the source node s. The source node s sends the data a and b in parallel at the same time. The encoding is mainly conducted on the relay node u_i, and the receiving node t can receive the coding vector and coding data at the same time. The source node s generates and distributes each relay node with the suitable coding vector. The generated function of coding vector is expressed as $g(m)$. The coefficient generation is elucidated in [4]. The encoding vectors are assumed as follows:

$$\left\{(p_1, q_1)^T, (p_2, q_2)^T, \ldots, (p_m, q_m)^T\right\}. \tag{3}$$

The encoding vectors satisfy the following rules:

(1) p_i, q_i are on the domain and linearly independent;

(2) (p_i, q_i) and (p_j, q_j) are linearly independent, and $i \neq j$;

(3) $(p_i, q_i)^T$ is the coefficient of a code and can encode the data

$$c_i = (a, b)\,(p_i, q_i)^T = p_i \cdot a + q_i \cdot b. \tag{4}$$

c_i and c_j are also linearly independent. In $C = (c_1, c_2, \ldots, c_m)$, the source node is assumed to produce five coding coefficients: $(1,0)^T, (0,1)^T, (1,1)^T, (2,1)^T$, and $(1,2)^T$; then $C = \{a, b, a+b, 2a+b, a+2b\}$. Any two groups of data in C are linearly independent. They can obtain the data a and b when the nodes of leave receive the two groups of data.

The core problem in the whole process is that how to make the relay node assigns the appropriate coding vector. The relay node is divided into the following two classes for processing:

(i) the relay nodes of indegree 1: as codes are not needed currently, the code vectors must be assigned;

(ii) the relay nodes of indegree 2: the encoding vector is $v_u = (v_1, v_2)^T$. Thus, for data x and y that passed through u, the code is $(x, y)v_u = (x, y)(v_1, v_2)^T = v_1 x + v_2 y$. u is then sent to the outside of the path.

The distribution of transiting node coding vector is divided into two stages, namely, the distribution and diffusion phases. Generated by the source node s, the corresponding distribution vector is assigned to each relay node. According to Algorithm 3, the source node s distributes the network coding vector to each node in the multicast diagram. The relay node has two indegrees that respond to the request and only one indegree that does not respond to the request. Assuming that a node responds to the request, the node s generates $m + j$ coding vectors, where m coefficient vectors are sent to m requesting nodes and j is the source node s' number of child nodes. As shown in Algorithm 4 at the diffusion stage, the source node s first distributes j coding vectors generated at the distribution stage to s' nodes of child. The child nodes will further distribute the coding vectors to its downstream nodes to diffuse. In the process, the following two types of nodes are needed to handle:

(i) node u of indegree 1: when receiving the coding vector $(p_i, q_i)^T$ from its parent node, the encoding vectors $w(u)$:$p_u = p_i$, $q_u = q_i$ are set;

(ii) node u of indegree 2: the node u is already at the distribution stage $w(u)$:$p_u = p_i$, $q_u = q_i$. From two

The receiving node t:
From the neighbor node of t or by flooding, find k unsaturated nodes $\{u_i\}$ that are not on the path Pf:
(1) Send paths to $\{u_i\}$;
(2) Request the unsaturated node u_i and let the best paths $\text{Path}_{s\to u_i}$ from the source node s to u_i and the weight value $w(\text{Path}_{s\to u_i})$ return;
(3) u = Find best path$(t, \{u_i\}, \{\text{Path}_{s\to u_i}\}, \{w(\text{Path}_{s\to u_i})\})$,
where Find best path() is as follows:
The link quality evaluation of query nodes $u_1, u_2, \ldots, u_m$ is $w(\text{Path}_{s\to u_i}) = (D, \text{Bandwidth}(\text{Path}_{s\to u_i}), \text{Latency}(\text{Path}_{s\to u_i}))$ for every unsaturated node u_i,
Calculate $w(\text{Path}_{s\to u_i} \cup e_{u_i t})$;
Choose the best path to return;
The transiting node u_i is
Receive the requests of the best path $\text{Path}_{s\to u_i}$ and the weight value $w(\text{Path}_{s\to u_i})$ sending from Pf and t;
If $\text{Path}_{s\to u_i}$ and Pf non-intersect, then
Send $\text{Path}_{s\to u_i}$ and $w(\text{Path}_{s\to u_i})$;
If $\text{Path}_{s\to u_i}$ and Pf intersect, but u_i shows another non-intersect path, then P_i and Pf non-intersect.

Algorithm 1: Node t building the algorithm of path Pf.

The receiving node t:
If the neighbor node of t is an unsaturated node$\{u_i\}$,
u = Find best path$(t, \{u_i\})$
$uf = u_i$ $pf = Pf \cup u_i$.
If the neighbor nodes of T are all saturated nodes,
Search the whole multicast tree
until an unsaturated node is found,
u = Find best path$(t, \{u_i\})$
$uf = u_i$ $pf = Pf \cup u_i$
Find best path() is defined as follows:
Find best path$(t, \{u_1, u_2, \ldots, u_m\})$.
The link quality evaluation of query nodes is $u_1, u_2, \ldots, u_m$,
$w(\text{Path}_{s\to u_i}) = (D, \text{Bandwidth}(\text{Path}_{s\to u_i}), \text{Latency}(\text{Path}_{s\to u_i}))$,
for every unsaturated node u_i,
calculate $w(\text{Path}_{s\to u_i} \cup e_{u_i t})$
Choose the best path to return.

Algorithm 2: Algorithm of node t building the path Pf.

SendP_i and $w(p_i)$;
If Indegree$(u_i) = 1$,
Contact the child node c of the source node s, and the subtree of c contained u_i. The unsaturated node v is then searching in the subtree of c.
Return the path $P = p(s \to v) \cap p(v \to u)$ and $w(P)$.

Algorithm 3: Stage of coding vector distribution.

Source node S:
Broadcast to each relay node of multicast diagram to distribute coding vector;
set T = (largest RTT) × 2;
set timer $t = 0$, initialize $m = 0$;
while ($t \le T$)
Receive $\langle$request code, address of $u\rangle$;
$m = m + 1$;
node addr $[m]$ = address of u;
j = Obtain the child nodes of the node s' number;
Obtain
$g(m + j) = \{(p_1, q_1)^T, (p_1, q_1)^T, \ldots, (p_{m+j}, q_{m+j})^T\}$
for $i = 1$ to m
send $\left\langle \text{code}, (p_i, q_i)^T \right\rangle$ to
node addr $[m]$;
Receive the distribution code of the transit node u;
If indegree(u) = 1, then
Do nothing
If indegree(u) = 2, then
send message $\langle$request code, address of $u\rangle$ to s;
Wait to receive $\left\langle \text{newcode}, (p_i, q_i)^T \right\rangle$;
Set $w(u) : p_u = p_i,\ q_u = q_i$.

Algorithm 4

Source node S:
S has produced $(m + j)$ coding vector.
For $i = 1$ to j
send $\langle$code, $g(m + j)\rangle$ to s' jth child node addr $[j]$;
The node of indegree 1 is as follows:
Receive $(p_i, q_i)^T$ sent from its father node;
Set $w(u) : p_u = p_i,\ q_u = q_i$;
The node of in-degree 2 is as follows:
Receive the coding vectors $(p_1, q_1)^T$ and $(p_2, q_2)^T$;
Set $v_u = \begin{pmatrix} p_1 & p_2 \\ q_1 & q_2 \end{pmatrix}^{-1} \begin{pmatrix} p_u \\ q_u \end{pmatrix}$.

Algorithm 5: Diffusion stage.

father nodes v, two vectors are obtained at the diffusion stage Algorithm 5. Supposing $(p_1, q_1)^T$ and $(p_2, q_2)^T$, then $v_u = (v_1, v_2)^T$. With 1 and 2 to represent the received data from two sides, then $(\alpha, \beta)(v_1, v_2)^T$ is sent to all of the nodes on the edge of the data. We then have

$$(a, b)\begin{pmatrix} p_1 & p_2 \\ q_1 & q_2 \end{pmatrix} = (\alpha, \beta). \tag{5}$$

The node u is sent to its degree of edge data as follows:

$$(a, b)\begin{pmatrix} p_u \\ q_u \end{pmatrix} = (\alpha, \beta)\begin{pmatrix} p_1 & p_2 \\ q_1 & q_2 \end{pmatrix}^{-1}\begin{pmatrix} p_u \\ q_u \end{pmatrix}. \tag{6}$$

The comprehensive Formulas (5) and (6) can be obtained as follows:

$$v_u = \begin{pmatrix} p_1 & p_2 \\ q_1 & q_2 \end{pmatrix}^{-1}\begin{pmatrix} p_u \\ q_u \end{pmatrix}. \tag{7}$$

In the above process, the relay node u distributes the appropriate coding vector. According to the above analysis, the receiving node t can be decoded to obtain (a, b) as it receives any two coded data.

Theorem 5. *In the 2-redundant multicast tree, after coding and forwarding by the middle node, the receiving node t can encode to obtain* (a, b) *sent from the source node s.*

Proof. Two mutually disjoint paths Pf and Ps exist according to the construction of 2-redundant multicast tree after the source s multicast forwarded the date (a, b). Given that the source node s is linearly independent, $(p_1, q_1)^T$, $(p_2, q_2)^T, \ldots, (p_m, q_m)^T$ is generated when coding vector. The assumption that the node t from two paths of encoded data is linearly independent can be proved if the two paths of the transit node coding vectors are proved to be linearly independent. □

This assumption can be verified using Formulas (5) and (6). If Formulas (5) and (6) were established, $\begin{pmatrix} p_1 & p_2 \\ q_1 & q_2 \end{pmatrix}$ must be reversible, that is, $\begin{pmatrix} p_1 & p_2 \\ q_1 & q_2 \end{pmatrix}$ is linearly independent.

3. Performance Analysis

3.1. k-Redundant Multicast Algorithm Complexity Analysis. The entire network node number is assumed as n from the preliminary figure to construct multicast tree. The main work is to build a tree diagram, of which the whole process is similar to the Bellman-Ford algorithm. The complexity is expressed as (VE), where V is for the entire figure on the number of nodes n, and E is used to determine the number of edges. The overall complexity of the algorithm is $O(n\text{^}2)$.

The main task of the receiving node joining is to find neighbor nodes. The number of control message is about $O(n\text{^}2)$. The receiving node in 2-redundant multicast figure finds suitable unsaturated node. The search process is aimed to find tree. The worst case is $O(\log n)$.

3.2. Network Coding Complexity Analysis. By distributing phase in the entire coding vector, the initialization of message transmission is the broadcasting of multicast tree. The number of packets is $O(n)$. A total of m messages are in response. The message number is $O(m)$. The diffusing phase of coding vector is to spread from each relay node to the child nodes. The number of packets is $2n$, so that the overall number of message is $O(n)$.

In the aspect of time delay, the maximum delay from the source node s to the receiving node is T. The source node s in the coding vector distribution stages sets the timing to $2T$ and initializes the broadcast message for T; thus, the overall worst case is $3T$. During the diffusion stage, t is assumed as any two nodes' maximum delay in the multicast figure. As the distributed algorithm is used, the extension of coding vector is generally not greater than t. Accordingly, the overall coding time delay is $O(T)$.

3.3. Simulation Analysis

3.3.1. Simulation Illustration. To evaluate and analyze the performance of multicast, the NS platform is used for simulation. The experimental environment is as follows: 500 sensor nodes are distributed to 100 m by 100 m square area randomly; the receiving node is 100, and the gathering node is in the network center. The signal collision and the influence of random factors, such as wireless channel interference, are ignored. The other important parameters used in the experiment are shown in Table 1. The topology of the network simulation is shown in Figure 6.

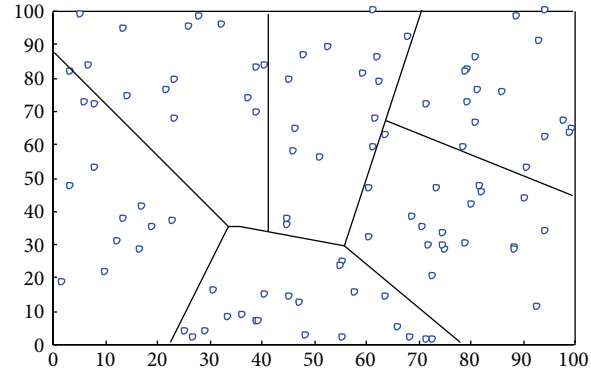


Figure 6: Topology of network simulation.

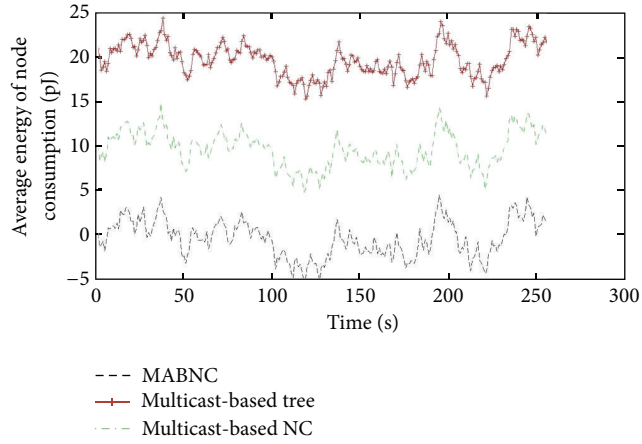


Figure 7: Average energy of node consumption.

3.3.2. Simulation Analysis. The comparative analyses of several aspects, such as the average energy consumption of nodes, end-to-end delay, and packet loss rate, with the traditional multicast mechanism based on tree and the tree multicast mechanism based on network coding are conducted.

(1) Average Energy Consumption of Nodes. The node energy consumption is that in WSNs, the average energy consumption in the node running is computed as follows:

$$\text{average}\left(\text{noden}_i\right) = \frac{1}{n}\sum_{i=1}^{n} E\left(n_i\right), \tag{8}$$

where n is the number of sensor nodes. As shown in Figure 7, the whole simulation ran for 256 s. The first 120 s is for the building of multicast topology. At this time, the energy consumption is more in the three algorithms. The following 136 s is for the data's multicast process. Based on the simulation curve, the computational cost of the MABNC algorithm of multicast figure based on network coding increased

Table 1: Simulation parameters.

Parameter	Value
d_0 (m)	75
Node initial energy (J)	1–3
E_{elec}	5 nJ/bit
ε_{fs}	10 pJ/(bit·m^2)
ε_{mp}	0.0013 pJ/(bit·m^4)
Message size	4000 bits

compared with the traditional multicast mechanism based on tree and tree multicast routing algorithms. However, the data retransmission times are effectively reduced and smaller energy consumption is yielded by using the network coding mechanism. Literature [11] indicates that in the sensor network, the bit data of energy consumption that the node sent is equivalent to 1000 times computing. Using the network coding can reduce the number of nodes sending data effectively. Compared with the tree routing mechanism based on network coding, the building and multicasting are more concise by using k-redundant multicast figure and can achieve the maximum theoretical flow of multicast routing.

(2) End-to-End Delay. Overall, the simulation running time is 1000 s. The first 120 s is for building a multicast topology, and the remaining time is for multicasting. As shown in the simulation Figure 8, the tree-based multicast routing mechanism of time delay is the largest. The second is the multicast mechanism by using the mechanism of network coding. As the sensor network nodes change dynamically, the topology is not stable. The tree-based multicasting mechanism needs to maintain and adjust tree and influences the performance of multicast. In the multicasting mechanism of k-redundant based on network coding, the time delay centered in three kinds of algorithm of end-to-end delay. The tree multicasting routing based on network coding has the best performance. After using k redundant, as a result of overlay routing, a bottom support of GEAR is needed and sometime delay is spent.

(3) Rate of Packet Loss. The rate of packet loss is the index of robustness to reflect the multicast routing algorithm. As shown in Figure 9, in the three kinds of multicast routing mechanism, the packet loss rate is higher because the sensor network environment is complicated and rapid changes exist in the network topology. The algorithm of k-redundant multicast routing has the least rate of packet loss. As k-redundant algorithm uses the idea of multipath, the network robustness is slightly better than the other two kinds of routing.

4. Summary

This paper aims to design the requirements of multicast routing protocol based on WSN as follows: energy efficiency, expandability, robustness, and fast convergence. Based on network coding, k-redundant multicast protocol is proposed and its working process is described. The routing protocol

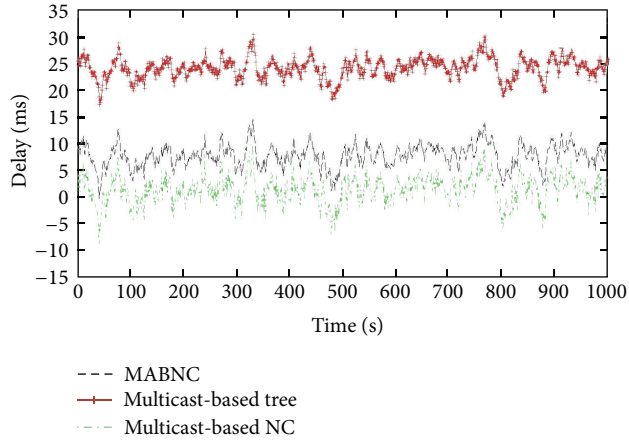


Figure 8: End-to-end delay.

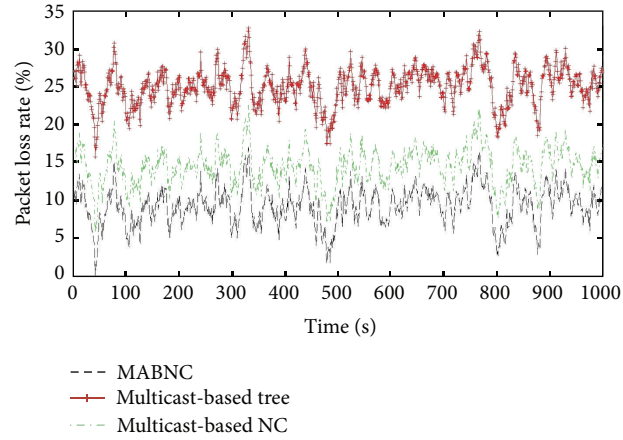


Figure 9: Packet loss rate.

is simulated at the platform of NS2.31. The simulation result indicates that the protocol routing has lower rate of packet loss, small delay, and relatively low cost of the entire network nodes.

Conflict of Interests

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

Acknowledgment

The work supported by the National Natural Science Foundation of China (Grants no. 61103195, and no. 61373137).

References

[1] S. Jaggi, P. Sanders, P. A. Chou et al., "Polynomial time algorithms for multicast network code construction," *IEEE Transaction on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.

[2] S. Y. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[4] R. Koetter and M. Médard, "Beyond routing: an algebraic approach to network coding," in *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, pp. 122–130, June 2002.

[5] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[6] Y. Z. Zhu, B. Li, and J. Guo, "Multicast with network coding in application-layer overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 107–120, 2004.

[7] L. Jiang, L. Yu, and Z. Chen, "Network calculus based QoS analysis of network coding in Cluster-tree wireless sensor network," in *Proceedings of the 11th International Symposium on Communications and Information Technologies (ISCIT '11)*, pp. 126–130, 2011.

[8] Y. Yuan, K. Wu, W. Jia, and Y. Jiang, "Performance of acyclic stochastic networks with network coding," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 7, pp. 1238–1245, 2011.

[9] H. Li, X. Liu, and W. He, "End-to-end decay analysis in wireless network coding: a network calculus-based approach," in *Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS '11)*, 2011.

[10] Y. Yu, R. Govindan, and D. Estrin, "Geographical and energy aware routing: a recursive data dissemination protocol for wireless sensor networks," Tech. Rep. TR-01-0023, Computer Science Department, University of California, Los Angeles, Calif, USA, 2001.

[11] S. Limin, *Wireless Sensor Network*, Tsinghua University Press, 2005.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 360426, 7 pages
http://dx.doi.org/10.1155/2013/360426

Research Article

Research on Achievable Rate of Interference Channel with Cooperative Transmission

Lin Xiao,[1] Dingcheng Yang,[1] Geng Su,[2] Anping Tan,[1] and Shengen Liu[1]

[1] *Information Engineering School, Nanchang University, Nanchang 330031, China*
[2] *School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK*

Correspondence should be addressed to Dingcheng Yang; ydcxuanyuan@msn.com

Received 20 May 2013; Revised 8 November 2013; Accepted 17 November 2013

Academic Editor: Hongli Xu

This paper studies the achievable data rate of the two-transmitter and two-receiver interference channel model with cooperative transmission. To implement the cooperation, a finite rate conferencing link is deployed at the transmitters side in order to share the message. A new transmit scheme with private message sharing is proposed based on the dirty paper coding. The achievable rate region is established in both strong interference regime and weak interference regime. In weak interference regime, the asymptotically rate region shows the conferencing link can improve the achievable rate region for cooperative receiver. In strong interference regime, the conferencing link not only improves the achievable rate for cooperative receiver but also improves the sum rate of the cooperative network. Numerical results demonstrate these theories in a Gaussian interference channel.

1. Introduction

The classic interference channel (IC) model is a two-transmitter and two-receiver model, which is firstly introduced in [1]. In this model, one transmitter sends information to the corresponding receiver and interferes with another transmitter-receiver pair. The study of interference channel is important for communication system design, because the practical systems are designed to operate in the interference scenario. In [2], Han and Kobayashi gave the classical achievable rate region for the interference channel, where the message is split into common information and private information. This splitting technique is used to partially decode and subtract the interfering signal.

Due to the scarcity of the spectrum resource and the high data transmit requirement, cooperative transmission had drawn much research attention [3–5]. Such cooperative transmission had been considered in the IC research. Interference channel with transmitter cooperation had attracted many researchers interests [6, 7]. The channel model is that two transmitters attempt to communicate with their respective receivers simultaneously through a common medium, and one transmitter has complete or partial knowledge about the message being transmitted by the other. It is first proposed in [6] that by utilizing cognitive techniques, the cognitive transmitter gains full knowledge of another transmitter message. During the transmission, the cognitive transmitter treats the message from the other transmitter as interference and tries to compensate for it by using a well-known Gelfand and Pinskers coding scheme [8]. This result enlarges the achievable rate region in [2] and reduces to the same region in [2] in the case where no interference mitigation is performed. In [9], a cooperative encoding scheme was proposed to derive new achievable rate regions, in which the channel is named as IC with degraded message sets (IC-DMS). Both [6, 9] proved that the rate region is achievable when applying the proposed coding schemes for the IC-DMS in the low interference regime, which means the cross-link gain between the receiver and its interfering transmitter is less than one. Considering the coding scheme in [9] is nonoptimal for IC-DMS in high interference regime, a cooperative coding scheme is proposed in [10]. By combining three coding methods, cooperative coding, collaborative coding, and Gelfand-Pinsker coding, the new coding scheme derives the achievable rate region not only in low interference regime but also in the high interference regime. Reference [11] proposed various

transmitter cooperation scenarios in the two-transmitter two-receiver interference channel (X-IC) and obtained the channel capacity in the strong interference regime.

However, the previous works assumed that the message sharing among transmitters via cognitive radio did not consider the limitation of the data rate of the information exchange on transmitter side. Reference [12] proposed a scheme with finite-rate cooperation at transmitters in Z-interference channel (Z-IC). It is proved that the achievable rate region can be improved by the conferencing link in both low interference regime and high interference regime in Z-IC. The cooperation can be deployed at not only the transmitter side but also the receiver side. In [13], the achievable rate region is studied with receiver cooperation, in which the receiver will relay the public information to the other receiver.

Most of the works are based on the assumption that the public message can be shared either between transmitters or receivers. In such cooperation scenario, the transmitter cooperation sharing the public message can mitigate the interference. In this paper, we extend the transmitter cooperation in X-IC with private message sharing for joint processing [3]. We consider the one-side information exchange in the transmitters side and cooperative transmission, which can be thought as one transmitter acting as the relay node in the X-IC [14] for cooperative transmission. In such scenario, the transmitter cooperation can not only mitigate the interference but also improve the useful signal transmission gain.

The contributions of this paper are as follows: (1) we study the X-IC channel with cooperative transmission, in which the private message of one transmitter can be transferred to another transmitter and then sent to the corresponding receiver cooperatively; (2) the achievable rate region is established in both strong interference regime and weak interference regime. In weak interference regime, the asymptotically rate region shows the conferencing link can improve the achievable rate region for cooperative receiver. In strong interference regime, the conferencing link not only improves the achievable rate for cooperative receiver but also improves the sum rate of the cooperative network.

The rest of the paper is organized as follows: the system model is given in Section 2; the achievable rate region in modified interference channel with cooperative transmission is proved in Section 3; Section 4 computes an achievable rate region in the AWGN case. The numerical results and the conclusion are given at the end of the paper.

2. Modified Interference Channel with Cooperative Transmission

In this section, a two-transmitter two-receiver interference channel with transmitter cooperation is defined in which transmitter T_{x1} has the knowledge of the message to be transmitted by the transmitter T_{x2}. The transmitter T_{x1} could refrain from transmitting its own information and act as a relay for transmitter T_{x2}. Similar to [2], a modified IC with cooperative transmission C_m is introduced and demonstrates an achievable region $\mathscr{R}^m$. After that, a relation between an achievable rate for C_m and an achievable rate for C is used

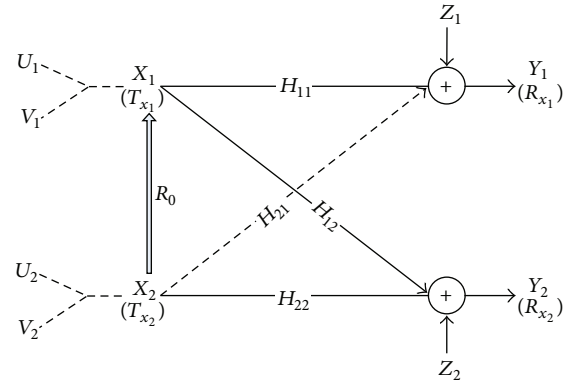


Figure 1: The modified interference channel model with transmitter cooperation.

to establish an achievable region for the IC with cooperative transmission. The modified C_m is defined as in Figure 1.

Let $X_1 \in \mathscr{X}_1$ and $X_2 \in \mathscr{X}_2$ be the random-variable inputs to the channel. Let $Y_1 \in \mathscr{Y}_1$ and $Y_2 \in \mathscr{Y}_2$ be the random-variable outputs of the channel. The conditional probabilities of the discrete memoryless channel are the same as in [2] and can be fully described by $p(Y_1 \mid X_1, X_2)$ and $p(Y_2 \mid X_1, X_2)$ for all values $X_1 \in \mathscr{X}_1$, $X_2 \in \mathscr{X}_2$, $Y_1 \in \mathscr{Y}_1$, and $Y_2 \in \mathscr{Y}_2$.

The modified IC with cooperative transmission introduces two pairs of auxiliary random variables: (U_1, V_1) and (U_2, V_2). In this work, U_1, U_2 denote the private message sending to the corresponding receivers. V_2 denote the public message sending to the two receivers. The message V_1 is generated to send message to receive V_2 cooperatively. Define $(n, K_{11}, K_{12}, K_{21}, K_{22}, \epsilon)$ code for the modified IC with cooperative transmission model as a set of $K_{21} \cdot K_{22}$ codewords $x_2^n(i, j) \in X_2^n$, and $K_{11} \cdot K_{12} \cdot K_{21} \cdot K_{22}$ codewords $x_1^n(i, j) \in X_1^n$, $i \in 1, 2, \ldots, K_{21}$, $j \in 1, 2, \ldots, K_{22}$, $k \in 1, 2, \ldots, K_{11}$, $l \in 1, 2, \ldots, K_{12}$ such that the average probability of decoding error is less than ϵ.

Denote the time-sharing random variable Q as defined in [2]. For notational simplicity, the following notation is used to replace the probability distributions of Q, U_1, U_2, V_1, and V_2

$$\begin{aligned} p(q) &= \Pr\{Q = q\}, \\ p(u_1 \mid q) &= \Pr\{U_1 = u_1 \mid Q = q\}, \\ p(u_2 \mid q) &= \Pr\{U_2 = u_2 \mid Q = q\}, \\ p(v_1 \mid q) &= \Pr\{V_1 = v_1 \mid Q = q\}, \\ p(v_2 \mid q) &= \Pr\{V_2 = v_2 \mid Q = q\}. \end{aligned} \tag{1}$$

Theorem 1. *Define* $Z := (Y_1, Y_2, X_1, X_2, U_1, V_2, U_1, V_2, Q,)$ *and let* $\mathscr{P}$ *be the set of distribution on* Z *that can be decomposed into the form*

$$\begin{aligned} &p(q)\, p(u_2 \mid q)\, p(v_2 \mid q)\, p(x_2 \mid u_2, v_2, q) \\ &\quad \times p(u_1 \mid u_2, v_2, q)\, p(v_1 \mid u_2, v_2, q)\, p(x_1 \mid u_1, v_1, q) \\ &\quad \times p(y_1 \mid x_1, x_2)\, p(y_2 \mid x_1, x_2). \end{aligned} \tag{2}$$

For any $Z \in \mathscr{P}$, let $S(Z)$ be the set of all quadruples (S_1, T_1, S_2, T_2) of nonnegative real numbers such that

$$S_1 \leqslant I\left(Y_1; U_1 \mid V_1 V_2 Q\right), \tag{3}$$

$$T_2 \leqslant I\left(Y_1; U_2 \mid U_1 V_1 Q\right), \tag{4}$$

$$S_1 + T_2 \leqslant I\left(Y_1; U_1 V_2 \mid V_1 Q\right), \tag{5}$$

$$S_2 \leqslant I\left(Y_2; U_2 \mid V_1 V_2 Q\right), \tag{6}$$

$$T_1 \leqslant I\left(Y_2; V_1 \mid U_2 V_2 Q\right) + R_0, \tag{7}$$

$$T_2 \leqslant I\left(Y_2; V_2 \mid U_2 V_1 Q\right), \tag{8}$$

$$S_2 + T_2 \leqslant I\left(Y_2; U_2 V_2 \mid V_1 Q\right), \tag{9}$$

$$S_2 + T_1 \leqslant I\left(Y_2; U_2 V_1 \mid V_2 Q\right) + R_0, \tag{10}$$

$$S_2 + T_1 + T_2 \leqslant I\left(Y_2; U_2 V_1 V_2 \mid Q\right) + R_0. \tag{11}$$

Furthermore, let S be the closure of $\bigcup_{Z \in \mathscr{P}} S(Z)$. Then any element of S is achievable for the modified IC with cooperative transmitter C_m.

Proof. It is sufficient to show the achievability of the interior elements of $S(Z)$ for each $Z \in p$. So, fix $Z = (Y_1, Y_2, X_1, X_2, U_1, V_2, U_1, V_2, Q)$ and take any (U_1, V_1, U_2, V_2) satisfying the constraints of the theorem. Given any $\eta \geqslant 0$, define L_a, N_a $(a = 1, 2)$ by

$$\frac{1}{n} \log L_a = S_a - \eta, \tag{12}$$

$$\frac{1}{n} \log N_a = T_a - \eta. \tag{13}$$

Encoding. To generate the codebook, first let $q^n \triangleq (q^{(1)}, q^{(2)}, q^{(3)} \cdot q^{(n)})$ be a sequence in Q^n chosen randomly according to $\Pi_{t=1}^n p(q^{(t)})$ and known to the transmitters and receivers. Considering the message exchange from T_{x_2} to T_{x_1}, note that

$$\begin{aligned} &p\left(u_1 \mid u_2, v_2, q\right) \\ &\quad = \sum_{u_2 \in U_2, v_2 \in V_2} p\left(u_1 u_2, v_2, q\right) p\left(u_2 q\right) p\left(v_2 q\right) p\left(q\right), \\ &p\left(v_1 \mid u_2, v_2, q\right) \\ &\quad = \sum_{v_1 \in U_2, v_2 \in V_2} p\left(v_1 \mid u_2, v_2, q\right) p\left(u_2 \mid q\right) p\left(v_2 \mid q\right) p\left(q\right). \end{aligned} \tag{14}$$

Then the codebook can be generated according to the distribution

$$\begin{aligned} &p\left(q\right) p\left(u_2 \mid q\right) p\left(v_2 \mid q\right) p\left(x_2 \mid u_2, v_2, q\right) \\ &\quad \times p\left(u_1 \mid u_2, v_2, q\right) p\left(v_1 \mid u_2, v_2, q\right) p\left(x_1 \mid u_1, v_1, q\right). \end{aligned} \tag{15}$$

If the probability of error could be arbitrarily small under such a message, the rates achieved will be (S_1, T_1, S_2, T_2) for the respective transmitter-receiver pairs $T_{x_1} \rightarrow R_{x_1}$, $T_{x_1} \rightarrow R_{x_2}, T_{x_2} \rightarrow R_{x_2}, T_{x_2} \rightarrow (R_{x_1}, R_{x_2})$.

Decoding. R_{x_1} and R_{x_2} decode independently, based on strong joint typicality. The inputs x_1^n, x_2^n are received at R_{x_1}, R_{x_2} as y_1^n, y_2^n according to the conditional distributions

$$p^n\left(y_1^n \mid x_1^n, x_2^n\right) = \prod_{t=1}^n p\left(y_1^{(t)} \mid x_1^{(t)}, x_2^{(t)}\right), \tag{16}$$

$$p^n\left(y_2^n \mid x_1^n, x_2^n\right) = \prod_{t=1}^n p\left(y_2^{(t)} \mid x_1^{(t)}, x_2^{(t)}\right), \tag{17}$$

where R_{x_1} attempts to recover (s_{11}, s_{21}) and R_{x_2} attempts to recover (s_{12}, s_{21}, s_{22}) based on the received signal y_1^n, y_2^n. Thus, the decoders at R_{x_1}, R_{x_2} are functions

$$\begin{aligned} &\psi_1 : Y_1^n \times Q^n \longrightarrow S_{11} \times S_{21}, \\ &\quad \psi_1\left(y_1^n, q^n\right) = \left(\psi_1^{11}\left(y_1^n, q^n\right), \psi_1^{21}\left(y_1^n, q^n\right)\right), \\ &\psi_2 : Y_2^n \times Q^n \longrightarrow S_{12} \times S_{21} \times S_{22}, \\ &\quad \psi_2\left(y_2^n, q^n\right) = \left(\psi_2^{12}\left(y_2^n, q^n\right), \psi_2^{21}\left(y_2^n, q^n\right), \psi_2^{22}\left(y_2^n, q^n\right)\right). \end{aligned} \tag{18}$$

When R_{x_1}, R_{x_2} receive the q^n and n-sequence y_1^n, y_2^n, respectively, it looks at the set of all input sequences (u_1^n, v_1^n, v_2^n) and (u_2^n, v_1^n, v_2^n), respectively. Thus R_{x_1}, R_{x_2} forms the set with the given q^n as follows:

$$\begin{aligned} S_1\left(y_1^w, q^n\right) &:= \left\{\left(u_1^n, v_1^n, v_2^n\right) : \left(y_1^n, u_1^n, v_1^n, v_2^n, q^n\right)\right\} \\ &\quad \in A_{\epsilon_d}^n\left(Y_1, U_1, V_1, V_2 \mid Q\right), \\ S_2\left(y_2^w, q^n\right) &:= \left\{\left(u_2^n, v_1^n, v_2^n\right) : \left(y_2^n, u_2^n, v_1^n, v_2^n, q^n\right)\right\} \\ &\quad \in A_{\epsilon_d}^n\left(Y_2, U_2, V_1, V_2 \mid Q\right). \end{aligned} \tag{19}$$

Valuation of Error Probability. To simplify the error probability calculation, defining each massage emitted to the channel yields the same error probability. First, the decoding error probability $\overline{P_{e1}^0}$ for R_{x1} is considered. Suppose that $Y_1 \in \mathscr{Y}_1$ was received by R_{x1}. Let $E_1(u_1 v_1)$ denote the decoding event (18), let define i, j, k, m as binary variables indicate the decode function of message u_1, v_1, u_2, v_2, respectively, where equals 1 indicates the decoding successfully, otherwise equals 0. Then it is easy to have

$$\begin{aligned} \overline{P_{e1}^0} &= \Pr\left\{E_1^c\left(11\right), \text{or} \bigcup_{im \neq 11} E_1\left(im\right)\right\} \\ &\leq \Pr E_1^c\left(11\right) + \sum_{im \neq 11} \left\{E_1\left(im\right) \mid u_1 \in U_1, v_1 \in V_1, v_2 \in V_2\right\}. \end{aligned} \tag{20}$$

It is easy to have

$$\Pr\left\{E_1^c\left(11\right) \leq \epsilon\right\}. \tag{21}$$

Considering the symmetry among the relevant random variables for the second part of (20), it is easy to have

$$\begin{aligned}&\sum_{im\neq 11}\Pr\{E_1(im)\mid u_1\in U_1,v_1\in V_1,v_2\in V_2\}\\&=(L_1-1)\Pr\{E_1(01)\mid u_1\in U_1,v_1\in V_1,v_2\in V_2\}\\&+(N_2-1)\Pr\{E_1(10)\mid u_1\in U_1,v_1\in V_1,v_2\in V_2\}\\&+(L_1-1)(N_2-1)\Pr\{E_1(00)\mid u_1\in U_1,v_1\in V_1,v_2\in V_2\}.\end{aligned}\tag{22}$$

Let us first evaluate $\Pr\{E_1(01)\mid u_1\in U_1,v_1\in V_1,v_2\in V_2\}$. It is easy to have

$$\begin{aligned}&\Pr\{E_1(01)\mid u_1\in U_1,v_1\in V_1,v_2\in V_2\}\\&\quad\le\exp\left[-n\left(H\left(U_1\mid Q\right)-\epsilon\right)\right]\\&\qquad\cdot\exp\left[n\left(H\left(U_1\mid V_1V_2Y_1Q\right)+\epsilon\right)\right]\\&\quad=\exp\left[-n\left(I\left(V_1V_2Y;U_1\mid Q\right)-2\epsilon\right)\right]\\&\quad=\exp\left[-n\left(I\left(Y;U\mid V_1V_2Q\right)-2\epsilon\right)\right].\end{aligned}\tag{23}$$

Using similar techniques for the other terms in (22) and substituting (12) and (13), it is easy to have

$$\begin{aligned}&\sum_{im\neq 11}\Pr\{E_1(im)\mid u_1\in U_1,v_1\in V_1,v_2\in V_2\}\\&\quad\le\exp\left[-n\left(I\left(Y_1;U_1\mid V_1V_2Q\right)-S_1+\eta-2\epsilon\right)\right]\\&\quad+\exp\left[-n\left(I\left(Y_1;V_2\mid U_1V_1Q\right)-T_2+\eta-2\epsilon\right)\right]\\&\quad+\exp\left[-n\left(I\left(Y_1;U_1V_2\mid V_1Q\right)-\left(S_1+T_2\right)+\eta-2\epsilon\right)\right].\end{aligned}\tag{24}$$

Since $\epsilon>0$ can be made arbitrarily small by letting n be sufficiently large, (3)–(5) yield

$$\sum_{im\neq 11}\Pr\{E_1(im)\mid u_1\in U_1,v_1\in V_1,v_2\in V_2\}\le\lambda,\tag{25}$$

where the parameter $\lambda\in(0,1)$. Then, substituting (22), (23), and (24), it is easy to have $\overline{P_{e1}^0}<2\lambda$. For R_{x_2}, $\overline{P_{e2}^0}$ denote the decode error probability. Let $E_2(u_2v_1v_2)$ denote the decoding event (18). Using similar techniques, it is easy to have the same result. To consider the achievability for the IC with transmitter cooperation channel, it is simply by using lemma 2.1 in [2]. Then it is demonstrated that if the rate pair (S_1,T_1,S_2,T_2) is achievable in the modified IC with transmitter cooperation channel, then the rate pair $(S_1,S_2+T_1+T_2)$ is achievable for the IC with transmitter cooperation channel. Furthermore, the convex hull is then achievable by standard time-sharing arguments. □

3. The Gaussian Interference Channel with Cooperative Transmission

Considering a memoryless Gaussian channel scenario, the received signals in standard form [1] for IC with transmitter cooperation are given by

$$\begin{aligned}Y_1&=X_1+\sqrt{a_{21}}X_2+Z_1,\\Y_2&=\sqrt{b_{12}}X_1+X_2+Z_2,\end{aligned}\tag{26}$$

where X_1,X_2 are the transmit signals with transmit power constraints P_1 and P_2, respectively. a_{21} and b_{12} are real numbers, and Z_1 and Z_2 are independent AWGN with zero means and variance N_1,N_2, respectively.

To simplify the notation, let $\gamma(x)=(1/2)\log_2(1+x)$. Let Q, the time-sharing random variable, be constant. Let define $\alpha,\beta\in\mathbb{R}$ and $\alpha,\overline{\alpha},\beta,\overline{\beta}\in[0,1]$ with $\alpha+\overline{\alpha}=1$, $\beta+\overline{\beta}=1$. Therefore, independent Gaussian codebooks of sizes $2^{nS_1},2^{nT_1},2^{nS_2}$, and 2^{nT_2} are generated according to i.i.d. Gaussian distributions $U_1\sim N(0,\alpha P_1)$, $V_1\sim N(0,\overline{\alpha P_1})$, $U_2\sim N(0,\beta P_1)$, and $V_2\sim N(0,\overline{\beta P_1})$, respectively.

At R_{x1}, (U_1,V_2) are decoded while U_2 is treated as noise. Since each mutual-information bound can be expanded in terms of entropies, the set of achievable rates (S_1,T_2) denoted here by C_1 is given by

$$\begin{aligned}S_1&\le\gamma\left(\frac{\alpha P_1}{N_1+a_{21}\beta P_2}\right),\\T_2&\le\gamma\left(\frac{a_{21}\overline{\beta}P_2}{N_1+a_{21}\beta P_2}\right),\\S_1+T_2&\le\gamma\left(\frac{\alpha P_1+a_{21}\overline{\beta}P_2}{N_1+a_{21}\beta P_2}\right).\end{aligned}\tag{27}$$

At R_{x2}, (U_2,V_1,V_2) are decoded while U_1 is treated as noise. This is a multiple access channel with a rate-limited link at transmitter, who has complete knowledge for T_{x_2}. This channel is a special case of the multiple access relay channel studied in [15]. The set of achievable rates (S_2,T_1,T_2) is denoted by C_2, where

$$\begin{aligned}S_2&\le\gamma\left(\frac{\beta P_2}{N_2+b_{12}\alpha P_1}\right),\\T_1&\le\gamma\left(\frac{b_{21}\overline{\alpha}P_1}{N_2+b_{12}\alpha P_1}\right)+R_0,\\T_2&\le\gamma\left(\frac{\overline{\beta}P_2}{N_2+b_{12}\alpha P_1}\right),\\S_2+T_1&\le\gamma\left(\frac{b_{12}\overline{\alpha}P_1+\beta P_2}{N_2+b_{12}\alpha P_1}\right)+R_0,\\S_2+T_1+T_2&\le\gamma\left(\frac{b_{12}\overline{\alpha}P_1+P_2}{N_2+b_{12}\alpha P_1}\right)+R_0.\end{aligned}\tag{28}$$

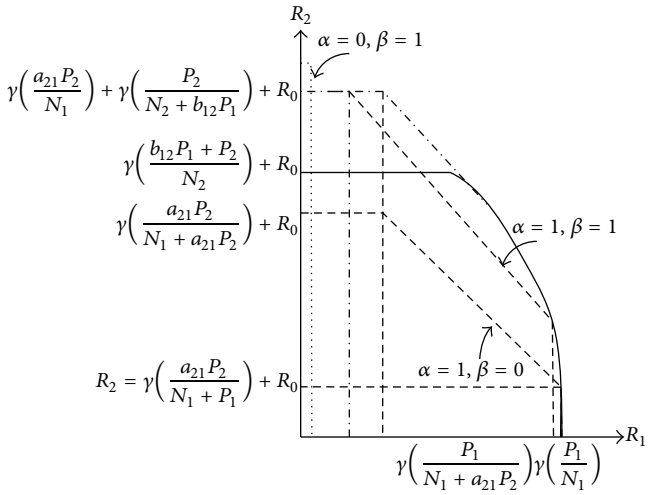


Figure 2: The union of rate region pentagons in weak interference scenario.

It is possible to use a Fourier-Motzkin elimination to verify the achievable rate region for arbitrary parameter value α, β, and the achievable (R_1, R_2) is formed according to (23)

$$R_{\alpha,\beta} = \left\{ (R_1, R_2) \;\middle|\; \begin{array}{l} R_1 \le \gamma\left(\dfrac{\alpha P_1}{N_1 + a_{21}\beta P_2}\right), \\ R_2 \\ \le \min\left\{\gamma\left(\dfrac{b_{12}\overline{\alpha}P_1 + P_2}{N_2 + b_{12}\alpha P_1}\right) + R_0, \right. \\ \qquad \gamma\left(\dfrac{a_{21}\overline{\beta}P_2}{N_1 + a_{12}\beta P_2}\right) \\ \qquad \left. + \gamma\left(\dfrac{b_{12}\overline{\alpha}P_1 + \beta P_2}{N_2 + b_{12}\alpha P_1}\right) + R_0\right\}, \\ R_1 + R_2 \le \gamma\left(\dfrac{\alpha P_1 + a_{21}\overline{\beta}P_2}{N_1 + a_{21}\beta P_2}\right) \\ \qquad + \gamma\left(\dfrac{b_{12}\overline{\alpha}P_1 + \beta P_2}{N_2 + b_{12}\alpha P_1}\right) + R_0 \end{array} \right\}. \tag{29}$$

The convex hull of the union of the rate region is over arbitrary and gives the complete achievability region. Since the union of the rate region $\bigcup_{0<\alpha,\beta<1} R_{\alpha,\beta}$ is convex, the convex hull is not needed. Therefore, the convex hull is not needed. A proof of this fact will be given in the following. Consider the regime where $a_{21}, b_{12} < 1$, that is, in weak interference and weak cooperation scenario. In this case, the constraint $R_2 \le \gamma((b_{12}\overline{\alpha}P_1+P_2)/(N_2+b_{12}\alpha P_1))+R_0$ is ignored. Then the achievable rate region can be represented in a more compact form as follows:

$$R_{\alpha,\beta} = \left\{ (R_1, R_2) \;\middle|\; \begin{array}{l} R_1 \le f_1(\alpha, \beta), \\ R_2 \le f_2(\alpha, \beta), \\ R_1 + R_2 \le f_3(\alpha, \beta) \end{array} \right\}, \tag{30}$$

where

$$\begin{aligned} f_1(\alpha, \beta) &= \gamma\left(\frac{\alpha P_1}{N_1 + a_{21}\beta P_2}\right), \\ f_2(\alpha, \beta) &= \gamma\left(\frac{a_{21}\overline{\beta}P_2}{N_1 + a_{12}\beta P_2}\right) + \gamma\left(\frac{b_{12}\overline{\alpha}P_1 + \beta P_2}{N_2 + b_{12}\alpha P_1}\right) + R_0, \\ f_3(\alpha, \beta) &= \gamma\left(\frac{\alpha P_1 + a_{21}\overline{\beta}P_2}{N_1 + a_{21}\beta P_2}\right) + \gamma\left(\frac{b_{12}\overline{\alpha}P_1 + \beta P_2}{N_2 + b_{12}\alpha P_1}\right) + R_0. \end{aligned} \tag{31}$$

To verify the union of rate region pentagons, it is easy to prove $f_1(\alpha, \beta)$, $f_2(\alpha, \beta)$, and $f_3(\alpha, \beta)$ are all continuous functions of α, β, as α, β are from 0 to 1. When α increases from 0 to 1, $f_1(\alpha, \beta)$ is monotonically increasing and both $f_2(\alpha, \beta)$ and $f_3(\alpha, \beta)$ are monotonically decreasing. On the other hand, when β increases from 0 to 1, $f_1(\alpha, \beta)$, $f_3(\alpha, \beta)$ are monotonically decreasing, while $f_2(\alpha, \beta)$ increasing. Therefore, as shown in Figure 2, the upper left corner point moves downward in the R_1-R_2 plane as α increases. Therefore, as shown in Figure 2, the upper left corner point moves downward in the R_1-R_2 plane as increases. Moreover, the lower right corner point moves downward and to the right as α increases and β decreases. Consequently, the union of theses expanded pentagons is defined by $R_1 \le (P_1/N_1)$, $R_2 \le ((b_{12}P_1 + P_2)/N_2) + R_0$, and the lower right corner points of the pentagons (R_1, R_2) with

$$\begin{aligned} R_1 &= \gamma\left(\frac{\alpha P_1}{N_1 + a_{21}\beta P_2}\right), \\ R_2 &= \gamma\left(\frac{b_{12}\overline{\alpha}P_1 + \beta P_2}{N_2 + b_{12}\alpha P_1}\right) + \gamma\left(\frac{a_{21}\overline{\beta}P_2}{N_1 + a_{12}\beta P_2 + \alpha P_1}\right) + R_0. \end{aligned} \tag{32}$$

To prove the convexity of the region, similar to [13], it is easy to have

$$R_2' = \frac{-\eta}{2^{2R_1} + \eta}, \qquad R_2'' = -\frac{2\eta \cdot \ln(2) \cdot 2^{2R_1}}{\left(2^{2R_1} + \eta\right)^2}, \tag{33}$$

where $\xi = \gamma((b_{12}\overline{\alpha}P_1 + \beta P_2)/(N_2 + b_{12}\alpha P_1))$, $\eta = \gamma(a_{21}\overline{\beta}P_1/(N_2 + b_{12}\alpha P_1))$. Since $\eta > 0$, it has $R_2' \le 0$ and $R_2'' \ge 0$. As a result, the curve (26) is concave. Therefore, in the weak interference regime where $a_{21}, b_{12} < 1$, the rate region is convex. Thus, convex hull is not needed. Thus the achievable rate region simplifies to

$$\begin{aligned} R_1 &\le \gamma\left(\frac{\alpha P_1}{N_1 + a_{21}\beta P_2}\right), \\ R_2 &\le \min\left\{\gamma\left(\frac{b_{12}P_1 + P_2}{N_2}\right) + R_0, \right. \\ &\qquad \gamma\left(\frac{b_{12}\overline{\alpha}P_1 + \beta P - 2}{N_2 + b_{12}\alpha P_1}\right) \\ &\qquad \left. + \gamma\left(\frac{a_{21}\overline{\beta}P_2}{N_1 + a_{21}\beta P_2 + \alpha P_1}\right) + R_0\right\}. \end{aligned} \tag{34}$$

Now, consider the strong interference and strong cooperation scenario, where $a_{21}, b_{12} > 1$. In this regime, as α, β are increase from 0 to 1, $f_1(\alpha, \beta)$ is monotonically increasing and both $f_2(\alpha, \beta)$ and $f_3(\alpha, \beta)$ are monotonically decreasing. Therefore, using similar techniques, the convex hull is proved to be unnecessary. Thus, the achievable rate region simplifies to

$$R_1 \le \gamma\left(\frac{P_1}{N_1}\right),$$
$$R_2 \le \gamma\left(\frac{b_{12}P_1 + P_2}{N_2}\right) + R_0, \tag{35}$$
$$R_1 + R_2 \le \gamma\left(\frac{a_{21}P_2}{N_1}\right) + \gamma\left(\frac{b_{12}P_1}{N_2}\right) + R_0.$$

So far, obtained achievable rate regions are obtained for regimes $a_{21}, b_{12} < 1$ and $a_{21}, b_{12} > 1$ as in (29) and (30), respectively.

4. Numerical Results

In this section, to evaluate the achievable rate region, the results are illustrated in Figures 3 and 4, respectively. As a numerical example, Figure 3 shows the achievable rate region of a Gaussian IC in the weak interference regime, with $P_1 = P_2 = 6$, $a_{21} = b_{12} = 0.55$. The red line denotes the classic Han and Kobayashi common-private power splitting scheme [2], and the blue line denotes the IC with transmitter cooperation. The cooperation builds a little contribution for R_2. However, it sacrifices the rate for R_1, because T_{x_1} spend more transmit power to the cooperative receiver R_{x_2} instead of the direct link receiver R_{x_1}. Since the interference link gain is smaller than that of direct link, the cooperation gain of T_{x_1} on R_{x_2} is tiny as well. This is obviously different in the case of strong interference regime as shown in Figure 4.

In the strong interference regime, the capacity region of IC with transmitter cooperation is achieved by transmitting only at the cross-link, that is, the private message from T_{x_1} to R_{x_2} and the common message from T_{x_2} to R_{x_1}. In the strong interference scenario, the cooperation link increases the capacity by helping the private information decoding at R_{x_2}. In fact, a conferencing link R_0 increases the sum capacity by exactly R_0. As a numerical result, Figure 4 illustrates the capacity region of an IC with transmitter cooperation in strong interference regime with and without cooperation. The channel parameters are set to be $P_1 = P_2 = 2$, $a_{21} = b_{12} = 1.5$. The results indicate that the achievable rate region increases when the R_0 increases. The capacity region without cooperation (H-K scheme) is the red pentagon. The capacity expands to the blue pentagon region with cooperation and $R_0 = 0.4$. If $R_0 \ge (1/2)\log(1 + P_2)$, the T_{x_1} can fully cooperative R_{x_2}. In this case, the achievable rate region became the black rectangular.

5. Conclusions

In this paper, rate region of the two-user IC with transmitter cooperation is studied and derived into Gaussian channel scenario. A potentially more efficient transmission model is proposed. In the weak interference scenario, a novel achievable rate region is provided to help the cooperative link by splitting the transmit power for the other direct link. In the strong interference scenario, a more flexible rate region is defined and proved. The larger rate region is achievable as the conferencing rate increases, which also improves its sum rate.

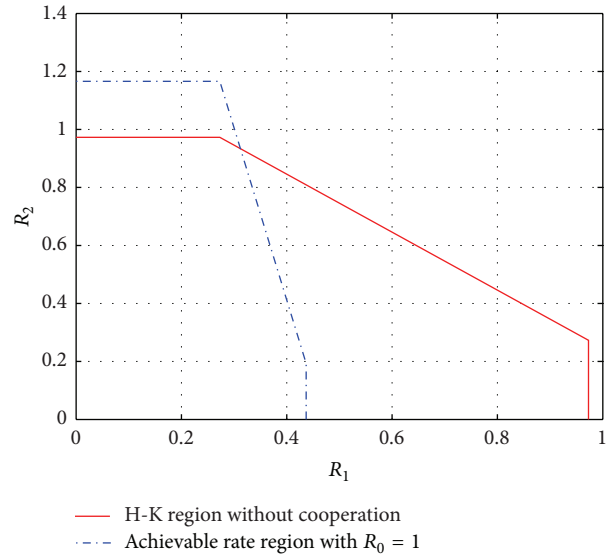


Figure 3: The achievable rate region for weak interference with $P_1 = P_2 = 6$, $a_{21} = b_{12} = 0.55$.

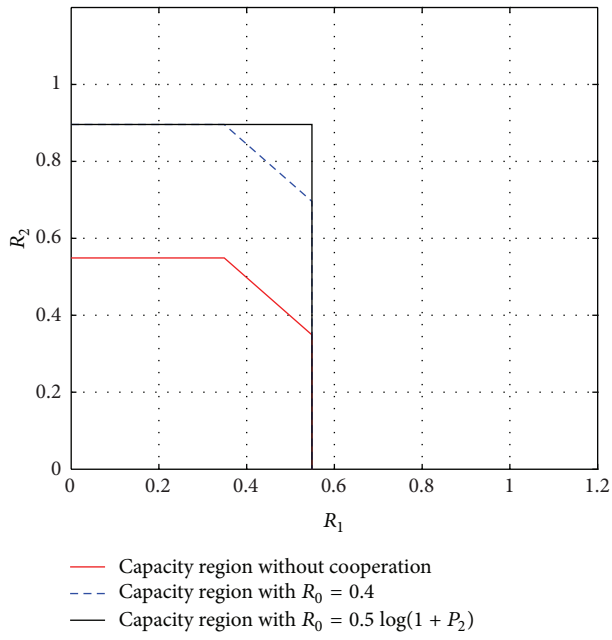


Figure 4: The achievable rate region for weak interference with $P_1 = P_2 = 2$, $a_{21} = b_{12} = 0.55$.

Acknowledgments

The authors wish to thank Zhang TianKui. This work was supported by National Natural Science Foundation of China nos. 61250005 and 61340025. JiangXi Natural Science Foundation 20122BAB2111015 and 20132BAB211035, Jiangxi Provincial Department of Education Youth Science Foundation GJJ13007, Jiangxi postdoctoral Merit-funded project funds 2013KY07, and China postdoctoral Science Foundation on the 49th Grant program no. 2013M541875.

References

[1] A. B. Carleial, "Interference channels," *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 60–70, 1978.

[2] T. S. Han and K. Kobayashi, "A new achievable rate region for the interference channel," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 49–60, 1981.

[3] Z. Tiankui, S. Xiaochen, C. Laurie et al., "Vector perturbation based adaptive distributed precoding scheme with limited feedback for CoMP systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, article 8, 2011.

[4] D. Lee, H. Seo, B. Clerckx et al., "Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges," *IEEE Communications Magazine*, vol. 50, no. 2, pp. 148–155, 2012.

[5] Z. Hu, T. Zhang, and C. Feng, "Study on codeword selection for per-cell code-book with limited feedback in CoMP systems," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '13)*, April 2013.

[6] N. Devroye, P. Mitran, and V. Tarokh, "Achievable rates in cognitive radio channels," *IEEE Transactions on Information Theory*, vol. 52, no. 5, pp. 1813–1827, 2006.

[7] A. Jovicic and P. Viswanath, "Cognitive radio: an information theoretic perspective," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '06)*, pp. 2413–2417, July 2006.

[8] S. I. Gel'fand and M. S. Pinsker, "Coding for channel with random parameters," *Problems of control and information theory*, vol. 9, no. 1, pp. 19–31, 1980.

[9] W. Wu, S. Vishwanath, and A. Arapostathis, "Capacity of a class of cognitive radio channels: interference channels with degraded message sets," *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4391–4399, 2007.

[10] J. Jiang and Y. Xin, "On the achievable rate regions for interference channels with degraded message sets," *IEEE Transactions on Information Theory*, vol. 54, no. 10, pp. 4707–4712, 2008.

[11] I. Marić, R. D. Yates, and G. Kramer, "Capacity of interference channels with partial transmitter cooperation," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3536–3548, 2007.

[12] D.-W. Seo, S.-W. Jeon, S.-Y. Chung, and J. Kim, "Rate enhancement for the Gaussian Z-interference channel with transmitter cooperation," *IEEE Communications Letters*, vol. 14, no. 9, pp. 821–823, 2010.

[13] L. Zhou and W. Yu, "Gaussian Z-interference channel with a relay link: achievable rate region and asymptotic sum capacity," in *Proceedings of the International Symposium on Information Theory and its Applications (ISITA '08)*, pp. 1–6, Auckland, New Zealand, December 2008.

[14] S. Zhao, T. Zhang, and Z. Zeng, "The achievable generalized degrees of freedom of interference channel with orthogonal relay," in *Proceedings of the IEEE GLOBECOM Workshops (GC Wkshps)*, pp. 741–745, Miami, Fla, USA, December 2010.

[15] D. H. Sato, "Capacity of the gaussian interference channel under strong interference," *IEEE Transactions on Information Theory*, vol. 27, no. 11, pp. 786–788, 1981.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 532986, 10 pages
http://dx.doi.org/10.1155/2013/532986

Review Article

A Comparison of Clock Synchronization in Wireless Sensor Networks

Seongwook Youn

Computer Science Department, University of Southern California, 941 Bloom Walk, SAL 300, Los Angeles, CA 90089-0781, USA

Correspondence should be addressed to Seongwook Youn; seongwook.youn@gmail.com

Received 8 June 2013; Revised 20 November 2013; Accepted 21 November 2013

Academic Editor: Hongli Xu

The recent advances in microelectro devices have led the researchers to an area of developing a large distributed system that consist of small, wireless sensor nodes. These sensor nodes are usually equipped with sensors to perceive the environment. Synchronization is an important component of almost all distributed systems and has been studied by many researchers. There are many solutions for the classical networks, but the traditional synchronization techniques are not suitable for sensor networks because they do not consider the partitioning of the network and message delay. Additionally, limited power, computational capacity, and memory of the sensor nodes make the problem more challenging for wireless sensor networks. This paper examines the clock synchronization issues in wireless sensor networks. Energy efficiency, cost, scalability, lifetime, robustness, and precision are the main problems to be considered in design of a synchronization algorithm. There is no one single system that satisfies all these together. A comparison of different clock synchronization algorithms in wireless sensor networks with a main focus on energy efficiency, scalability, and precision properties of them will be provided here.

1. Introduction

Wireless sensor networks are the networks that consist of mobile wireless computing devices, in which these devices are usually equipped with sensors to perceive the environment. Along with the recent advances in technology and the increasing demand, sensor networks are now being widely used in many applications. Wireless sensor networks have many applications including environmental monitoring, health monitoring, inventory location monitoring, and objects tracking. Features of a sensor network, such as size (number of nodes), density, and connectivity, vary depending on the application. Sensor nodes in the network are mostly mobile devices equipped with limited power and computation capabilities. Hence, a reasonable ordering of events in such environments is a challenging task.

This paper examines the clock synchronization issues in ad hoc and sensor networks [1]. Clocks can be out of synchronization in two ways: shifting (clock offset or phase offset) or drifting (clock skew-oscillator's frequency offset). In the case of shifting, they run at the same frequency, but their clock readings differ by a constant value—the offset between the clocks. In the case of drifting, they run at different frequencies. Synchronizing drifting clocks is much more costly and difficult than synchronizing two shifting clocks. Clocks of nodes may run at slightly different frequencies, which is the main reason why clock offsets keep drifting away due to the imperfections in the quartz crystal. Adjusting clock skew can guarantee long-term reliability of synchronization and reduce the number of message exchanges. Some of the previous algorithms adjust the frequency offset (clock skew) relative to a certain frequency [2, 3]. Maggs et al. [4] proposed a consensus clock synchronization that provides internal synchronization to a virtual consensus clock. It is sensitive to the limited resources available to sensor nodes and is robust to many of the challenges faced in dynamic ad hoc networks.

Feedback-based synchronization (FBS) scheme to compensate the clock drift caused by both internal perturbation and external disturbance was proposed by Chen et al. [5]. It showed that FBS is much more robust than the delay measurement time-synchronization (DMTS) protocol. Misra and Vaish [6] suggested a reputation-based role assigning scheme for RBAC. The main objective of this scheme is to manage reputation locally with minimum communication

and delay overhead and to assign appropriate role or level to the deserved nodes in order to increase the throughput of overall network. Their scheme showed the increase in throughput by around 32% at the consumption of little more energy. Liu et al. [7] proposed a Kalman filter based advanced SCTS (ASCTS) mechanism. They proved the close relationship between the basic phase locked loop (PLL) employed by SCTS and the Kalman filter employed by ASCTS.

Many synchronization techniques have been proposed in literature for either central [8, 9] or distributed [10–12] systems. However, these classical synchronization techniques are not suitable for wireless sensor networks, since they do not take into account the partitioning of the network and the message delay.

There are mainly six requirements to be considered in design of a synchronization algorithm: energy efficiency, cost, scalability, lifetime, robustness, and precision. There is no one single system that satisfies all these together. The author will be providing a comparison of different clock synchronization algorithms in wireless networks along three axes: energy efficiency, scalability, and precision properties.

The paper is organized as follows. I will be first talking about the synchronization problem in the next subsection. Then, I will give an overview of traditional synchronization methods in Section 2. After a brief introduction to wireless sensor networks, challenges, design issues, sources of error, and requirements of a synchronization method for wireless sensor networks will be analyzed in Section 3. I will present four different synchronization methods and then compare them based on three aspects.

2. Traditional Synchronization Methods

Possible set of solutions to the problem depends on the system in use. For instance, the problem can be addressed by a centralized server in centralized systems. Two such systems are described by Cristian [8] and Gusella and Zatti [9]. The method presented in [8] depends on a central time server that is connected to an accurate time source like UTC (Coordinated Universal Time). To get the actual time, the client sends a request to the server. Receiving the request, the server prepares a response by appending its current clock time t to it and sends it to the client. Then the client adjusts its time as $t + rrt/2$, where rrt is the round trip time elapsed for the message to travel from and then back to a sender.

Rather than the clients asking for time as in [8], a time server polls the machines periodically in [9]. The procedure starts with the server requesting for current clock times of the clients. Once all the responses are received, the server finds the time of each client by using the round trip times of the messages, a method similar to [8]. The actual time is then calculated in the server by averaging those values as well as its own time. Instead of sending the calculated time, the server sends to each client the amount of time that it should adjust. The idea behind this is to avoid the errors introduced by the round-trip time estimations.

The clock synchronization problem gets more complicated in distributed systems due to lack of a global clock. There are two concepts to be considered in this case, either to synchronize the physical clock or the logical clock. In physical clock synchronization, the goal is to bring together the physical clocks of each machine to a very similar point; whereas in logical clock synchronization, gravity is the accurate ordering of relevant events.

Network Time Protocol (NTP) is the most commonly used method on the Internet for physical clock synchronization [12]. NTP is a layered client-server architecture based on UDP message passing. It operates with a hierarchy of levels, where levels are assigned a number called the stratum. At the lowest level are the stratum 1 (primary) servers, which are directly synchronized to national time services. In the next level, there are the stratum 2 (secondary) servers that are synchronized to stratum 1 servers. And the hierarchy continues the same way until the highest level.

In some distributed systems, it is more important to have a consistent and logical ordering of events, rather than knowing the actual occurrence time for each individual event. For such systems, it is not required to have absolute clock synchronization as it was the case in physical clock synchronization. Lamport [11] and Fidge [10] are the two most remarkable solutions for logical clock synchronization in distributed systems.

Lamport [11] defines an ordering of events using the concept of causality. If an event a could have affected the outcome of event b, then it is referred as event a "happened before" event b. The partial ordering of events is discussed in the paper, which is obtained by the "happened before" relation. For the partial ordering, there are two rules to be considered. The first is to increment the local clock between any two successive local events. The second is, upon receiving a message from another process with a local timestamp t of that sending process, to set the local clock greater than or equal to the maximum value of either t or the local clock value. Finally, they use these logical clocks to obtain total ordering across all processes and events.

Fidge [10] also defines a partial ordering of events using the causality concept. However, rather than using a single value for each timestamp, they choose to use a vector of values. The vector is initially set to $(0, 0, \ldots, 0)$, where each index corresponds to a processor. In case of a local event at processor P_i, the value at index i is incremented. When a processor P_i receives a message from processor P_j with timestamp vector T, P_i sets the time in each index to maximum value of either the corresponding value of T or the local vector value. The advantage of keeping a vector of timestamps and maximizing it among processors is that it allows ordering not only the events within a process, but also the events in different processes.

3. Synchronization Issues in Wireless Sensor Networks

Sensor networks are the networks that consist of mobile wireless computing devices that are equipped with sensors to perceive the environmental conditions, such as temperature, pressure, and humidity. The traditional synchronization techniques described above are not suitable for such networks.

In this section, I will first talk about the sensor networks in general and then discuss the main challenges of synchronization in sensor networks, which involves desired properties and design principles of a synchronization scheme, as well as main possible resources of error.

Although the initial settings might be the same, real clocks at different computing devices can be different due to some variances in the counting rates of the clocks. For some node i in the network, a hardware oscillator assisted computer clock installed in that node implements an approximation $C(t)$ of real-time t as

$$C_i(t) = a_i t + b_i, \tag{1}$$

where a_i denotes the angular frequency (or rate) of the hardware oscillator and b_i denotes the difference to the real time t. In literature, this angular frequency is usually referred as clock drift or skew and the difference as clock offset [13]. In a perfect case, the rate of a clock (dC/dt) would be equal to zero. However, due to the environmental conditions, such as temperature, pressure, and humidity, the clocks are subject to some drift with a maximum value of ρ such that

$$1 - \rho \leq \frac{dC}{dt} \leq 1 + \rho. \tag{2}$$

Although having different values for different computers, today's clock hardware typically provide a value of 10^{-6} for ρ [14], which means the clocks drift away from each other by at most one second in ten days. Behaviors of clocks with different dC/dt values are the standard timescale used by most of the nations in the world, which is based on the Earth's rotation about its axis [15].

Clock synchronization problem deals with ways of bringing the clocks of different computers (or processes, devices...) close to each other by communicating among them. More precisely, the clock synchronization problem aims to equalize $C_i(t)$ for all the nodes $i = 1, \ldots, n$ or some subset of nodes in the network. Adjusting the clock values for once is not enough, since the clocks will be drifting away again later. Hence, one can choose to either equalize the rates along with the offset or apply the synchronization repeatedly.

3.1. Sensor Networks. The recent advances in small electro devices have aroused interest in the development of large and distributed systems of small, wireless sensor nodes that communicate with each other. These sensor nodes mostly consist of components with sensing, data processing, and communicating capabilities. Although being limited in power, computational capacities, and memory, they can be used collaboratively to monitor the environmental conditions. Hence, a sensor network is network that is composed of a large number of spatially distributed sensor nodes that monitor physical or environmental conditions.

Types of sensors in a sensor network include low sampling rate magnetic, thermal, visual, infrared, acoustic, and radar. These sensors are able to monitor a wide variety of ambient conditions such as temperature, humidity, vehicular movement, lightning condition, pressure, soil makeup, noise levels, the presence or absence of certain kinds of objects, mechanical stress levels on attached objects, and speed, direction, and size of an object [16]. Initially, the development of wireless sensor networks originated by military-oriented applications (i.e., monitoring forces, battlefield surveillance). However, today wireless sensor networks are being used in various domains for many other applications. Some of these domains and sample applications in those domains have been listed below.

(i) Military: battlefield surveillance, targeting, monitoring forces, equipment and ammunition, and battle damage assessment.

(ii) Environmental: fire, flood, earthquake detection, and biocomplexity mapping.

(iii) Health: tracking and monitoring doctors/patients in a hospital, human physiological data telemonitoring.

(iv) Scientific: space and undersea exploration, cosmic radiation, and nuclear reactor control.

(v) Home: home automation, smart environment design.

(vi) Commercial: virtual keyboards, monitoring product quality, interactive museums, and detecting and monitoring car thefts.

Depending on the application, the number of nodes in a sensor network can be in the order of hundreds or thousands. These nodes are usually inaccessible and unattended, making the network topology prone to dynamic changes. Thus, robustness and self-configuration are the important requirements to be considered in design of a sensor network. Energy efficiency is another important concern for wireless sensor networks, since nodes are often inaccessible and have small sizes, which causes them to possess or produce limited power.

3.2. Desired Properties. In this section, the main requirements of a synchronization method for wireless sensor networks are listed and discussed. There is a trade-off among each of these features, and no one single system satisfies all these together [3, 13, 17].

(i) Energy efficiency: sensors in a wireless network are small and untethered devices. Hence, a synchronization scheme should take into account the limited energy resources and utilize energy in an efficient way.

(ii) Scalability: sensor networks usually consist of hundreds to thousands of nodes. Hence, a synchronization scheme should be able to scale well with increasing node density or number of nodes.

(iii) Precision: refers to how much the local clocks differ from either each other or an external standard clock. Desired precision can range from milliseconds to seconds depending on the application. For some applications, it is enough to only have a reasonable ordering of events, whereas for others a very high precision might be required.

(iv) Robustness: sensors in the network are usually mobile and untethered. There is great chance for a node

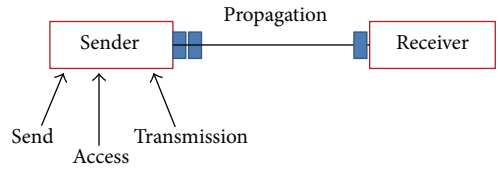


Figure 1: Sources of delay while transmitting a packet from a source to a destination in the wireless sensor network.

to fail or left unattended for a long time. Hence, robustness to such failures is a desired property for a synchronization method.

(v) Lifetime: amount of time for synchronization to last. Depending on the scheme, it might be either instantaneous or as long as the network operates.

(vi) Scope: for some applications, it is enough to synchronize only a subset of the network at a time, whereas for others a global synchronization might be required. Scope defines the geographic span of nodes that need to be synchronized.

(vii) Cost: sensor nodes are usually small and low cost devices. So, it is not a reasonable thing to equip a node with expensive hardware, such as a GPS receiver. Cost can play an important role for the overall system, considering that the number of sensors can get extremely large.

3.3. Main Sources of Errors. In this section, main possible sources of error in a synchronization algorithm are presented. When two nodes want to synchronize, they need to communicate with each other by message exchange. However, there are different types of delays on the path from the sender of a message to the receiver that cause errors in clock estimations. Figure 1 explains a schematic representation of where these delays happen. In the remaining, I discuss each error source individually.

(i) Send time: corresponds to the time spent in the sender node for constructing the packet at the application layer and sending it to the MAC layer. This time depends on the operation system being used, hence causing a nondeterministic delay on sender.

(ii) Access time: the time spent at the MAC layer waiting for access to the transmission channel. This delay plays an important role for most of the systems.

(iii) Transmission time: corresponds to the time taken for a message to be transmitted on the wireless link. This is a deterministic delay and can be estimated by the length of the message and the speed of the radio.

(iv) Propagation time: this is the time spend on the wireless link from sender to the receiver, once the packet leaves the sender. This delay is also deterministic and depends on the distance between the nodes.

(v) Reception time: this refers to the time spent on the receiver for receiving the packet and passing it to the MAC layer. This time corresponds to the transmission time on the receiver side and can be estimated in a similar manner.

(vi) Receive time: corresponds to the time for processing the incoming packet at the receiver and sending it to the application layer. This time can be thought as dual of send time at the sender.

3.4. Design Principles. In [3], Elson and Romer discussed five main design properties of a wireless network synchronization algorithm. The first principle is a multimodal, tiered, and tunable design. As mentioned in Section 3.2, there is always a trade-off among the desired attributes of an algorithm. According to this, the first principle says that synchronization should contain different models with different attributes, so that one can tune it by changing a set of parameters for different applications.

The second principle offers each node in the network to store relative drift and phase information locally, rather than keeping a global timescale. This kind of design purveys the error to be dependent on the distance between the nodes, not the distance to a master clock.

The third principle is postfacto synchronization, which has been widely used by many algorithms [2, 17, 18]. Postfacto synchronization offers the local node clocks to run asynchronous until the timestamps of different clocks need to be compared. This provides a lot of energy savings by forcing the resources to be used only when required. Final two principles involve being adaptable to different applications and exploiting the domain knowledge.

Apart from these, there are certain other concerns that should be taken into account for the design. I will be mentioning only two of them that I will give better understanding for further reading of this paper and refer the reader to [3, 15] for more detailed analysis. First issue is single-hop versus multihop synchronization. Most of the traditional methods assume that all the nodes in the network can communicate with each other or the network topology has lowlatency. However, for sensor networks, this may not be always the case that there might be more than one broadcast domains. Nodes in different domains can communicate with each other via routers that appear in both domains (at the intersection of two domains).

Second important concern is static versus dynamic network topology. Sensors are usually moving devices in the network. Hence, the network topology is subject to change frequently. Also, due to limited power or range of sensor, it is possible to have link failures any time in the network. A synchronization scheme should be able to adapt these changes dynamically.

4. Methods for Synchronization in Wireless Sensor Networks

Four different synchronization methods are presented in here. These methods are selected because either they are one of the first systems proposed for sensor networks or most widely used/referred systems.

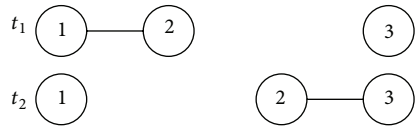


Figure 2: Network topology and message flow in ad hoc networks.

4.1. Synchronization in Ad Hoc Networks. Römer [14] proposed a time synchronization algorithm for ad hoc networks. In ad hoc networks, nodes are usually mobile and have limited communication range. Thus, the network topology is prone to frequent changes. This has been depicted in Figure 2 by an example. At time t_1, only nodes 1 and 2 are within communication range of each other. Then, node 2 moves closer to node 3; hence, at time t_2, only nodes 2 and 3 can communicate. There is no time between t_1 and t_2, in which nodes 1 and 3 can communicate to each other directly or indirectly. In contravention of this, nodes 1 and 3 can communicate with each other in a unidirectional way: at time time t_1 node 1 sends a message to node 2, which is stored at node to and then forwarded to node 3 at time t_2.

Traditional methods assume that nodes in the network can send messages to each other periodically and the round trip time between two nodes can be estimated. However, these assumptions no longer hold for ad hoc networks. According to Römer, an ad hoc network synchronization algorithm should not require a particular network topology and be able to handle all kinds of partitioning. This in mind, Römer, makes two assumptions about the network. First assumption is that the maximum clock drift ρ_i is known for all computer clocks. Secondly, it is assumed that if two adjacent nodes start to communicate with each other, then the connection lasts long enough to allow the two nodes to exchange one more (additional) message.

The main idea of the proposed algorithm is to transform the timestamp generated by an untethered local clock of a sending node to the local clock of the receiver node. According to this, if a node wants to send a message to another node in the network, it creates a timestamp using its own local clock and attaches it to the message. When the message is received by the other node, the timestamp is first transformed from local time of sender to UTC and then from UTC to the local time of the receiver. Due to various reasons, such as unpredictability of computer clocks, these transformations cannot be done exactly. So, the algorithm uses lower and upper bounds for the interval of the exact time. The relationship between the computer clock difference ΔC and the real time difference Δt can be given as

$$1 - \rho \leq \frac{\Delta C}{\Delta t} \leq 1 + \rho, \tag{3}$$

which can be transformed into

$$\begin{aligned} (1-\rho)\,\Delta t \leq \Delta C \leq (1+\rho)\,\Delta t \\ \frac{\Delta C}{1+\rho} \leq \Delta t \leq \frac{\Delta C}{1-\rho}. \end{aligned} \tag{4}$$

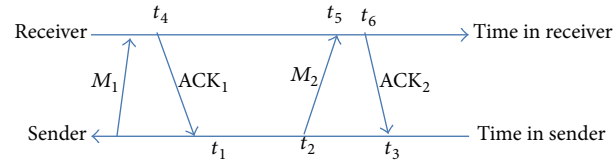


Figure 3: Estimation of message delay.

These lead to the consequence that the computer clock difference ΔC lies within $[(1-\rho)\Delta t \leq \Delta C \leq (1+\rho)\Delta t]$. Similarly, the interval for real time difference Δt can be given by $[\Delta C/(1+\rho), \Delta C/(1-\rho)]$. Let ρ_s and ρ_r be the maximum clock drifts of the sender and the receiver nodes, respectively. Then, the algorithm runs as follows.

(i) Sender node s generates a message with its local timestamp.

(ii) When the receiver node r gets the message, it first estimates the computer clock difference ΔC from the real time interval as $[\Delta C/(1+\rho_s), \Delta C/(1-\rho_r)]$.

(iii) Then, the receiver calculates the computer clock difference relative to its local time as $[\Delta C((1-\rho_r)/(1+\rho_s)), \Delta C((1+\rho_s)/(1-\rho_r))]$.

There is the message delay d that the transformation algorithm needs to take into account in order to find these intervals exactly. However, this delay is not constant for all message exchanges. Therefore, they choose to estimate a delay interval for each message independently. The delay for message M_2 in terms of the receiver's clock using two consecutive message exchanges can be given by

$$0 \leq d \leq (t_5 - t_4) - (t_2 - t_1)\,\frac{1-\rho_r}{1+\rho_s}. \tag{5}$$

However, this estimation has two disadvantages. First, the time between two consecutive messages can be quite high, resulting in the values for $(t_5 - t_4)$ and $(t_2 - t_1)$ to be large. Second, the delay is calculated by using two different message exchanges. This requires keeping track of state information in case of multiple message transfers among many other nodes. Römer proposes to avoid the first disadvantage by sending dummy messages when if these values sum up to be large. For the second case, he offers to delete the state information at the cost of a later dummy message. The estimation process of message delay is shown in Figure 3.

The algorithm described above works for the message exchanges between two adjacent nodes in the network. However, as mentioned before, the message exchanges between two nodes can be unidirectional and delayed. For a message to be transfered from node 1 to node n with message exchanges in the sequence of nodes $1, 2, \ldots, n$, the round trip time between each pair of nodes and the idle time of the node are also maintained to be used in the time calculations.

Once the time intervals, $[t_1, t_2]$ and $[t_3, t_4]$, for two different events are computed as described above, finding whether either one of them happened before the other can be done by comparing these intervals; that is, $[t_1, t_2]$ ¡ $[t_3, t_4]$.

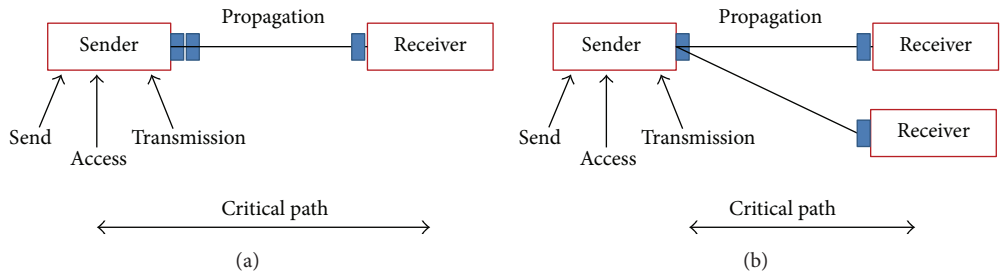


Figure 4: (a) Critical path for traditional methods. (b) Critical path for RBS.

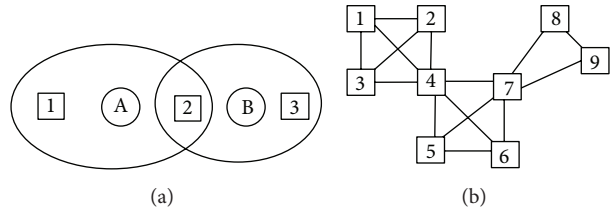


Figure 5: (a) A sample topology with two broadcast domains. (b) A logical topology.

The answer is either "yes" if $t_2 \text{ ¡ } t_3$ or "no" if $t_4 \text{ ¡ } t_1$ or "maybe" otherwise. Also, the distance between these two intervals can also be given as

$$\left|[t_1, t_2] - [t_3, t_4]\right| \leq \frac{\left(\max\left(t_4, t_2\right) - \min\left(t_3, t_1\right)\right)}{\left(1 - \rho\right)}. \quad (6)$$

The prototype experiments presented in the paper show that the performance of the system decreases in proportional to the increase in timestamp intervals, which may be caused by either the age of the timestamp or the number of hops used to pass the message.

4.2. Reference Broadcast Synchronization (RBS). Elson et al. [2] present a synchronization scheme called Reference Broadcast Synchronization (RBS) for wireless sensor networks. The main idea of the algorithm lies under the assumption that when a node broadcasts a reference beacon to its neighbors, the receivers will get the message approximately at the same time. As opposed to the traditional synchronization methods that try to synchronize the time between the sender and receiver nodes, they propose to synchronize a set of receivers with each another. The behind principle for doing this is to remove the errors caused by the sender's non-determinism. Elson defines the critical path in a message exchange to be the path from the sender node to the receiver node that includes all sources of error (send, access, transmission, propagation, reception, and receive delays) as explained earlier in Section 3.2. Since RBS takes into account only the arrival times of a message in each receiver, the three error sources (send, access, and transmission delays) are eliminated by default. Figure 4 shows the critical path of traditional methods and RBS.

The algorithm estimates the phase offset between two receiver nodes i and j as follows. When the sender broadcasts m reference messages, the n receivers record the arrival time of the message according to their local time. After the receivers exchange the reordered times, the receiver i can compute the phase offset between the receiver j by

$$B_{i,j} = \frac{1}{m}\sum_{k=1}^{m}\left(T_{jk} - T_{ik}\right), \quad (7)$$

where T_{jk} refers to the time of node j at receiving the message k. The right side of the equation represents the average of all phase offsets between i and j for all m messages. This averaging provides better precision statistically.

To find the clock drift, they propose to perform a least-squares linear regression over the phase offsets of all exchanged messages. This implies fitting a best line to all the phase offset observations of two receiver nodes over time. The slope and the intercept of the fitted line then provide the drift and the offset of one node with respect to the other. Such an approach allows the relative clock values to be computed even in case of missing messages.

As explained, the algorithm finds the relative clock offset and the drift from over multiple message exchanges. This provides postfacto synchronization by saving energy for the cases where synchronization is needed infrequently. When desired, nodes can turn on their power and transfer messages until the best fit line is computed reasonably. Storing the relative clock drifts and the offsets with respect to the other nodes, rather than correcting the local clocks according to a global time scale, also provides important energy savings.

Elson et al. [2] show that the proposed method can be generalized to clock synchronization in multihop networks. Consider the example on the left of Figure 5. The larger circles represent two different broadcast domains, rectangles represent nodes in corresponding domains, and A and B are the reference nodes. According to this topology, nodes 1 and 2 cannot communicate with each other directly. To compare the two events, e1 on node 1 and e3 on node 3, node 2 first

uses A's reference broadcast to convert the clock value of e1 to its own clock value. Similarly, using B's reference broadcast, node 2 converts this value to node 3's clock value. The final value can then be compared with e3, since both are based on node 3's clock.

The technique explained above can be extended to networks of more than two domains with multiple gateways. To do this, the network topology can be represented by a logical graph, in which there is a link between two nodes if they receive a common broadcast. See the graph on the right of Figure 5 for a sample logical topology graph. A series of conversions can be performed on this graph by finding a shortest path between the nodes. Also, the weights of the links can be used for representing the quality of the conversion.

4.3. Timing-Sync Protocol (TSPN). Ganeriwal et al. [18] propose a synchronization scheme called Timing-Sync Protocol for sensor networks (TSPN). One of the main concerns of the paper is to achieve high accuracy even for a large number of nodes being deployed. The algorithm presented is based on a sender-receiver synchronization approach. They argue that this classical approach gives better results than synchronizing only the receivers with each other (i.e., RBS [2]). The principle is that messages are time stamped at the MAC layer, which removes sources of error at the sender and the receiver.

There are two assumptions about the network that make the proposed algorithm work. First assumption is that every node in the network knows the set of nodes that it can communicate with. Secondly, they suppose that it is possible to create a spanning tree in the network by using the bidirectional links among the nodes. Based on these, there are two main steps of the algorithm, which involves "level discovery phase" and "synchronization phase."

The first step of the algorithm is the "level discovery phase" to create a hierarchical topology of the network. A root node is assigned at the level 0, and it initiates the phase by broadcasting a level discovery message. Nodes that receive this message assign themselves to one level greater than the received message. These nodes then broadcast a new level discovery message that contains the level of the broadcasting node. For instance, a node that receives a level discovery message from level 0 sets its level to 1 and then broadcasts another level discovery message affirming that it is from level 1. This scheme continues until every node in the network establishes a level.

The second step of the algorithm is the "synchronization phase," which involves a pairwise synchronization along the edges of the composed topology. This phase is also initiated by the root by broadcasting a time-sync message, and proceeds from the nodes of a lower level until the highest level. A node receiving a time sync from the root sends a synchronization-pulse message to the root indicating that it wants to adjust its clock to the root. The root responds back with an acknowledgment containing the required information for the node to synchronize. Nodes at level 2 will also be receiving the synchronization pulse message sent to the root, which will act as a time sync for these nodes. This hierarchical way of synchronizing from root up to the highest level of the topology induces every node in the network to be synchronized with the root node. Therefore, the root node is referred as a user node and is usually equipped with a GPS receiver.

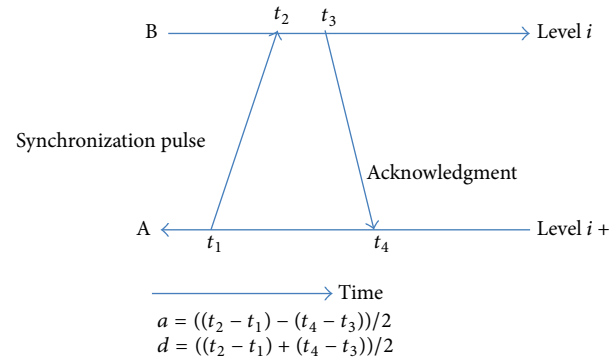


Figure 6: Message exchange between the nodes A and B.

Consider Figure 6, where node A from a higher level $i+1$ wants to synchronize its clock to a node B from a lower level i. At time t_1, A sends a synchronization-pulse message to node B, which contains the level number $i+1$ and time t_1. B receives this message at time t_2 and sends back an acknowledgment message at time t_3 including the values for t_1, t_2, t_3, and the level information i. When A gets the acknowledgment message at time t_4, it can calculate the clock drift a and propagation delay d as follows:

$$a = \frac{(t_2 - t_1) - (t_4 - t_3)}{2}$$
$$d = \frac{(t_2 - t_1) + (t_4 - t_3)}{2}. \quad (8)$$

After finding a and d, A can now adjust its clock to the clock of B. Authors claim that postfacto synchronization can be applied for energy savings in multihop networks. Assume that node A needs to send a message to node E through the path A-B-C-D-E. Then, synchronization is started between two nodes, once the there is a message exchange. For instance, when B gets a message from A, it synchronizes with A first using TSPN and then sends the message to the next hop C.

4.4. Flooding Time Synchronization Protocol (FTSP). Maróti et al. [19] proposes the Flooding Time Synchronization Protocol (FTSP) to achieve network-wide time synchronization. The goal of the scheme is to provide high accuracy and scalability under large number of mobile nodes. For better accuracy, MAC layer time stamping is utilized to eliminate most of the error resources on the critical path from sender to receiver.

Two of the error resources mentioned in Section 3.2—transmission and reception delays—are analyzed in more detail and divided further into four categories: interrupt handling time (the delay between and microcontroller), encoding time (time to encode and transform message to electromagnetic waves), decoding time (time to decode and transform message from electromagnetic waves to binary data), and byte alignment time (delay from different alignment of sender and receiver). The two triangles at the top and bottom show

the time when the message is time-stamped. This time-stamping mechanism eliminates the jitter of interrupt handling time on the sender.

FTSP further removes other sources of error except the propagation delay as follows. A broadcast message is time stamped both at the sender and the receiver. A message consists of four regions: preamble bytes, sync bytes, actual data with a descriptor, and crc bytes. While preamble bytes are transmitted, the receiver radio can synchronize to the carrier frequency of the message signal. Bit offset is calculated from sync bytes at the receiver for bit alignment, and the message is time stamped after sync bites are sent/received. Finally, the clock drift is found from a best line fitted on data points representing time and clock offset by linear regression (a similar approach applied in TSPN).

For multihop synchronization, FTSP uses reference points that hold both the local and global timestamp of an instance of time. All the nodes in the network synchronize to a root node. If a node can communicate with the root directly, then it will collect reference points from the root to synchronize. However, if it is not in the communication range of the root, then it can get the reference points from other synchronized nodes in the network. The root node is selected dynamically according to the changing network topology.

5. Comparison of Methods

5.1. Precision. Precision designates the accuracy of the algorithm and can refer to either absolute (with respect to an external standard clock) or relative (with respect to nodes within a network). Römer [14] conducts their experiments on a prototype system and reports 3ms inaccuracy on a test set of 5 hops. Their accuracy degreases as the age of time stamp and the number of hops increase. This number is the lowest precision among all other three methods in consideration. This is mainly due to two reasons. First reason is that Römer's method takes into account only the current and one previous message exchange between two nodes. Although this reduces the overall system complexity, the accuracy degrades from all the error sources (send, access, transmission, propagation, reception, and receive delays) on the critical path from sender to receiver. Unlike RBS and FTSP, no averaging or best line fitting on the previous offset points is applied. This causes Römer's method also to be sensitive to failures. Second reason is that Römer does not exploit the MAC layer time stamping. As analyzed in [18, 19], time stamping messages at the MAC layer remove considerable amount of error sources on the critical path. Recall that TPSN does not use previously exchanged messages but utilizes the MAC layer implementation.

RBS [2] follows a different approach than all other three methods in the sense that it synchronizes receivers with each other, rather than receiver to sender. This removes the nondeterminism at the sender (send, access, and transmission delays). With a 4-hop network on Berkeley motes, average error is reported as 3.68 μs. This is considered to a high precision. TSPN also conducts its experiments on Berkeley motes and reports an average error of 20 μs. Although this is worse than RBS, the authors of TSPN argue that this is due to the fact that RBS experiments were conducted on superior operating system. Under same circumstances, TSPN is claimed to be achieving two times better precision. Authors argue that this is due to two reasons: MAC layer utilization and two way message exchange between two nodes in TSPN.

Table 1: Synchronization error (in μs).

	Romer	RBS	TSPN	FTSP
Average error	207	29.13	16.9	0.95
Worst case error	274	93	44	4.32
Best case error	0	0	0	0

Table 2: Synchronization error over multihop.

Synchronization ad hoc network (Romer)	3 ms error on 5 hops
RBS	3.68 μs error on 4 hops
TSPN	20 μs error on 4 hops
FTSP	3 μs error on 6 hops

Despite these, if RBS was implemented with MAC layer utilization ability, it would give better results than TSPN under the scenario that the nodes synchronize with each other frequently and at constant time intervals. The reason is that averaging and line fitting for clock offset and drift in RBS will eventually dominate the two-way message exchange in TSPN statistically.

FTSP [19] is implemented on UCB Micra platforms and achieves 3 μs error on 6 hops, which is 0.5 μs error per hop. This is the best precision among all other three methods. There are two contributors for this result. First, FTSP exploits MAC layer time stamping more than TSPN does. It removes the sources of errors on the sender and receiver by a smarter implementation that adjusts the receiver of a message to the carrier frequency. Second, FTSP exploits from linear regression to estimate clock drift and offset, like RBS. But different than RBS, FTSP is a sender-receiver synchronization method that uses multiple time stamps (both global and local) corresponding to a reference point. Synchronization error statistics of each method shown in Tables 1 and 2 show the synchronization error in case of multihop of each method.

5.2. Energy Efficiency. Considering that sensors are small and untethered devices, energy resources should be utilized in an efficient manner. There are three issues to be considered: post-facto synchronization, computation, and communication. Post-facto synchronization proposes the nodes to stay in a low-power state with unsynchronized clocks, until a event of interest occurs. All three algorithms (except FTSP) suggest a post-facto synchronization scheme.

There is no quantitative data; however, Römer's methods behave the best in terms of energy utilization. Römer takes into account that a node can go idle for a long time; hence, the exchanged messages contain this idle time information to be used in clock drift calculation. Also, the computational complexity and message overload are very low. Unlike RBS and FTSP, Römer does not require a complex computation

mechanism based on message exchanges in the past, but only one previous message exchange.

Second best energy efficient method is RBS. RBS also takes into account post-facto synchronization. Clock drift is estimated by linear regression based line fitting on past clock phase offsets. Thus, computational complexity is considerably higher than Römer's and TSPN and requires more message exchanges than Römer's. However, it purveys energy savings by not updating local clocks of the nodes.

Energy utilization in TSPN is moderate. Although being a post-facto synchronization scheme, it has high computation and communication costs. In pairwise synchronization, the method requires three message exchanges between the nodes (time sync, synchronization pulse, and acknowledgment). This results in a higher communication load. Clock drift computation is not as complex as RBS or FTSP; but unlike RBS, local clocks of the nodes are updated.

FTSP is the most inefficient synchronization scheme among the other three, in terms of energy utilization. No argument on post-facto synchronization is made in the paper. Clock drift estimations are done in a similar way with RBS; hence, it has high computation cost and requires to have enough message exchanges to be completed before convergence. Also, the method updates the local clocks. All these result in high energy consumption for FTSP.

To sum up our conclusion for energy, three factors affect the efficiency of an algorithm. Among those three, post-facto synchronization plays the most important role. Then comes the computation complexity, including whether local times are updated or not. Finally, the communication complexity refers to the number of message exchanges required.

5.3. Scalability. Scalability requires a synchronization scheme to be able to scale well with size of the network. In other words, scalability measures how much the efficiency is affected by the increasing number of nodes. By 12 efficiencies, I mean precision and energy utilization and will be comparing the methods in terms of these two constraints.

Römer utilizes the energy in an efficient way. This makes it scalable in terms of energy consumption. However, accuracy of the system degrades by two things: age of time stamps and the number of hops. Even with a network of 5-hops, the average inaccuracy was reported as 3ms, which is a considerable low precision. Therefore, it can be concluded that scalability of Römer's method depends on the application. For a system, in which constraints on energy consumption are stricter than constraints on accuracy, Römer's would be the best scalable method. The method was tested on a prototype system; thus, scalability of the system was not discussed in the paper.

RBS is a scalable system in terms of both precision and energy efficiency. Although TSPN and FTSP achieve better precision than RBS, energy is utilized in a more efficient way by RBS. In their discussion, authors claim that TSPN is a very scalable system, considering that it would give better accuracy than RBS. However, TSPN has moderate scalability. This is because it requires a topological hierarchy and does not assume dynamic changes in the network structure. Also, the experiments reported by the TSPN paper use 300 nodes at the maximum case. This number is not enough, considering that number of sensor in a network can be in order of thousands.

FTSP is a scalable network in terms of accuracy. It would be the best system for an application, in which precision requirements are high and power is not an issue. Similar to TSPN, FTSP also exploits from a hierarchical network structure. However, unlike TSPN, the root node is selected dynamically among the nodes in the node. This makes FTSP more robust to failures in the networks. Experimental set of FTSP consists of 1000 motes, which is the highest number of nodes used among all other three systems, as well as many other synchronization schemes not mentioned here.

6. Conclusions

I have discussed synchronization issues in wireless sensor networks. Dynamic topological structure and limited capabilities of sensor nodes make the synchronization problem more difficult for sensor network environments. Moreover, possible delays in message exchange between the nodes make it even harder. There are certain issues to be considered in designing a synchronization scheme, such as multimodality, post-facto synchronization, single or multihop topology, and adaptability. The desired features of the synchronization scheme include energy efficiency, scalability, precision, lifetime, scope, and cost.

I have summarized four different synchronization schemes for wireless sensor networks. For comparison, I have analyzed three features—efficiency, scalability, and precision—of these systems in detail. It is seen that there is a trade-off between these features and no single system provides all these together.

Systems that provide high precision either remove possible sources of error on sender side by taking a receiver-receiver based approach or remove critical error sources by exploiting the MAC layer time stamping with a sender-receiver based approach. Furthermore, estimating the clock drift with linearregression on past clock offsets makes the system robust to possible failures in the network.

Post-facto synchronization, computation, and communication complexities determine the energy efficiency of a synchronization method. Systems that require synchronization when an event of interest occurs save energy by letting the nodes go idle and save power. Moreover, local clock updates also require high amount of energy. Systems, which do not require this, utilize the energy more efficiently.

Finally, scalability can be in terms of either precision or energy efficiency. It is seen that the more energy a system requires, the more precision that it can achieve and vice versa. So, scalability depends on application requirements, whether more precision or better energy utilization is needed.

References

[1] J. Wu, L. Jiao, and R. Ding, "Average time synchronization in wireless sensor networks by pairwise messages," *Computer Communications*, vol. 35, no. 2, pp. 221–233, 2012.

[2] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS—Operating Systems Review*, vol. 36, pp. 147–163, 2002.

[3] J. Elson and K. Romer, "Wireless sensor networks: a new regime for time synchronization," *ACM SIGCOMM Computer Communication Review*, vol. 33, pp. 149–154, 2003.

[4] M. K. Maggs, S. G. O'Keefe, and D. V. Thiel, "Consensus clock synchronization for wireless sensor networks," *IEEE Sensors Journal*, vol. 12, no. 6, pp. 2269–2277, 2012.

[5] J. Chen, Q. Yu, Y. Zhang, H.-H. Chen, and Y. Sun, "Feedback-based clock synchronization in wireless sensor networks: a control theoretic approach," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2963–2973, 2010.

[6] S. Misra and A. Vaish, "Reputation-based role assignment for role-based access control in wireless sensor networks," *Computer Communications*, vol. 34, no. 3, pp. 281–294, 2011.

[7] B. Liu, F. Ren, J. Shen, and H. Chen, "Advanced self-correcting time synchronization in wireless sensor networks," *IEEE Communications Letters*, vol. 14, no. 4, pp. 309–311, 2010.

[8] F. Cristian, "Probabilistic clock synchronization," *Distributed Computing*, vol. 3, no. 3, pp. 146–158, 1989.

[9] R. Gusella and S. Zatti, "The accuracy of the clock synchronization achieved by tempo in berkeley unix 4.3bsd," Tech. Rep. UCB/CSD-87-337, EECS Department, University of California, Berkeley, Calif, USA, 1987.

[10] C. Fidge, "Logical time in distributed computing systems," *Computer*, vol. 24, no. 8, pp. 28–33, 1991.

[11] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, 1978.

[12] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.

[13] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, 2004.

[14] K. Römer, "Time synchronization in ad hoc networks," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '01)*, pp. 173–182, October 2001.

[15] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.

[16] 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.

[17] J. Elson and D. Estrin, "Time synchronization for wireless sensor networks," in *Proceedings of the 15th International Parallel and Distributed Processing Symposium*, p. 186, San Francisco, Calif, USA, April 2001.

[18] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 138–149, November 2003.

[19] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 39–49, November 2004.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 941473, 15 pages
http://dx.doi.org/10.1155/2013/941473

Research Article

Optimal Joint Expected Delay Forwarding in Delay Tolerant Networks

Jia Xu,[1,2,3] Xin Feng,[1,2,3] Wen Jun Yang,[1,2,3] Ru Chuan Wang,[1,2,3] and Bing Qing Han[4]

[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, Jiangsu 210003, China*
[3] *Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education Jiangsu Province, Nanjing, Jiangsu 210003, China*
[4] *Department of Information Science, Nanjing Audit University, Nanjing 210029, China*

Correspondence should be addressed to Jia Xu; xujia@njupt.edu.cn

Received 5 June 2013; Accepted 25 August 2013

Academic Editor: Shukui Zhang

Multicopy forwarding schemes have been employed in delay tolerant network (DTN) to improve the delivery delay and delivery rate. Much effort has been focused on reducing the routing cost while retaining high performance. This paper aims to provide an optimal joint expected delay forwarding (OJEDF) protocol which minimizes the expected delay while satisfying a certain constant on the number of forwardings per message. We propose a comprehensive forwarding metric called *joint expected delay* (JED) which is a function of remaining hop-count (or ticket) and residual lifetime. We use backward induction to calculate JED by modeling forwarding as an optimal stopping rule problem. We also present an extension to allow OJEDF to run in delay constrained scenarios. We implement OJEDF as well as several other protocols and perform trace-driven simulations. Simulation results confirm that OJEDF shows superiority in delay and cost with acceptable decrease of delivery rate.

1. Introduction

A delay tolerant network (DTN) [1] is a sparse mobile multihop network where a complete path from the source to the destination may not exit most of the time. A DTN can be viewed as a collection of time-varying and separate node clusters. Under this condition, conventional mobile ad hoc network (MANET) or wireless sensor network (WSN) routing will become impractical because of frequent disruption and high packet loss. The Delay-Tolerant Networking Research Group (DTNRG) [2] is concerned with architecture and relevant protocols in order to provide interoperable communications with and among extreme and performance-challenged environments. In DTNs, the messages can be forwarded asynchronously using the store-carry-forward routing mechanism which relies on the contact opportunities between nodes [3].

Routing in DTN has been an active research area in recent years. Epidemic [4] exchanges the copies whenever contact occurred within pair of nodes. Epidemic can guarantee the maximized delivery rate and the minimized end-to-end delay under ideal conditions. However, unrestrained flooding is impractical because of huge consumption of energy, buffer and bandwidth. Much literature has been focused on reducing the number of copies of each message while retaining a high routing performance. Previous works proposed a variety of metrics, including movement trend [5], delivery probability [6], encounter frequency [7], the distance to destination [8], end-to-end delay [9], energy [10], signal to noise ratio (SNR) [11], and comprehensive utility [12]. These metrics are used to make decision for forwarding in order to optimize certain performance such as expected delay and delivery rate.

In this paper, we propose optimal joint expected delay forwarding whose main idea is similar to that of optimal probabilistic forwarding (OPF) [13]. This means that optimal stopping rule [14] is used again. Different from maximizing the delivery rate in OPF, we concentrate on optimizing the expected delay of each message with the constraint on the maximum of copies per message. The problem is how to

forward the copies of each message in each discrete time slot in order to minimize the expected delay when the number of copies is given. To solve this problem, we firstly employ the comprehensive dynamic forwarding metric called *joint expected delay* (JED) which is a function of two important states of a message copy: remaining hop-count and residual lifetime. Then, we propose the forwarding rules to decide whether to forward the copies. Based on the hop-count constraint forwarding scheme, we propose a general ticket constraint OJEDF which can be viewed as a development of Spray and Wait [15] with special spray metric. We also extend OJEDF in delay constrained scenarios.

We performed simulations using St. Andrews encounter trace [16] and Illinois encounter trace [17] to evaluate the routing performance of OJEDF against several DTN routing protocols, including Spray and Wait, ProphetV2 [7], and Epidemic, in terms of average delay, delivery rate, and the number of forwardings for each message. Simulation results confirmed that OJEDF improves the average delay and cost with acceptable decrease of delivery rate.

This paper is organized as follows. Section 2 summarizes the related works. Section 3 introduces preliminaries on optimal joint expected delay forwarding and presents the motivation and assumption of OJEDF. Section 4 proposes the calculation algorithms of our delivery metric JED and optimal forwarding rule in hop-count constraint OJEDF and ticket constraint OJEDF, respectively. Section 5 proposes an extension of OJEDF which makes it work in delay constrained scenarios. Section 6 presents our simulation methods and results. Finally, the paper is concluded in Section 7.

2. Related Works

The single-copy routing protocols such as MobiSpace [18] and CAR [19] are energy efficient but always have large delay. Due to uncertainty in nodal mobility and network partition, DTN routing algorithms usually spawn and keep multiple copies for the same message in different nodes. Epidemic is the first distribution routing protocol for DTN, and it gains good delivery delay. However, Epidemic wastes a lot of energy and suffers from severe contention.

Much literature has aimed to balance the delay and energy. Some restricted flooding or spray mechanisms disseminate a small number of message copies to potential relay nodes, and then each copy finds routing to the destination independently. These mechanisms can obtain better delay performance without a lot of resource consumption.

Spray and Wait is a multicopy routing without any routing knowledge. Spray and Wait combines the speed of Epidemic routing with the simplicity and thriftiness of direct transmission. This mechanism "sprays" a number of copies into the network and then "waits" till one of these nodes meets the destination. Spray and Focus [20] replaced the direct transmission in wait phase with utility-based single-copy strategy and proposed the utility transfer mechanism to disseminate the history contact information. Thrasyvoulos was absorbed in utility-based spraying [21] and proposed three potential utility functions: last-seen-first spraying, most-mobile-first spraying, and most-social-first spraying. Jindal proposed distance utility-based spray strategy [22] which utilized dynamic programming to calculate the optimal relay node. An adaptive distributed spray mechanism (AMR) was performed in [23] which determined the depth of spray tree by relay nodes. But AMR is just efficient in random waypoint model and sometimes can produce at most $M/2$ copy redundancy (M is the number of nodes). In addition, theoretical analyses of expected delay in single-copy case and multicopy case were proposed in [24, 25], respectively.

Different from Spray and Wait, delegation forwarding [26] forwards the messages based on different delivery probability metrics. Delegation forwarding maintains a forwarding threshold τ when contact of node pairs occurred. Forwarding threshold indicates the quality of node such as cost, delivery rate, and average delay. RAPID [27] treats DTN routing as a resource allocation problem that translates the routing metric into per-packet utilities such as minimizing average delay, minimizing missed deadlines, and minimizing maximum delay.

Some routing protocols aimed to achieve the desired performance with different levels of prior knowledge about the network. OPF [28] was proposed to maximize the delivery rate of each message when mean inter-meeting times between all pairs of nodes were known. Cyclic MobiSpace [29] assumes that each node has full contact information to other nodes, and it optimizes minimum expected delay.

3. Preliminaries

3.1. Hop-Count Constrained Forwarding and Ticket Constrained Forwarding. We consider two constraint patterns related to the number of message copies: hop-count constrained forwarding and ticket constrained forwarding.

In hop-count pattern, a maximal remaining hop-count K was defined in source node. The remaining hop-count decreased progressively when forwarding occurred. The calculation of the remaining hop-count for the copies of each message is independent. Figure 1 depicts the forwarding process with $K = 3$ in node A. Ticket pattern uses the total number of copies L which can be redistributed in each forwarding. Figure 2 depicts the ticket constrained forwarding process with $L = 8$ in node A using binary spray [9].

3.2. Motivation. Many routing protocols in DTNs forward the messages based on the specific routing metrics by means of comparing the direct forwarding qualities of node pairs. However, such forwarding strategy ignores the time dimensionality which is an important character in DTNs. In fact, the forwarding decision only relied on the direct forwarding qualities of node pairs and cannot stand for the optimal choice in whole time dimensionality of delivery.

In this paper, we optimize the expected delay of each message based on the comprehensive forwarding metric $\mathrm{JED}_{i,d,K,\mathrm{Tr}}$ for each copy in i for destination d with remaining hop-count K (or L tickets) and residual time-to-live Tr.

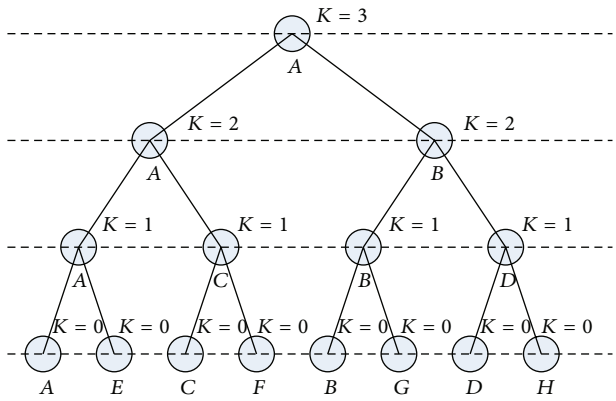


Figure 1: An illustration of hop-count constrained forwarding with $K = 3$.

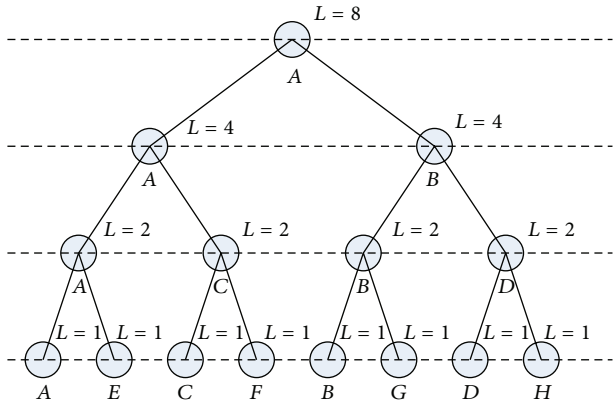


Figure 2: An illustration of ticket constrained forwarding with $L = 8$.

We use backward induction to calculate JEDs by modeling forwarding as an optimal stopping rule problem. Our contributions are as follows.

(1) We propose a comprehensive forwarding metric JED which is related to the time dimensionality and the number of copies for each message.

(2) We propose the backward induction algorithms to calculate JEDs with hop-count constraint and ticket constraint, respectively.

(3) We solve the delay restraint optimal cost forwarding by the extension of OJEDF.

3.3. Assumption. To model our optimal forwarding problem as an optimal stopping rule problem, we have the following assumptions.

(1) Like other probabilistic forwarding protocols, nodal mobility exhibits long-term regularities and each node knows the matrix $M_{n\times n}$ which contains the mean inter-meeting times $\mathrm{MIT}_{i,j}$ between all pairs of nodes $\{i, j\}$. This matrix can be estimated from contact history.

(2) Time dimensionality in whole forwarding process is discrete. OJEDF employs a discrete residual time-to-live Tr with time slot size U for each message copy. So we can state the next time-slot as $\mathrm{Tr}-1$.

(3) JEDs are exponentially distributed. Therefore, the JED of two copies with JED_1 and JED_2 can be calculated by

$$\frac{1}{1/\mathrm{JED}_1 + 1/\mathrm{JED}_2}. \tag{1}$$

Proof. Let $\mathrm{JED}_1 = 1/a$ and $\mathrm{JED}_2 = 1/b$, and let the joint probability density function be $f(x) = ae^{-ax} \cdot be^{-bx}$. Then, the expected value is

$$\begin{aligned} E(X) &= \int_{-\infty}^{+\infty} xf(x)\,dx = \int_0^{+\infty} xabe^{-(a+b)x}dx \\ &= -\frac{ab}{a+b}\int_0^{+\infty} xde^{-(a+b)x} \\ &= -\frac{ab}{a+b}xe^{-(a+b)x}\Big|_0^{+\infty} + \int_0^{+\infty} e^{-(a+b)x}dx \\ &= -\frac{1}{a+b}e^{-(a+b)x}\Big|_0^{+\infty} = \frac{1}{a+b} = \frac{1}{1/\mathrm{JED}_1 + 1/\mathrm{JED}_2}. \end{aligned} \tag{2}$$

Similarly, the meeting probability of two nodes in any time slot with size U can be estimated by

$$P_{i,j} = 1 - \exp\left(-\frac{U}{\mathrm{MIT}_{i,j}}\right). \tag{3}$$

□

4. Optimal Joint Expected Delay Forwarding (OJEDF)

4.1. OJEDF with Hop-Count Constraint. Without loss of generality, we consider a copy in node i with destination d, remaining hop-count K ($K \geq 1$), and residual time-to-live Tr. Upon meeting with j, i can either forward the copy to j or not. Since we assume the forwarding process is discrete, the JED in the next time slot can be stated as $\mathrm{JED}_{i,d,K,\mathrm{Tr}-1}$ in no forwarding case. As an alternative, the copy can be forwarded and is logically regarded as being replaced by two new copies, both of which have $K - 1$ remaining hop-count. If the copy is forwarded, the JED of the copy in node j will be $\mathrm{JED}_{j,d,K-1,\mathrm{Tr}-1}$ and that of the copy in node i will become $\mathrm{JED}_{i,d,K-1,\mathrm{Tr}-1}$. The joint JED can be calculated using formula (1). Consequentially, the forwarding option depends on the comparison of $\mathrm{JED}_{i,d,K,\mathrm{Tr}-1}$ and the joint JEDs of $\mathrm{JED}_{i,d,K-1,\mathrm{Tr}-1}$ and $\mathrm{JED}_{j,d,K-1,\mathrm{Tr}-1}$. The copy is forwarded only if

$$\mathrm{JED}_{i,d,K,\mathrm{Tr}-1} > \frac{1}{1/\mathrm{JED}_{i,d,K-1,\mathrm{Tr}-1} + 1/\mathrm{JED}_{j,d,K-1,\mathrm{Tr}-1}}. \tag{4}$$

We only consider unicast forwarding. When a node contacts with several nodes at the same time slot,

(1) $\text{JED}_{i,d,K,\text{Tr}} := 0$
(2) $P'_{i,N} := 1$
(3) for each relay node j ($j \neq i \cap j \neq d$){
(4) $\quad D_{i,j} = \text{JED}_{i,d,K-1,\text{Tr}-1} \times \text{JED}_{j,d,K-1,\text{Tr}-1}/(\text{JED}_{i,d,K-1,\text{Tr}-1} + \text{JED}_{j,d,K-1,\text{Tr}-1})$}
(5) Append all $D_{i,j}$ in a priority queue Q in ascending order
(6) While (Q is not empty){
(7) $\quad$ if ($j :=$ ServeQueue(Q) and $D_{i,j} < \text{JED}_{i,d,K-1,\text{Tr}-1}$){
(8) $\quad\quad \text{JED}_{i,d,K,\text{Tr}} := \text{JED}_{i,d,K,\text{Tr}} + P'_{i,N} \times P_{i,j} \times D_{i,j}$
(9) $\quad\quad P'_{i,N} := P'_{i,N} - P'_{i,N} \times P_{i,j}$}}
(10) $\text{JED}_{i,d,K,\text{Tr}} := \text{JED}_{i,d,K,\text{Tr}} + P'_{i,N} \times \text{JED}_{i,d,K,\text{Tr}-1}$

Algorithm 1: The calculation of $\text{JED}_{i,d,K,\text{Tr}}$ in hop-count based forwarding.

Table 1: The mean intermeeting times matrix.

	0	1	2	3	4	5	6	7
0	0	20	33	42	20	50	60	35
1	20	0	45	50	30	40	25	10
2	33	45	0	30	40	55	60	20
3	42	50	30	0	60	45	37	27
4	20	30	40	60	0	35	29	100
5	50	40	55	45	35	0	23	80
6	60	25	60	37	29	23	0	26
7	35	10	20	27	100	80	26	0

the copy is forwarded to the node j which has the minimum $\text{JED}_{j,d,K-1,\text{Tr}-1}$.

The forwarding metric $\text{JED}_{i,d,K,\text{Tr}}$ equals the sum of the weighted expected delay in forwarding case and not forwarding case. In forwarding case, node i has the opportunity to meet the relay node $j, K, \ldots$ with probability $P_{i,j}$, $P_{i,K}, \ldots$, respectively. So the joint expected delay that node i meets one of the nodes $j, K, \ldots$ in time-slot Tr and then forwards the copy to it is

$$P_{i,j} \times \frac{1}{1/\text{JED}_{i,d,K-1,\text{Tr}-1} + 1/\text{JED}_{j,d,K-1,\text{Tr}-1}} + \left(1 - P_{i,j}\right) \times P_{i,K} \times \frac{1}{1/\text{JED}_{i,d,K-1,\text{Tr}-1} + 1/\text{JED}_{k,d,K-1,\text{Tr}-1}} + \cdots. \tag{5}$$

If node i did not meet any relay node, the copy was kept in node i with joint expected delay $\text{JED}_{i,d,K,\text{Tr}-1}$ and probability $P'_{i,N} = 1 - P_{i,j} - (1 - P_{i,j}) \times P_{i,K} - \cdots$. Algorithm 1 shows the calculation of $\text{JED}_{i,d,K,\text{Tr}}$ in hop-count based forwarding using the backward induction.

Specially, the copy only can be delivered to the destination when the remaining hop-count $K = 0$, and we set $\text{JED}_{i,d,0,\text{Tr}} = \text{MIT}_{i,d}$. For each pairs of nodes $\{i, d \mid i = d\}$, we let $\text{JED}_{i,d,K,\text{Tr}} = \delta$ (δ is a very small number such as $10e - 6$) in order to make the formula (1) calculable.

In our OJEDF, Tr is only used to define the finite upper bound on the number of stages which is essential for modeling OJEDF as optimal stopping rule problem. So it is feasible to calculate the JEDs of the previous stage using the backward induction method even when $\text{JED}_{i,d,K,\text{Tr}} > \text{Tr}$. This implies that the OJEDF rule depicted in formula (4) is valid when $\text{JED}_{i,d,K,\text{Tr}} > \text{Tr}$. However, this kind of JEDs cannot be employed in delay constrained scenarios which we will introduce in Section 5 since it is impossible to deliver the message within residual time-to-live.

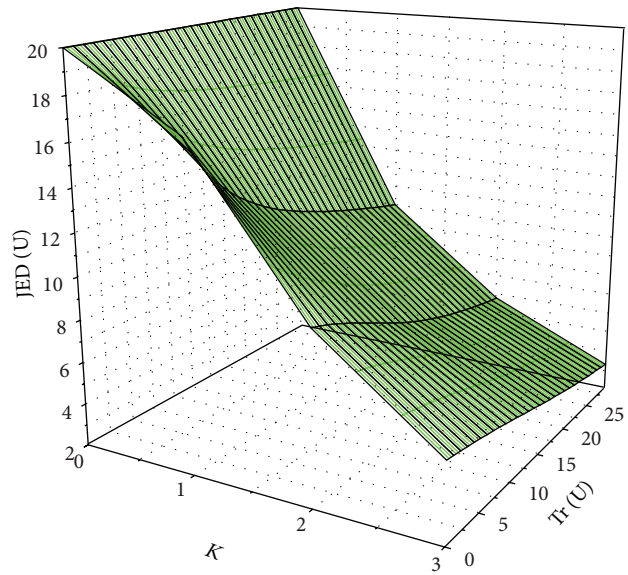


Figure 3: The JEDs calculated by OJEDF in hop-count based forwarding.

As an example, a matrix $M_{8\times8}$ including discrete mean inter-meeting times (normalized by time-slot size U) between each pair of nodes was defined in Table 1. The JEDs calculated by OJEDF using Algorithm 1 were shown in Figure 3 with $K = 3$ (enough to forward each message to all nodes), maximum Tr = 30, $i = 0$, $d = 1$, and $U = 1$. In Figure 3, the JEDs are decreased with increasing K and Tr. It means that a message has smaller expected delay when more remaining hop-count and residual time can be used to accomplish delivery. As a result, we can make forwarding options based on formula (4) when node i meets potential relay nodes with different remaining hop-count K and residual time-to-live Tr. Figures 4(a), 4(b), and 4(c) show the forwarding options based on OJEDF with $K = 1$, $K = 2$, and $K = 3$. We can see

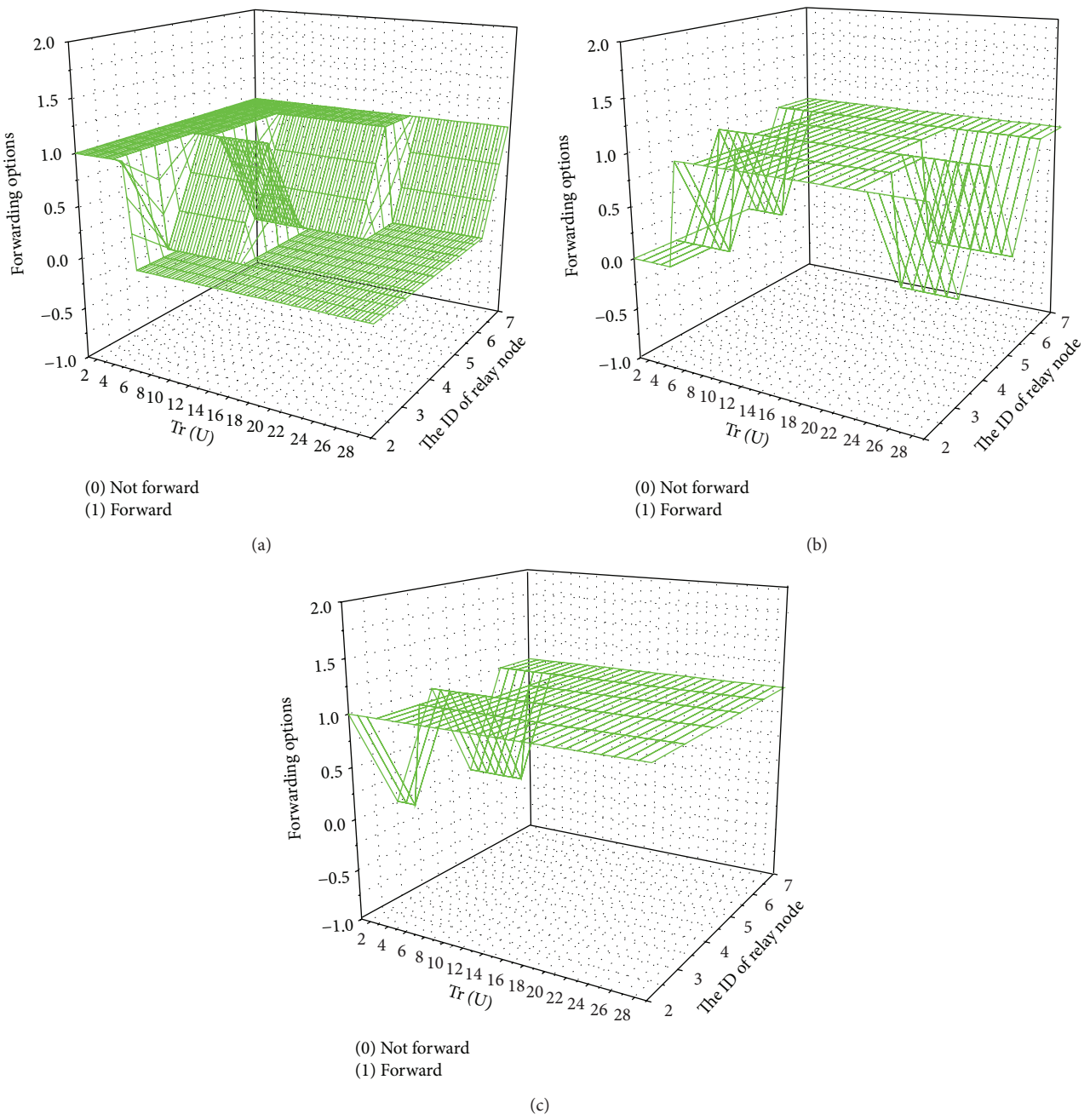


Figure 4: (a) Forwarding options based on OJEDF in hop-count based forwarding with $K = 1$. (b) Forwarding options based on OJEDF in hop-count based forwarding with $K = 2$. (c) Forwarding options based on OJEDF in hop-count based forwarding with $K = 3$.

that the message forwarding tends to be more cautious when the remaining hop-count declines gradually along with the forwarding process.

4.2. OJEDF with Ticket Constraint. Another copy constraint pattern is ticket constraint which is more precise than hop-count constraint since it replaces the remaining hop-count K with the number of tickets L. Without loss of generality, we consider a copy in node i with destination d, the number of tickets L ($L > 1$), and residual time-to-live Tr. When node i meets any relay node j, the forwarding option depends on $\mathrm{JED}_{i,d,L,\mathrm{Tr}-1}$ and the joint JED of $\mathrm{JED}_{i,d,L1,\mathrm{Tr}-1}$ and $\mathrm{JED}_{j,d,L2,\mathrm{Tr}-1}(L1 + L2 = L)$. We define $D_{i,j\_\mathrm{MIN}} = \min(1/(1/\mathrm{JED}_{i,d,L1,\mathrm{Tr}-1} + 1/\mathrm{JED}_{j,d,L2,\mathrm{Tr}-1}))$, and the copy is forwarded only if $\mathrm{JED}_{i,d,L,\mathrm{Tr}-1} > D_{i,j\_\mathrm{MIN}}$.

Algorithm 2 shows the backward induction algorithm to calculate $\mathrm{JED}_{i,d,L,\mathrm{Tr}}$ for ticket based forwarding. Specially, the copy only can be delivered to the destination when the number of tickets $L = 1$, and we set $\mathrm{JED}_{i,d,1,\mathrm{Tr}} = \mathrm{MIT}_{i,d}$.

We use the same matrix defined in Table 1. The JEDs calculated by OJEDF in ticket based forwarding were shown in Figure 5 with $L = 4$, maximum Tr = 30, $i = 0$, and

(1) $\text{JED}_{i,d,L,\text{Tr}} := 0$
(2) $P'_{i,N} := 1$
(3) for each relay node j ($j \neq i \cap j \neq d$){
(4) $D_{i,j} = \text{JED}_{i,d,L1,\text{Tr}-1} \times \text{JED}_{j,d,L2,\text{Tr}-1}/(\text{JED}_{i,d,L1,\text{Tr}} + \text{JED}_{j,d,L2,\text{Tr}-1})$}
(5) Append all $D_{i,j}$ in a priority queue Q in ascending order
(6) while (Q is not empty){
(7) if (j := ServeQueue(Q) and $D_{i,j} < \text{JED}_{i,d,L,\text{Tr}-1}$){
(8) $\text{JED}_{i,d,L,\text{Tr}} := \text{JED}_{i,d,L,\text{Tr}} + P'_{i,N} \times P_{i,j} \times D_{i,j}$
(9) $P'_{i,N} := P'_{i,N} - P'_{i,N} \times P_{i,j}$}}
(10) $\text{JED}_{i,d,L,\text{Tr}} := \text{JED}_{i,d,L,\text{Tr}} + P'_{i,N} \times \text{JED}_{i,d,L,\text{Tr}-1}$

Algorithm 2: The calculation of $\text{JED}_{i,d,L,\text{Tr}}$ in ticket based forwarding.

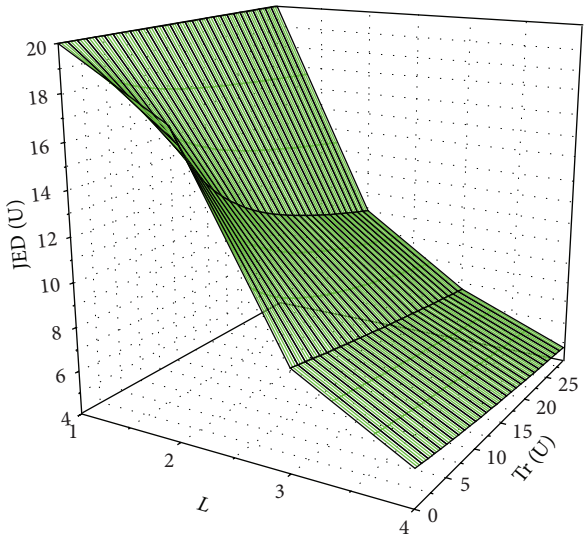


Figure 5: The JEDs calculated by OJEDF in ticket based forwarding.

$d = 1$. As a result, we can make forwarding options based on formula (4) when node i meets potential relay nodes with the different number of ticket L and residual time-to-live Tr. Figures 6(a), 6(b), and 6(c) show the forwarding options based on OJEDF with $L = 1$, $L = 2$, and $L = 3$.

5. An Extension of OJEDF for Delay Constrained Scenarios

We have solved the optimal joint expected delay problem with the constraint of remaining hop-count or ticket of messages. As a further consideration, we extend OJEDF to run in delay constrained DTN applications. For example, electronic notice in campus networks [30] and village networks [31] and short-term weather information in large national parks [32] must be forwarded with constrained delay. On the other hand, as the access networks, configuration and forecast for QoS are also necessary in order to provide acceptable service (such as e-mail) in DTNs.

Since JED is a function of remaining hop-count (or the number of tickets) and residual lifetime, the OJEDF can also be used to estimate the minimum hop-count K or the minimum tickets L to meet the constrained delay. We give a simple hop-count estimation algorithm with constrained delay based on JED in Algorithm 3. We define the K_MAX as the maximum of hop-count which equals $\lceil \log_2 M \rceil$ (M is the number of nodes) which is large enough to spray copies to all of the nodes in hop-count based forwarding. The constrained delay (CD) and initial time-to-live TTL (TTL_{init}) must be defined in advance. Both of them are normalized by time-slot U and satisfy $\text{TTL}_{\text{init}} \geq \text{CD}$. Similarly, Algorithm 4 shows the ticket estimation algorithm in which the maximum of tickets L_MAX equals M.

We can get the minimum of hop-count with different constrained delay for each destination node using Algorithm 3. Assuming that the message was generated in node 0 and the mean inter-meeting times matrix was in Table 1, Figure 7 showed the minimum of hop-count with TTL = 29. The ticket estimation was shown in Figure 8 with the same parameters.

6. Simulation

We evaluated OJEDF with hop-count constraint (OJEDF-H) and OJEDF with ticket constraint (OJEDF-T) against Spray and Wait (SNW), ProphetV2 (PRO2), and Epidemic (EP) using St. Andrews encounter trace and Illinois encounter trace. We implemented the above protocols in The ONE [33] which was developed by the Helsinki University of Technology. We obtained the performance indicators including average delay (only of the copies delivered successfully), the delivery rate, and the average number of copies for each message (no acknowledgment mechanism) from The ONE. The simulation parameters were shown in Table 2 (only the default values without any special instruction).

6.1. Simulation Using St. Andrews Encounter Trace. St. Andrews encounter trace set up a mobile sensor network comprising mobile IEEE 802.15.4 sensors (T-mote invent devices) carried by human users and Linux-based basestations that bridge the 802.15.4 sensors to the wired network. St. Andrews encounter trace deployed 27 T-mote invent devices among 22 undergraduate students, 3 postgraduate students, and 2 members of the staff of the University of St. Andrews

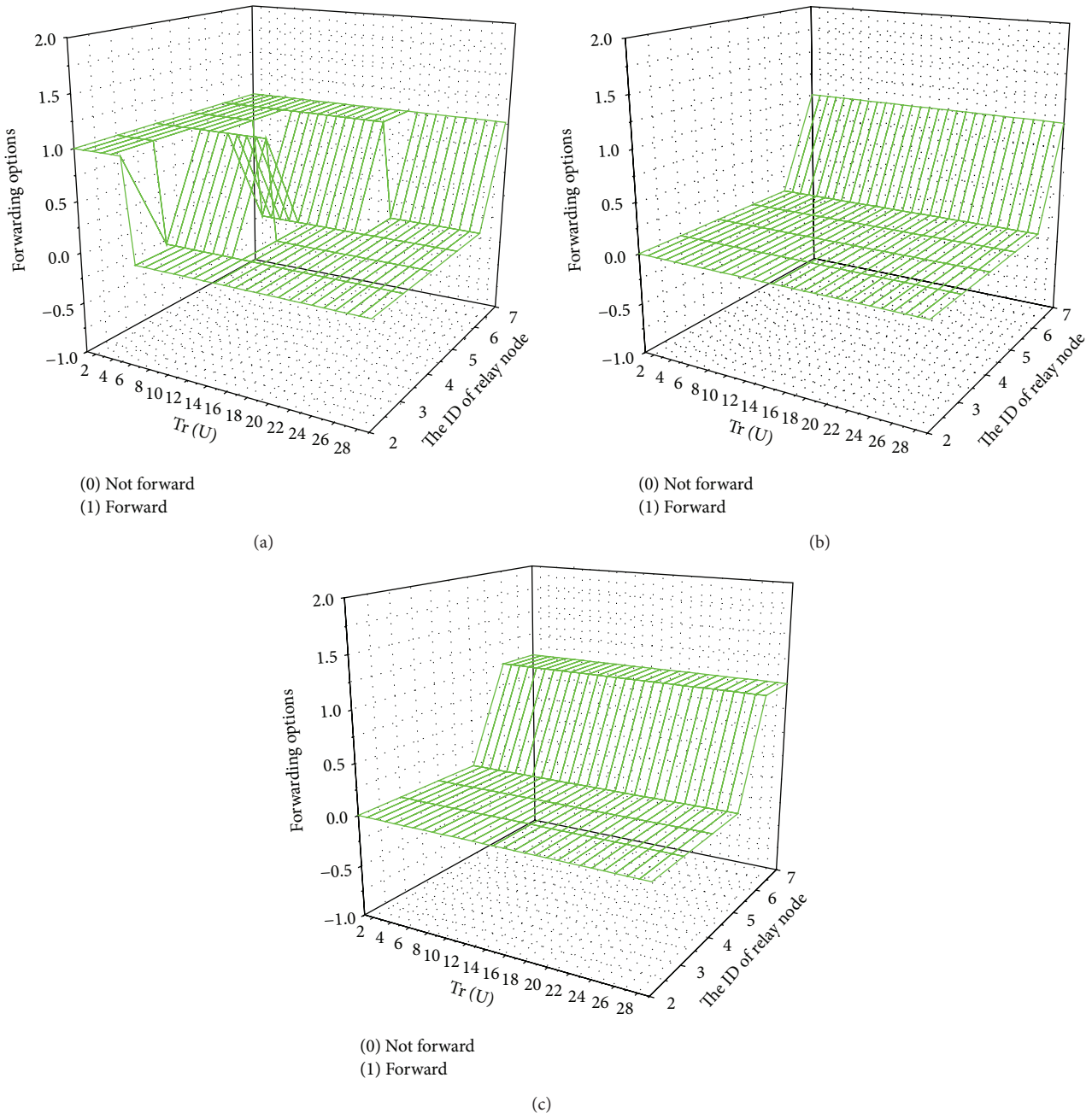


Figure 6: (a) Forwarding options based on OJEDF in ticket based forwarding with $L = 2$. (b) Forwarding options based on OJEDF in ticket based forwarding with $L = 3$. (c) Forwarding options based on OJEDF in ticket based forwarding with $L = 4$.

from the participants' Facebook friend lists. Participants were asked to carry the devices whenever possible.

Due to the limit of buffer and the queue of external events, we only employ the encounter data from time 0 to 5369. We optimize the original data in order to remove the reduplicative records and invalid users. The reduplicative records are the same encounter records in the same time, and the invalid users mean the users who never appeared before time 5369. As a result, we get 25 valid users and 591 external encounter events. The final encounter data must be formatted into standard external event queue data which can be read by The ONE.

We get the performance with different message overload. The intervals of message generator varied from 10 s to 50 s.

Figure 9(a) showed the average delay with different intervals of message generator. The average delay increased with the increasing intervals. When the message overload was high, the multihop forwardings were difficult due to the constrained buffer size. As a result, the average delay was low. The average delay increased with the decreased message

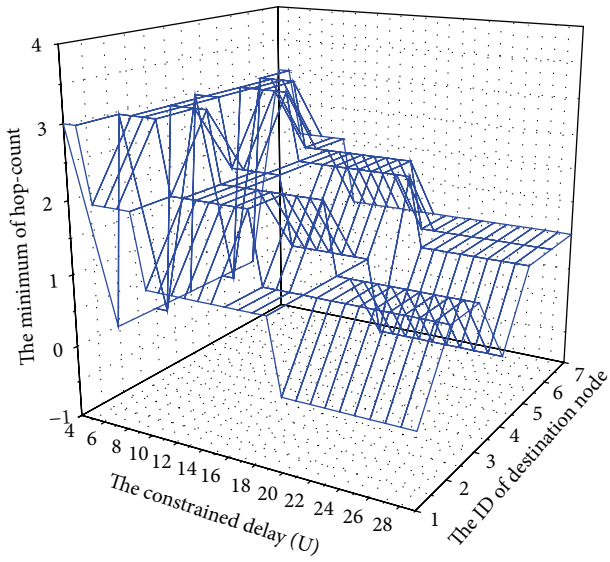


Figure 7: The minimum of hop-count with TTL = 29.

(1) $K\_MAX := \lceil \log_2 M \rceil$
(2) CD := Constrained Delay
(3) for ($K = 0$; $K \leq K\_MAX$; $K + +$){
(4) for (TTL = CD; TTL $\leq$ TTL_{init}; TTL + +){
(5) Calculate $JED_{i,d,K,TTL}$, using Algorithm 1; }
(6) if ($JED_{i,d,K,TTL} \leq CD$)
(7) return K; }
(8) return false;

Algorithm 3: Hop-count estimation algorithm.

(1) $L\_MAX := M$
(2) CD := Constrained Delay
(3) for ($L = 1$; $L \leq L\_MAX$; $L + +$) {
(4) for (TTL = CD; TTL $\leq$ TTL_{init}; TTL + +){
(5) Calculate $JED_{i,d,L,TTL}$ using Algorithm 2; }
(6) if ($JED_{i,d,L,TTL} \leq CD$)
(7) return L; }
(8) return false;

Algorithm 4: Ticket estimation algorithm.

overload since the multihop forwardings became common. There was no copy control in EP and PRO2, so the average delay of them was high compared with that of SNW and OJEDF due to the constrained buffer size set in simulation. We can see from Figure 9(a) that the average delay of PRO2 was lower than that of SNW and OJEDF when the interval of message generator was long enough. This is because the buffer size becomes sufficient in low message overload, and the average delay is low when there is no copy control in PRO2. OJEDF which optimizes the joint expected delay can be viewed as an improvement of SNW. The average delay rates of OJEDF-H and OJEDF-T were 8.89% and 14.05% lower than that of SNW, respectively.

Figure 9(b) showed the delivery rate with different intervals of message generator. The delivery rate increased when the overload decreased. All of the delivery rates are lower than 50%. This is because only 244 contacts occurred between 25 users in the first 5369 s. This is a very low contact probability. OJEDF did not forward the copies to the node first encountered but forwarded based on the expected delay. So the delivery rates of OJEDF-H and OJEDF-T were 14.20% and 13.61% lower than that of SNW in average, respectively. Most of the time, the delivery rate of OJEDF is higher than that of EP and PRO2.

We also got the number of forwardings with different intervals of message generator. We can see form Figure 9(c) that the numbers of forwardings of OJEDF-H and OJEDF-T were much lower than those of EP, PRO2, and SNW. OJEDF only forwarded the messages to the optimal node, and the numbers of forwardings of OJEDF-H and OJEDF-T were 34.86% and 36.00% lower than that of SNW, respectively.

We also compared the performance with different TTLs (from 10 minutes to 90 minutes).

Figure 10(a) showed the average delay. The average delay means the delivery delay of messages which were delivered successfully, so the average delay was low when TTL was low. The average delay rates of EP and PRO2 were higher than those of SNW and OJEDF. OJEDF outperformed other protocols in terms of the average delay with all TTLs. Specially, the average delay rates of OJEDF-H and OJEDF-T were 9.24% and 18% lower than that of SNW, respectively.

Figure 10(b) showed the delivery rate with different TTLs. With increasing TTL, more messages were unable to be forwarded due to the restraint of buffer size. As a result, the delivery rate of EP and PRO2 trended to decrease. On the contrary,

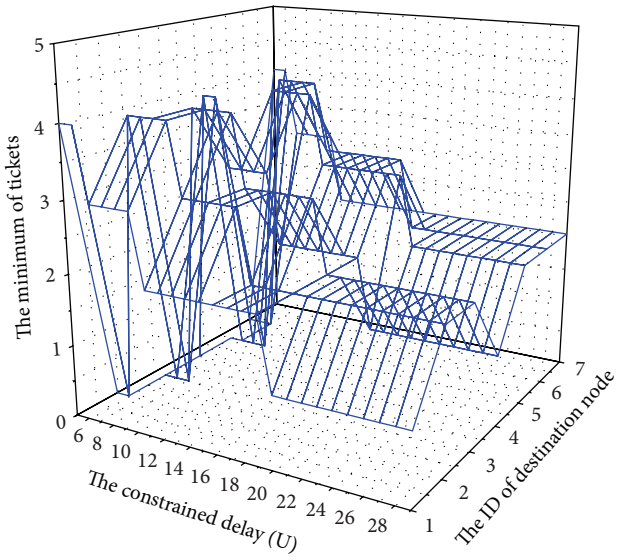


Figure 8: The minimum of tickets with TTL = 29.

Table 2: Settings for simulation.

Parameters	Values for St. Andrews trace	Values for Illinois trace	Remarks
Time slot for OJEDF-H	70 s	176 s	Only for OJEDF-H
Time slot for OJEDF-T	140 s	245 s	Only for OJEDF-T
Mean self-meeting time δ	0.000001 s	0.000001 s	Only for OJEDF
[-1pt] Remaining hop-count initialization constant	3	2	Only for OJEDF-H
Number of tickets for OJEDF-T	6	4	Only for OJEDF-T
Time unit for PRO2	30 s	30 s	Only for PRO2
Delivery predictability initialization constant	0.5	0.5	Only for PRO2
Delivery predictability transitivity scaling constant β	0.9	0.9	Only for PRO2
Delivery predictability aging constant γ	0.999885791	0.999885791	Only for PRO2
Number of tickets for SNW	6	6	Only for SNW
Binary mode	True	True	Only for SNW
Number of nodes	25	7	
Time-to-live	90 min	500 min	
Number of external encounter events	591	7017	
Default connections	False	False	
Message size	500 kB	50 kB	
Interval of message generator	25 s–35 s	250 s–350 s	
Buffer size	5 M	1 M	
Simulation times	20	20	
End time of simulation	5369 s	240000 s	

the delivery rate of SNW and OJEDF increased with increasing TTL. The delivery rates of OJEDF-H and OJEDF-T were 4.9% and 3.8% lower than that of SNW, respectively.

Figure 10(c) showed the number of forwardings with different TTLs. OJEDF outperformed all protocols in terms of the number of forwardings with all TTLs. Specially, the numbers of forwardings of OJEDF-H and OJEDF-T were 34.69% and 37.66% lower than that of SNW, respectively.

6.2. Simulation Using Illinois Encounter Trace. Illinois encounter trace is the dataset of MACs of Bluetooth and Wi-Fi access points collected by the University of Illinois Movement

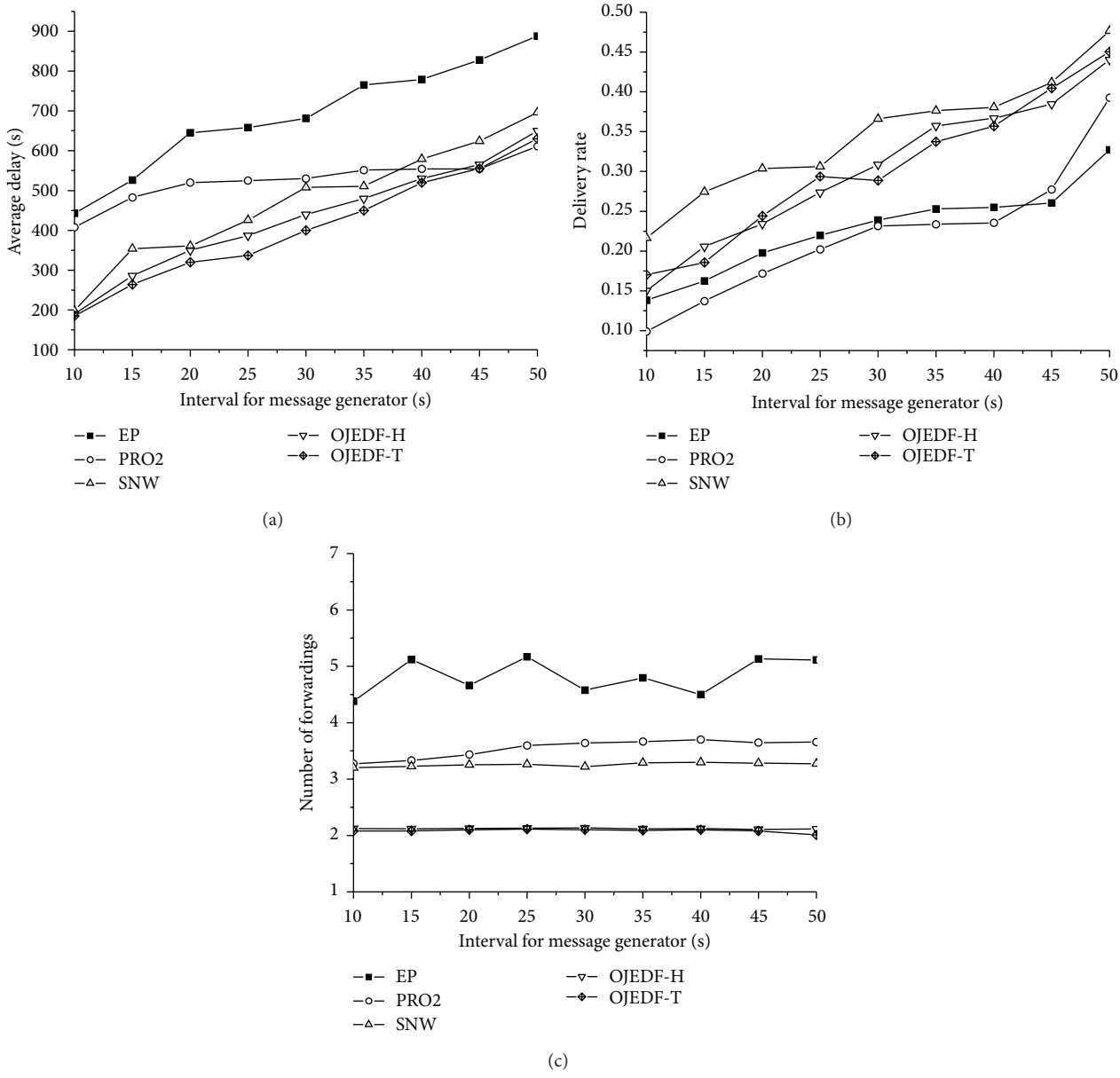


Figure 9: (a) Average delay with different interval for message generator using St. Andrews encounter trace. (b) Delivery rate with different interval for message generator using St. Andrews encounter trace. (c) Number of forwardings with different interval for message generator using St. Andrews encounter trace.

(UIM) framework using Google Android phones. The set of Bluetooth and WiFi traces were collected by 28 users forweeks in 2010. These people are staff, faculties, graduates, and undergraduates at the University of Illinois. Each UIM experiment phone encompasses a Bluetooth scanner and a Wi-Fi scanner capturing both Bluetooth MAC addresses and Wi-Fi access point MAC addresses in proximity to the phone.

To unify the network interface in simulator, we only used the Bluetooth trace. Considering the constraint of buffer and external event queue, only 7017 external encounter events from February 26, 2010 07:00:42 to February 26, 2010 22:59:57 were employed. We found seven different MAC addresses in this period.

We tested the performance with different message overload. The intervals of message generator were varied from 100 s to 500 s. Figures 11(a), 11(b), and 11(c) showed the sim ulation results using Illinois encounter trace. Like the results in St. Andrews encounter trace, EP and PRO2 still have higher average delay than other copy constrained algorithms. The average delay rates of OJEDF-H and OJEDF-T were 6.09% and 12.14% lower than that of SNW, respectively. The delivery rate increased when the overload decreased. All of the delivery rates are lower than 21%. This is because about 38.08% of the total 7017 contacts occurred between user 3 and user 4 or between user 4 and user 6. These unevenly distributed contacts led to low delivery rate when we only

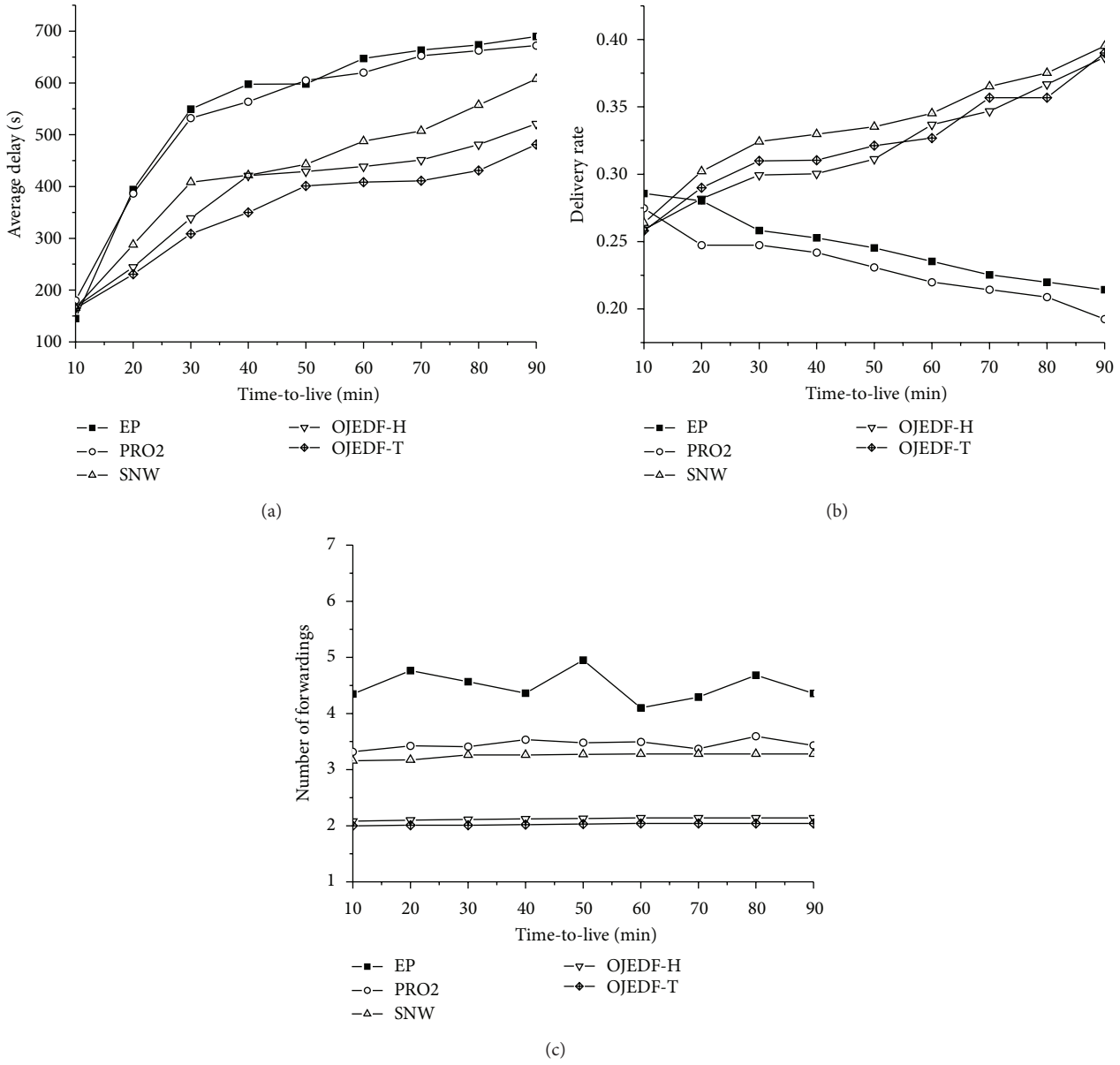


Figure 10: (a) Average delay with different TTLs using St. Andrews encounter trace. (b) Delivery rate with different TTLs using St. Andrews encounter trace. (c) Number of forwardings with different TTLs using St. Andrews encounter trace.

employed the encounter trace in one day. Different from Saint Andrews encounter trace, we set a big buffer relative to message size, so the delivery rates of EP and PRO2 were higher than those of OJEDF-H and OJEDF-T. The delivery rates of OJEDF-H and OJEDF-T were 15.55% and 9.66% lower than that of SNW in average, respectively. OJEDF still had the least forwardings for each message. The numbers of forwardings of OJEDF-H and OJEDF-T were 10.90% and 11.86% lower than that of SNW, respectively.

We compared the performance with different TTLs. Large TTLs (from 100 minutes to 500 minutes) were used in our simulations. Firstly, the valid encounters lasted for 230301 seconds at 02-26-2010 in Illinois encounter trace. Secondly, the encounters were unevenly distributed between seven persons. As a result, many copies need more time to find useful encounters during all simulation time and to meet the destinations ultimately.

Figures 12(a), 12(b), and 12(c) showed the performance of different algorithms with different TTLs using Illinois encounter trace. The average delay rates of EP and PRO2 were lower than that of SNW. This profited from flooding based diffusion and sufficient buffer. OJEDF outperformed all protocols in terms of the average delay with all TTLs. The average delay rates of OJEDF-H and OJEDF-T were 47.93% and 52.76% lower than that of SNW, respectively. The delivery rates of all protocols increased with increasing TTLs in Illinois encounter trace scenario. The delivery rates of OJEDF-H and OJEDF-T were 28.61% and 21.91% lower than

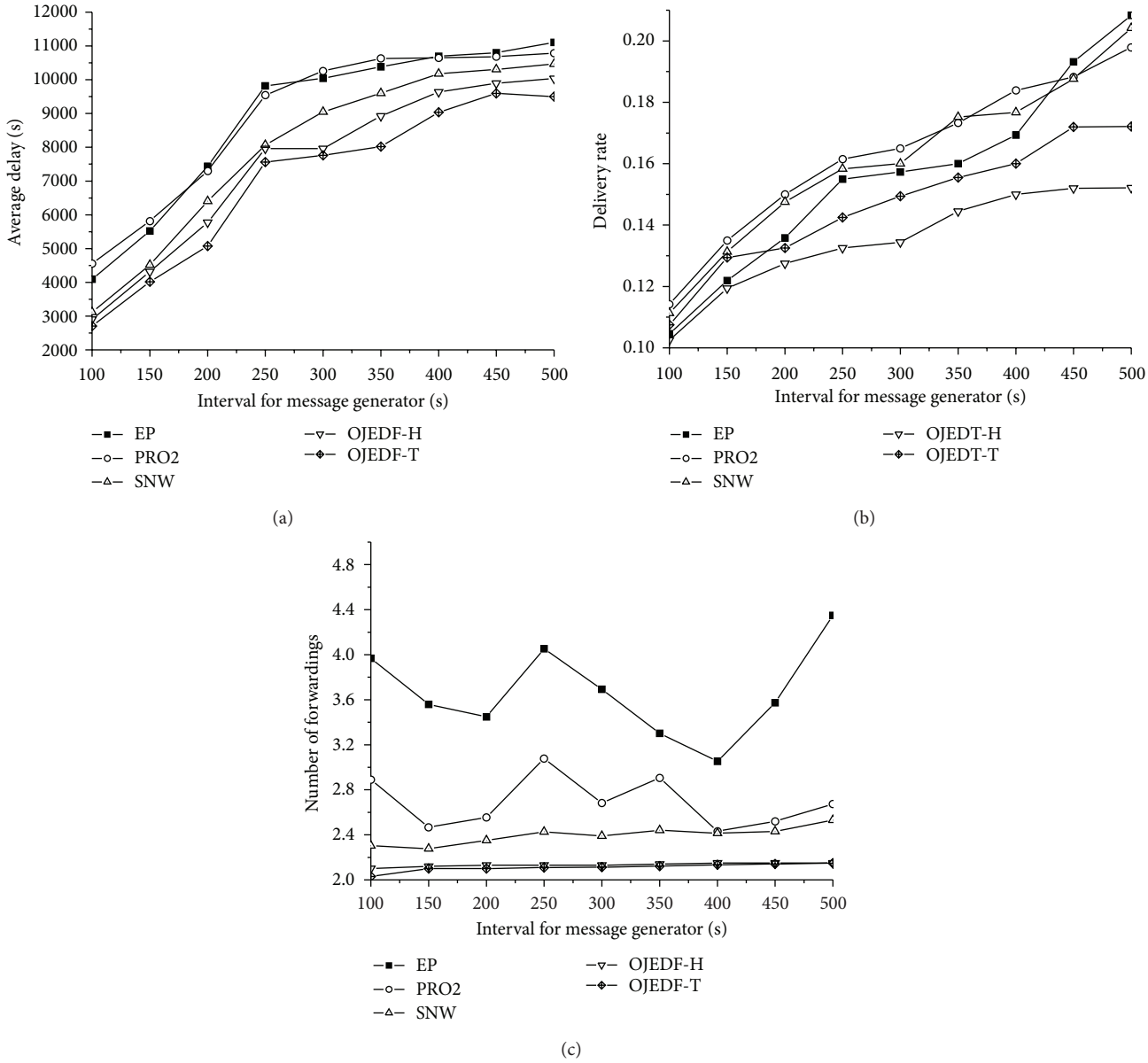


Figure 11: (a) Average delay with different interval for message generator using Illinois encounter trace. (b) Delivery rate with different interval for message generator using Illinois encounter trace. (c) Number of forwardings with different interval for message generator using Illinois encounter trace.

that of SNW, respectively. But the numbers of forwardings of OJEDF-H and OJEDF-T were 11.65% and 12.37% lower than that of SNW, respectively.

6.3. Summary of Simulation. Simulation results confirm that OJEDF outperforms all protocols in terms of average delivery delay and forwarding cost using both St. Andrews encounter trace and Illinois encounter trace. The delivery delay and the number of forwardings of OJEDF-H are 18% and 23% lower than those of SNW while decreasing only 15% delivery rate. OJEDF-T is more accurate than OJEDF-H in terms of copy control. The delivery delay and the number of forwardings of OJEDF-T are 24% and 25% lower than those of SNW while decreasing only 12% delivery rate.

7. Conclusion

In this paper, we provide an optimal forwarding protocol which minimizes the expected delay while satisfying the constant on the number of forwardings per message. We propose the optimal joint expected delay forwarding which makes optimal forwarding decisions by modeling forwarding as an optimal stopping rule problem. We firstly employ the comprehensive dynamic forwarding metric called *joint*

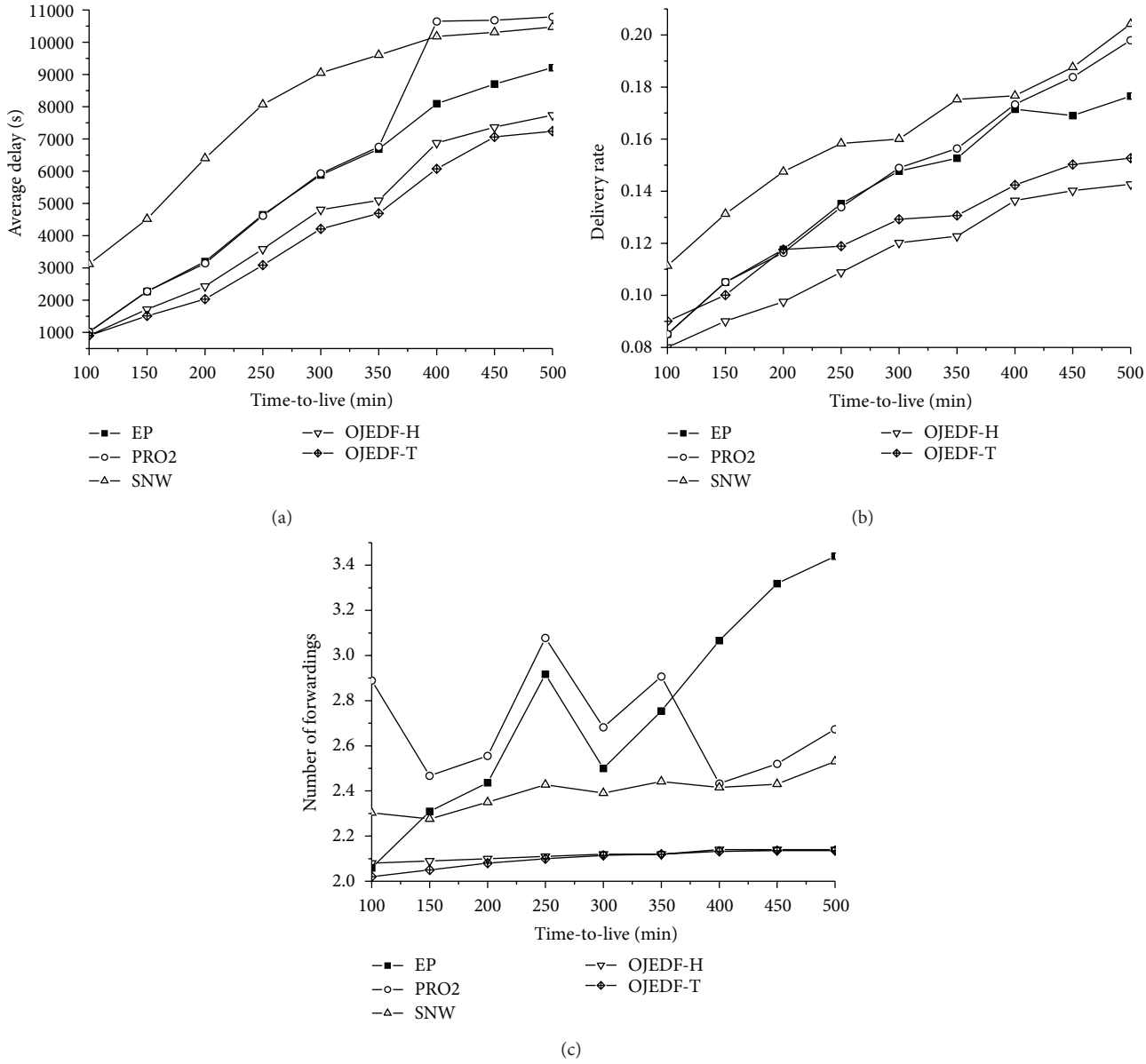


Figure 12: (a) Average delay with different TTLs using Illinois encounter trace. (b) Delivery rate with different TTLs using Illinois encounter trace. (c) Number of forwardings with different TTLs using Illinois encounter trace.

expected delay (JED) which is a function of two important states of a message copy: remaining hop-count and residual lifetime. Then we propose the forwarding rules to decide whether to forward the copies. Based on the hop-count constraint forwarding scheme, we propose a general ticket constraint OJEDF which can be viewed as a development of Spray and Wait with special spray metric. We also present an extension to allow OJEDF to run in delay constrained scenarios. We implemented OJEDF as well as several other protocols and performed trace-driven simulations. Simulation results verified the efficiency of OJEDF.

In the future, we will perform simulations on the extended version of OJEDF, that is, estimating the minimum hop-count or tickets to meet the constrained delay. We will also do research on the expansibility and adaptability of OJEDF when only partial routing information is known.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (61100199), the Scientific and Technological Support Project (social development) of Jiangsu Province (BE2013666), the Natural Science Key Fund for Colleges and Universities in Jiangsu Province (12KJA520002), the Special Fund of postdoctor (2013T60553), the Jiangsu Provincial Natural Science Foundation of China (BK2011692), the Postdoctoral Foundation

(20110491453), the Jiangsu Planned Projects for Postdoctoral Research Funds (1101126c), the talents project of Nanjing University of Posts and Telecommunications (NY210077), and the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (yx002001).

References

[1] V. Cerf, S. Burleigh, A. Hooke et al., "Delay Tolerant Networking Architecture," DTN Research Group, 2007, http://tools.ietf.org/html/rfc4838.

[2] F. Kevin and F. Stephen, "The Delay-Tolerant Networking Research Group (DTNRG)," 2010, http://www.dtnrg.org/wiki/Home.

[3] Y.-P. Xiong, L.-M. Sun, J.-W. Niu, and Y. Liu, "Opportunistic networks," *Journal of Software*, vol. 20, no. 1, pp. 124–137, 2009.

[4] V. Amin and B. David, "Epidemic routing for partially connected ad hoc networks," CS-2000-06, Department of Computer Science, Duke University, Durham, NC, USA, 2000.

[5] M. K. Bromage, J. T. Koshimoto, and K. Obraczka, "TAROT: trajectory-assisted routing for intermittently connected networks," in *Proceedings of the Annual International Conference on Mobile Computing and Networking Workshops (MobiCom '09) and 4th ACM Workshop on Challenged Networks (CHANTS '09)*, pp. 9–17, September 2009.

[6] B. Burns, O. Brock, and B. N. Levine, "MV routing and capacity building in disruption tolerant networks," in *Proceedings of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 398–408, March 2005.

[7] A. D. Lindgren and O. Schelèn, "Probabilistic routing in intermittently connected networks," *Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.

[8] F.-L. Xu, M. Liu, H.-G. Gong, G.-H. Chen, J.-P. Li, and J.-Q. Zhu, "Relative distance-aware data delivery scheme for delay tolerant mobile sensor networks," *Journal of Software*, vol. 21, no. 3, pp. 490–504, 2010.

[9] E. P. C. Jones, L. Li, and P. A. S. Ward, "Practical routing in delay-tolerant networks," in *Proceedings of the ACM Conference on Computer Communications (SIGCOMM '05)*, 2005.

[10] Z. Guo, B. Wang, and J.-H. Cui, "Prediction assisted single-copy routing in underwater delay tolerant networks," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, December 2010.

[11] J. P. Singh, N. Bambos, B. Srinivasan, D. Clawin, and Y. Yan, "Proposal and demonstration of link connectivity assessment based enhancements to routing in mobile ad-hoc networks," in *Proceedings of the IEEE 58th Vehicular Technology Conference (VTC '03)*, pp. 2834–2838, October 2003.

[12] C. Mascolo and M. Musolesi, "SCAR: context-aware adaptive routing in delay tolerant mobile sensor networks," in *Proceedings of the ACM International Wireless Communications and Mobile Computing Conference (IWCMC '06)*, pp. 533–538, July 2006.

[13] C. Liu and J. Wu, "An optimal probabilistic forwarding protocol in delay tolerant networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, 2009.

[14] S. F. Thomas, "Optimal Stopping and Applications," 2008, http://www.math.ucla.edu/~tom/Stopping/Contents.html.

[15] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the ACM Conference on Computer Communications (SIGCOMM '05)*, pp. 252–259, August 2005.

[16] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti, "Exploiting self-reported social networks for routing in ubiquitous computing environments," in *Proceedings of the 4th IEEE International Conference on Wireless and Mobile Computing, Networking and Communication (WiMob '08)*, pp. 484–489, October 2008.

[17] V. Long, D. Quang, and N. Klara, "3R: fine-grained encounter-based routing in delay tolerant networks," in *Proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM '11)*, 2011.

[18] J. Leguay, T. Friedman, and V. Conan, "DTN routing in a mobility pattern space," in *Proceedings of the ACM Conference on Computer Communications (SIGCOMM '05)*, pp. 276–283, August 2005.

[19] M. Musolesi, S. Hailes, and C. Mascolo, "Adaptive routing for intermittently connected mobile Ad Hoc Networks," in *Proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM '05)*, 2005.

[20] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom '07)*, pp. 79–85, March 2007.

[21] T. Spyropoulos, T. Turletti, and K. Obraczka, "Utility-based message replication for intermittently connected heterogeneous networks," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM '07)*, June 2007.

[22] A. Jindal and K. Psounis, "Optimizing multi-copy routing schemes for resource constrained intermittently connected mobile networks," in *Proceedings of the 40th Asilomar Conference on Signals, Systems, and Computers (ACSSC '06)*, pp. 2142–2146, November 2006.

[23] L. Zhuoqun, S. Lingfen, and E. C. Ifeachor, "Adaptive multi-copy routing for intermittently connected mobile ad hoc networks," in *Proceeding of the IEEE Global Telecommunications Conference (GLOBECOM '06)*, December 2006.

[24] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 63–76, 2008.

[25] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multiple-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, 2008.

[26] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 251–259, May 2008.

[27] A. Balasubramanian, B. Levine, and A. Venkataramani, "DTN routing as a resource allocation problem," in *Proceedings of the ACM Conference on Computer Communications (SIGCOMM '07)*, pp. 373–384, August 2007.

[28] C. Liu and J. Wu, "An optimal probabilistic forwarding protocol in delay tolerant networks," in *Proceedings of the 9th ACM*

International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09), 2009.

[29] C. Liu, J. Wu, and I. Cardei, "Message forwarding in cyclic MobiSpace: the multi-copy case," in *Proceedings of the IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (MASS '09)*, pp. 413–422, October 2009.

[30] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Pocket Switched Networks: real-world mobility and its consequences for opportunistic forwarding," Tech. Rep. UCAM-CL-TR-617, Computer Laboratory, University of Cambridge, 2005.

[31] A. Pentland, R. Fletcher, and A. Hasson, "DakNet: rethinking connectivity in developing nations," *IEEE Computer*, vol. 37, no. 1, pp. 78–4, 2004.

[32] A. Beaufour, M. Leopold, and P. Bonnet, "Smart-tag based data dissemination," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 68–77, September 2002.

[33] K. Ari, J. Jörg, and K. Teemu, "The ONE simulator for DTN protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation (Simutools '09)*, 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 646781, 7 pages
http://dx.doi.org/10.1155/2013/646781

Research Article

LIRT: A Lightweight Scheme for Indistinguishability, Reachability, and Timeliness in Wireless Sensor Control Networks

Wei Ren,[1] Liangli Ma,[2] and Yi Ren[3]

[1] *School of Computer Science, China University of Geosciences, Lumo Road 388, Wuhan 430074, China*
[2] *Department of Computer Engineering, Naval University of Engineering, Jiefang Road 717, Wuhan 430033, China*
[3] *Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan*

Correspondence should be addressed to Wei Ren; weirencs@cug.edu.cn

Received 30 June 2013; Accepted 25 September 2013

Academic Editor: Hongli Xu

Wireless sensor control networks (WSCNs) are important scenarios and trends in mobile wireless sensor networks. Compared with traditional wireless sensor networks, WSCNs have two specific characteristics: entities in networks are extended from passive sensors to active sensors (e.g., actuators and actors) and the transmitting messages are extended from data only to data plus (e.g., data and control instructions). Thus new security problems arises. In this paper, we make the first attempt to specify the security requirements for WSCNs, namely, indistinguishability, reachability, and timeliness. In addition, several new attacks in WSCN, distinguishing risks, dropping attacks, and disordering attacks, are pointed out at the first time. A lightweight scheme LIRT is proposed with tailored design to guarantee the indistinguishability, reachability, and timeliness in WSCNs. Extensive and rigorous analysis on LIRT justifies its security strength and performance measures.

1. Introduction

Mobile wireless sensor networks are an extended form of traditional wireless sensor networks where nodes are not static but mobile. Recently, sensors in mobile wireless sensor networks become more and more versatile, for example, underwater sensors, body sensors, and control sensors. The control sensors are usually divided into two classes: actuators and actors. Those actuator and actor sensor networks extend traditional wireless sensor networks from passive networks to active networks, from data networks to control networks, via adding functionalities such as response and action. The actuator and actor sensor networks start to be applied in new applications such as smart grid, smart city, smart building, and factory automation, to name a few [1–3].

Wireless actuator and actor sensor networks can be viewed as wireless sensor control networks (WSCNs) over a group of sensors. WSCNs have two distinctions compared with traditional wireless sensor networks as follows. (1) The entities in networks are extended from sensors only to sensors plus. For example, there exist sensors, actuators, and actors in WSCNs. Actuators may perform actively for controlling, but sensors in traditional wireless sensor networks usually act passively for collecting data. (2) The transmitting messages in networks are extended from data only to data plus, for example, data messages and control messages. Therefore, new security problems arise in WSCNs. If entities in WSCNs can be distinguished by adversaries, adversaries will be able to launch a target attack (that has been explored in our previous paper [4]); if data or control messages are dropped by adversaries, the control loop will be terminated; if data or control messages are disordered, the control status or sequences may be disturbed. We called them indistinguishability, reachability, and timeliness problems in WSCNs. Note that those security problems cannot be solved by previous security schemes for traditional wireless sensor networks due to the specialities of WSCNs. We thus have to explore new methods to solve them, especially, in a tailored design manner.

Concretely, security in wireless control networks starts to attract more and more attention [5–9]. Those work majorally address different contexts from WSCNs, so the solutions may not be able to tackle the aforementioned security requirements. Moreover, the security problems in WSCNs are challengeable due to the inherent properties: wireless lossy channels, jamming-sensitive links, resource-constraint sensor devices, control timing demands, and control sequence ordering requirements.

In this paper, we make the first attempt to clarify and analyze the security requirements in WSCNs and then propose a lightweight scheme called LIRT to guarantee those requirements, namely, indispensability, reachability, and timeliness in WSCNs. We formally prove the achievement of the proposed scheme. Different from other works and previous approaches, all presentations in the paper strictly follow formal expressions for better clarity and rigorous generality.

The contributions of the paper are listed as follows.

(i) We make the first attempt to define formal attacks and security requirements in WSCNs, namely, indistinguishability, reachability, and timeliness in WSCNs.

(ii) We make the first attempt to propose a lightweight scheme to guarantee those security requirements and formally prove the security goals that are achieved.

The rest of the paper is organized as follows. Section 2 gives an overview on relevant prior work. In Section 3 we discuss the basic assumption and models used throughout the paper. Section 4 provides the detailed description and analysis of our proposed scheme. Finally, Section 5 concludes the paper.

2. Related Work

Wireless sensor networks for automation control have attracted more and more attention in recent years [5, 8, 10–12]. Yen et al. [5] proposed packet loss problem in wireless networked control system over IEEE 802.15.4e. They proposed redundant transmission. de Filippi et al. [7] proposed single-sensor control strategies for semiactive steering damper control in two-wheeled vehicles. Thurman et al. [9] explored acoustic sensors in an unmanned underwater vehicle to provide full autonomy control. Au et al. [13] proposed energy-efficient classification algorithms for wearable sensor systems. All the above work focuses on control performance but not control security.

The security problems in WSCNs have not been thoroughly explored in recent work. Target attacks for wireless machine-to-machine control networks are first pointed out by our previous work [4]. We also proposed a scheme called RISE to mitigate target attacks. Stealthy deception attacks in water SCADA systems are first pointed out by Amin et al. [6]. They discuss sensor networks but mainly in wired SCADA networks. Zheng et al. [3] discussed reliable problem in wireless communication networks that support demand and response control. They proposed several methods for deriving reliability performance. Short et al. [2] discussed burst errors in wireless control networks.

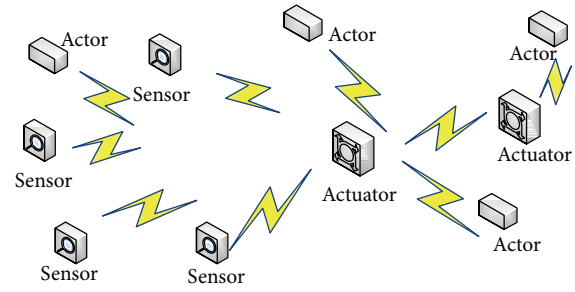


Figure 1: Wireless sensor control networks. Sensors collect sensing data; actuators respond corresponding instructions; actors execute those instructions.

They proposed application-level strategies for ameliorating the effects of packet losses and burst errors in sampled-data control systems. Above related work are independent with our discussion and solution in the following, as our analysis for the security requirements are different from them.

3. Problem Formulation

3.1. Network Model and Attack Model. There exist three major entities (denoted as E) in WSCNs: sensors (denoted as S), actuator (denoted as U), and actors (denoted as C). Usually, sensors send data to actuators. Actuators send instructions to actors. Actors execute instructions. The number of sensors is usually more than that of actuators. The number of actuators is usually less than that of actors. Figure 1 depicts the entities in WSCNs.

We assume that the links among sensors, actuators, and actors are not secure. The adversaries (denoted as A) in the links can launch the following attacks. We assume that the security boundary is out of the entities of WSCNs. That is, we assume entities are trustworthy. The trust models in WSCN scenarios are analyzed in detail in our previous work [14].

Definition 1 (message distinguishing risk (R_m)). Adversaries may distinguish data and instructions in transmitting messages in WSCNs, after viewing the behavior and messages among entities in WSCNs.

The observation is the only advantage of adversaries, as we suppose the links are not secure. It can be formally described as follows:

$$\mathsf{R}_m = \Pr\Big\{\mathsf{A}\ \text{correctly guesses}\ m^* \in D \parallel m^* \in I \mid m^* \Longleftarrow D \cup I, 2^{\mathsf{E}_i \to \mathsf{E}_i:\{m^*\}}, \mathsf{A} \Longleftarrow m^*\Big\}, \tag{1}$$

where $\Pr\{A \mid B\}$ denotes the probability that A happens after event B happens; D means data; I means instructions; m^* is any message in D or I; 2^{ϕ} means the power set of a set ϕ.

Definition 2 (entity distinguishing risk (R_e)). Adversaries may distinguish sensors, actuators, and actors among entities in WSCNs, after viewing the behavior and messages among entities in WSCNs.

It can be formally described as follows:

$$\mathsf{R}_e = \Pr\Big\{\mathsf{A} \text{ correctly guesses } \mathsf{E}_i \in \mathsf{S}\,(\|\mathsf{U}\|\,\mathsf{C}) \mid E_i \Longleftarrow \mathsf{S}\,(\|\mathsf{U}\|\,\mathsf{C}), 2^{\mathsf{E}_i \to \mathsf{E}_j:\{m^*\}}, \mathsf{A} \Longleftarrow m^*\Big\}. \tag{2}$$

Definition 3 (dropping attack (A_d)). Adversaries may drop data that are sent from sensors to actuators and instructions that are sent from actuators to actors.

Definition 4 (disordering attack (A_i)). Adversaries may disturb the arrival time of data at actuators and the arrival time of instructions at actors.

It is natural to see that the prerequisite for dropping attack is message distinguishing risk and entity distinguishing risk.

The disordering attack can be launched without any prerequisite information about message distinguishing risk and entity distinguishing risk. It is thus much easier to be launched via just jamming arbitrary packets into channels, and it is thus more difficult to be defended against.

3.2. Security Definition and Design Goal. The security requirements are defined as follows.

Definition 5 (indistinguishability). The data and instruction cannot be distinguished from messages by adversaries from all their observations. The sensors, actuators, and actors cannot be distinguished from entities by adversaries from their observations.

Indistinguishability is formally described as

$$H\,(\mathsf{S} \mid O) = H\,(\mathsf{U} \mid O) = H\,(\mathsf{C} \mid O) = H\,(\mathsf{S}) = H\,(\mathsf{U}) = H\,(\mathsf{C}), \tag{3}$$

where O is the observation of adversaries; H is the entropy of correctly guessing on entities.

Definition 6 (reachability). The data can arrive at designated actuators finally, and instructions can arrive at designated actors finally.

Definition 7 (timeliness). The data can arrive at actuators timely, and instructions can arrive at actors timely.

Therefore, the design goal is to propose a scheme for guaranteeing indistinguishability, reachability, and timeliness in a lightweight way.

4. Proposed Schemes

We list major notations used in the remainder of the paper in Table 1.

4.1. Indistinguishability. As message distinguishing risk, R_m, and entity distinguishing risk, R_e, are the prerequisite of dropping attack, A_d, we first propose a method to eliminate those risks.

Table 1: Notations.

WSCNs	Wireless sensor control networks
E	Entities
S	Sensors
U	Actuators
C	Actors
A	Adversaries
R_m	Message distinguishing risk
R_e	Entity distinguishing risk
A_d	Dropping attack
A_i	Disordering attack

Proposition 8. *Entity indistinguishability is equivalent to message indistinguishability.*

Proof (straightforward). If entities are distinguishable, messages will be distinguishable via the entities who send; if messages are distinguishable, entities will be distinguishable by analyzing their sending messages. □

Thus, we discuss two risks together. We firstly analyze the information or advantages that can be obtained by adversaries. Adversaries can observe the following behavior and messages in WSCNs.

The messages that can be observed for message distinguishing are as follows:

(M-O1) the length of the messages that are sent among entities,

(M-O2) the format and semantics of the messages that are sent among entities.

The behavior that can be observed for distinguishing entities is as follows.

(B-O1) The sending sequence of the messages and entities, it is a list of messages and entities which send messages in an observing time span. For example, in an observation time span with k minutes, the sending sequences of messages are $\{m_1, m_2, \ldots, m_n\}$. In an observation time span with k minutes, the sending sequences of entities who send messages are $\{E_1, E_2, \ldots, E_n\}$, where $E_i \in \mathsf{E}$, $i = 1, \ldots, n$. The sequences can be observed for distinguishing entities. For example, sensors may always stand at the head of the sequence, and actors may always stand at the rear of the sequence.

(B-O2) The interval of two sequentially sending messages at any two entities; for example, E_i sends m_p, and then E_j sends m_{p+1}. m_p, m_{p+1} are two consecutively sending messages. The time interval between these two messages can be observed for distinguishing entities. For example, actuators may always send a message after sensors send a message, and actors may always send a message after actuator sends a message.

(B-O3) The interval of two consecutively sending messages at one entity, for example, E_i sends m_p and

m_{p+1} sequentially, where $E_i \in \mathsf{E}, m_p, m_{p+1}$. That is, m_p, m_{p+1} are two sequential messages sent from E_i. The time interval between these two messages can be observed for distinguishing entities. For example, sensors may always send messages in a fixed interval.

(B-O4) The interval of k sequentially sending messages at any k entities, it is a generalization of (B-O2). For example, suppose k entities send k messages sequentially. They are $E_1, E_2, \ldots, E_k$. The time interval among them can be observed for distinguishing entities. For example, sensors, actuators, and actors may form a control loop. Observing loops may infer the entity observed.

(B-O5) The interval of k sequential sending messages at one entity, it is a generalization of (B-O3). For example, E_i sends k messages: $m_1, m_2, \ldots, m_k$, where $E_i \in \mathsf{E}$. The time interval among them can be observed for distinguishing entities. For example, intervals for message sending at sensors, at actuators, or at actors may be quite different. Observing those difference may infer the entity observed.

Proposition 9. *If and only if the observation is indistinguishable, the message and entity are indistinguishable.*

Proof. The observation at adversaries is the only knowledge to distinguish message and entity. If and only if the observation is indistinguishable, the message and entity are indistinguishable. □

Therefore, we propose the following strategies via randomization to make observation indistinguishable. Each strategy addresses one observation.

(IND-S1) All messages that are sent among entities have the same length.

(IND-S2) All messages that are sent among entities are encrypted for hiding semantics.

(IND-S3) The sending sequence of the messages among entities is randomized.

(IND-S4) The interval of two sequentially sending messages at any two entities is randomized.

(IND-S5) The interval of two sequentially sending messages at one entity is randomized.

(IND-S6) The intervals of k sequentially sending messages at any k entities are randomized.

(IND-S7) The intervals of k sequentially sending messages at one entity are randomized.

Proposition 10. *Strategy (IND-S6) can be guaranteed by (IND-S4).*

Proof (straightforward). As any interval of two sequentially sending messages at any two entities is randomized, and the intervals of k sequentially sending messages at any k entities in k are also randomized. □

```
Date: M = {D ‖ I ‖ NULL}, W
Result: Sending Packets with Indistinguishability
Initialization;
While T do
    M ⇐ Get Out Queue Buffer();
    //Get packet from Outgoing Queue
    t ⇐ Random() %W;
    //W is the suspended time slot
    Suspend(t);
    if (M <> NULL) then
        TempPkt ⇐ ExtendToFixLen(M);
        //Maintain the same length
    else
        TempPkt ⇐ Create FixLen Dummy();
        //Create dummy packet
    end
    M' ⇐ MaskMsg(TempPkt);
    //Encryption before sending
    SendMsg(M');
end
```

Algorithm 1: Sending algorithm for indistinguishability (SAI algorithm).

Proposition 11. *Strategy (IND-S7) can be guaranteed by (IND-S5).*

Proof (straightforward). The interval of two sequentially sending messages at one entity is randomized, the intervals of k sequentially sending messages at one entity are thus also randomized. □

Thus, the sending algorithm for indistinguishability (called SAI algorithm) at each entity is proposed in Algorithm 1.

4.2. Reachability. As adversaries cannot distinguish messages and entities, they have to drop messages (e.g., by jamming channels) randomly to launch a dropping attack.

To guarantee the reachability of the data and instruction messages, we propose the following strategies via redundancy.

(RCH-S1) Data and instruction are sent for α times.

Proposition 12. *If the dropping probability of a packet is p, strategy (RCH-S1) can guarantee the reachability with probability $1 - p^\alpha$.*

Proof. As the dropping probability of one packet is p, its reachability is $1 - p$. The probability of α packets that are dropped is p^α, and the reachability of at least one in α packets is thus $1 - p^\alpha$. □

The repeat sending for α times can increase the probability of reachability, but it also causes communication overhead. In the following strategies, we will tackle the communication overhead by optimization.

(RCH-S2) Data and instruction are sent for random times in $[\beta_1, \beta_2]$. The repeat times are varied. Usually,

we have $\alpha = \beta_2$. Therefore, it can both increase the probability of reachability and tackle the communication overhead.

Proposition 13. *If the dropping probability of a packet is p, strategy (RCH-S2) can guarantee the reachability with probability at least* $(1/(\beta_2 - \beta_1)) \sum_{\beta_1}^{\beta_2}(1 - p^i)$.

Proof. The reachability of one packet in $[\beta_1, \beta_2]$ is $1 - p^i$, $i \in [\beta_1, \beta_2]$. The expectation of this probability for all i, $i \in [\beta_1, \beta_2]$ is thus $(1/(\beta_2 - \beta_1)) \sum_{\beta_1}^{\beta_2}(1 - p^i)$. □

Proposition 14. *The average communication overhead in (RCH-S2) is less than (RCH-S1) by* $1 - (\beta_1 + \beta_2)/2\alpha$.

Proof (straightforward). The communication overhead in (RCH-S1) is $O(\alpha)$; the communication overhead in (RCH-S2) is $O((\beta_1 + \beta_2)/2)$. Thus, the advantages in (RCH-S2) compared with (RCH-S1) are $(\alpha - (\beta_1 + \beta_2)/2)/\alpha$. That is, $1 - (\beta_1 + \beta_2)/2\alpha$. □

(RCH-S3) The repeating times at originators for data or instruction are γ_1. The repeating times at forwarders for dummy packets are γ_2. Usually, $\gamma_1 \ll \beta_1, \gamma_1 + \gamma_2 \approx \alpha$.

Originators are the entities where data or instruction are originated from. For example, the first sensor who sends the data is the originator for that data. Forwarders are the entities between originators and designated destination entities. That is, forwarders forward the data or instruction to packet destination. The dummy packets at immediate forwarders are not created from meaningless dummy string (NULL) but created from data or instruction received previously by immediate forwarders. That is, before forwarders send dummy packets, they choose the last one in received data or instruction as a dummy packet. This strategy can both improve the reachability of data or instruction and mitigate the communication overhead.

Proposition 15. *If the dropping probability of a packet is p, strategy (RCH-S3) can guarantee the reachability with probability* $1 - p^{\gamma_1+\gamma_2}$. *The communication overhead is* γ_1/α *of that in strategy (RCH-S1).*

Proof (straightforward). The proof is similar to the former proposition. □

The sending algorithm for reachability (SAR algorithm) at each entity is given in Algorithm 2.

```
Date: M = {D ‖ I ‖ NULL}, γ1, γ2, W
Result: Sending Packets for Reachability
//at Originators:
Initialization;
while T then
  M ⇐ Get Out Queue Buffer();
  if (M <> NULL) then
    Count ⇐ 0;
    while Count < γ1 do
      t ⇐ Random() %W;
      //W is suspended time slot
      Suspend(t);
      TempPkt ⇐ ExtendToFixLen(M);
      M′ ⇐ MaskMsg(TempPkt);
      SendMsg(M′);
      Count + +;
    end
  else
    t ⇐ Random() %W;
    Suspend(t);
    TempPkt ⇐ CreateFixLenDummy();
    M′ ⇐ MaskMsg(TempPkt);
    SendMsg(M′);
  end
end
//at Forwarders:
Initialization;
while T do
  M ⇐ Get In Queue Buffer();
  //Get packet from ingress queue
  if (M <> NULL) then
    Count ⇐ 0;
    while (Count < γ2) do
      t ⇐ Random() %W;
      Suspend(t);
      TempPkt ⇐ Extend To FixLen(M);
      M′ ⇐ MaskMsg(TempPkt);
      SendMsg(M′);
      Count + +;
    end
  else
    t ⇐ Random() %W;
    Suspend(t);
    TempPkt ⇐ Create FixLen Dummy();
    M′ ⇐ MaskMsg(TempPkt);
    SendMsg(M′);
  end
end
```

Algorithm 2: Sending algorithm for reachability (SAR algorithm).

4.3. Timeliness. Adversaries cannot distinguish messages and entities. The dropping attack cannot aim at designated messages or entities. The dropping is thus randomly dropping, for example, by jamming channels. The jamming subsequently results in disordering attack. The timeliness of the control operations is damaged. To guarantee the timeliness of the data and instruction messages, we propose following strategies.

(TML-S1) The suspended time is randomly chosen from a time slot that is shortened exponentially. That is, $W_s \overset{r}{\Leftarrow} 1/2^n * W$, where W is the maximal suspended time slot at the first time; n is the suspending times; $\overset{r}{\Leftarrow}$ means "is randomly chosen from";

W_s is the actual suspended time. The timeliness can be improved with the exponentially shortening of suspended time. This strategy is corresponding to (RCH-S1).

Proposition 16. *If the suspended time slot W in α times is shortened to $1/2^{\alpha}W$, strategy (TML-S1) can guarantee the timeliness with total suspended time $\sum_{i=1}^{\alpha}(1/2^i) * W$.*

Proof. Suppose the suspended time slot is W, the suspended time in expectation is $1/2W$. If the suspended time slot is shortened to $1/2^n * W$, the suspended time in expectation is $1/2^{n+1} * W$. If first $\alpha - 1$ are all dropped by adversaries, the worst suspended time is $\sum_{i=1}^{\alpha}(1/2^i) * W$. □

Similarly, (TML-S2) can be proposed corresponding to (RCH-S2). That is, when data and instruction are sent for random times in $[\beta_1, \beta_2]$, the suspending time between two consecutively sending is randomly chosen from a time slot that is shortened exponentially. That is, $W_s \overset{r}{\Leftarrow} 1/2^n * W$.

(TML-S3) We propose to shorten the suspended time slot exponentially at forwarders. The minimum is lower bounded by a threshold value, denoted as Th.

The sending algorithm for timeliness (SAT algorithm) at each entity is given in Algorithm 3.

Proposition 17. *Strategy (TML-S3) does not damage indistinguishability.*

Proof. Everyone in WSCNs may be originators or forwarders. It depends on messages to forwarder or originator. Originators or forwarders both shorten suspended time slot exponentially. Thus, strategy (TML-S3) does not damage indistinguishability. □

Proposition 18. *If the suspended time slot W in γ_1 times is shortened to $1/2^{\gamma_1}W$ at originators and in γ_2 times is shortened to $1/2^{\gamma_2}W$, strategy (TML-S1) can guarantee the timeliness with time $(\sum_{i=1}^{\gamma_1}(1/2^i) + \sum_{i=1}^{\gamma_2}(1/2^i)) * W$.*

Proof (straightforward). The proof is similar to the proof of the former proposition. □

The proposed scheme—LIRT—is the combination of strategies for indistinguishability, reachability, and timeliness. As the strategy (TML-S3) includes SAI, SAR, and SAT, it can be viewed as an appropriate version of LIRT. As the scheme is described intentionally in an incremental manner in this section, the advantages of LIRT are clear to follow for its advantages due to the improvements step by step.

```
Date: M = {D ‖ I ‖ NULL}, γ1, γ2, W
Result: Sending Packets for Timeliness
//at Originators: Initialization;
While T do
  M ⇐ Get Out Queue Buffer();
  if (M <> NULL) then
    Count ⇐ 0;
    W' ⇐ W;
    While (Count < γ1) do
      t ⇐ Random()% W';
      W' ⇐ max(1/2 * W', Th);
      //Exponentially suspending
      Suspend(t);
      TempPkt ⇐ Extend To FixLen(M);
      M' ⇐ MaskMsg(TempPkt);
      SendMsg(M');
      Count + +;
    end
  else
    t ⇐ Random() %W;
    Suspend(t);
    TempPkt ⇐ Create FixLen Dummy();
    M' ⇐ MaskMsg(TempPkt);
    SendMsg(M');
  end
end
//at Forwarders: Initialization;
while T do
  M ⇐ Get In Queue Buffer();
  if (M <> NULL) then
    Count ⇐ 0;
    W' ⇐ W;
    while (Count < γ2) do
      t ⇐ Random() %W';
      W' ⇐ max(1/2 * W', Th);
      Suspend(t);
      TempPkt ⇐ Extend To FixLen(M);
      M' ⇐ MaskMsg(TempPkt);
      SendMsg(M');
      Count + +;
    end
  else
      t ⇐ Random() %W;
      Suspend(t);
      TempPkt ⇐ Create FixLen Dummy();
      M' ⇐ MaskMsg(TempPkt);
      SendMsg(M');
  end
end
```

Algorithm 3: Sending algorithm for timeliness (SAT algorithm).

5. Discussion

In former discussion, feedback information such as networking status and receiver's acknowledgement is not used for simplicity. If feedback information is available, it can be used to enhance previous strategies by achieving adaptive and optimal overall performance. The feedback information that can be gathered by senders is as follows.

(i) The network feedback on network status, it is sent by intermediate forwarders or detectors, and it reports congestion, risks, and dropping rate of messages.

(ii) The feedback from receivers, it is sent by designated destination of messages, and it reports message arrival, delay, jitter, and timeliness.

If the feedback information is available in WSCNs, the strategies can be enhanced by intelligent method for adaptivity and optimization.

6. Conclusions

WSCNs are important types in mobile wireless sensor networks and present their own characteristics compared with traditional wireless sensor networks. In this paper, we made the first attempt to specify the security requirements for WSCNs, in which of the upmost importance are indistinguishability, reachability, and timeliness. To clarify and illustrate the security requirements, several new attacks in WSCNs were pointed out at the first time, for example, distinguishing risks, dropping attacks, and disordering attacks. To defend against those attacks, a lightweight scheme LIRT was proposed. LIST can guarantee the indistinguishability, reachability, and timeliness in WSCNs, which is justified by extensive and rigorous analysis on security strength. The performance of LIRT is also measured by communication overhead, to confirm its applicability in realistic scenarios.

Acknowledgments

Wei Ren's research was financially supported by the National Natural Science Foundation of China (61170217), the Open Research Fund from the Shandong Provincial Key Laboratory of Computer Network (SDKLCN-2011-01), Fundamental Research Funds for the Central Universities (CUG120109), and Wuhan Planning Project of Science and Technology (2013010501010144). Yi Ren's research was sponsored in part by the "Aim for the Top University Project" of the National Chiao Tung University and the Ministry of Education, Taiwan.

References

[1] M. A. S. Masoum, P. S. Moses, and K. M. Smedley, "Distribution transformer losses and performance in smart grids with residential plug-in electric vehicles," in *Proceedings of the IEEE PES Innovative Smart Grid Technologies (ISGT '11)*, pp. 1–7, January 2011.

[2] M. Short, U. Abrar, and F. Abugchem, "Application level compensation for burst errors in wireless control networks," in *Proceedings of the 17th IEEE Conference on Emerging Technologies Factory Automation (ETFA '12)*, pp. 1–8, 2012.

[3] L. Zheng, N. Lu, and L. Cai, "Reliable wireless communication networks for demand response control," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 133–140, 2013.

[4] W. Ren, L. Yu, L. Ma, and Y. Ren, "RISE: a reliable and secure scheme for wireless machine to machine communications," *Tsinghua Science & Technology*, vol. 18, no. 1, pp. 100–117, 2013.

[5] B. Yen, D. Hop, and M. Yoo, "Redundant transmission in wireless networked control system over IEEE 802.15.4e," in *Proceedings of the International Conference on Information Networking (ICOIN '13)*, pp. 628–631, 2013.

[6] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen, "Cyber security of water scada systems—part I: analysis and experimentation of stealthy deception attacks," *IEEE Transactions on Control Systems and Technology*, vol. 21, no. 1963, p. 1970, 2012.

[7] P. de Filippi, M. Corno, M. Tanelli, and S. M. Savaresi, "Single-sensor control strategies for semi-active steering damper control in two-wheeled vehicles," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 2, pp. 813–820, 2012.

[8] J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch, "Comparative study of energy-efficient sampling approaches for wireless control networks," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 416–424, 2010.

[9] E. Thurman, J. Riordan, and D. Toal, "Real-time adaptive control of multiple colocated acoustic sensors for an unmanned underwater vehicle," *IEEE Journal of Oceanic Engineering*, vol. 38, no. 3, pp. 419–432, 2013.

[10] G. Lee, J. Lee, E. Lee, and D. Kim, "Synchronization algorithm for hybrid control networks: can and wireless control networks," in *Proceedings of the IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS '11)*, pp. 1–4, 2011.

[11] N. Son, D. Tan, and D. Kim, "Backoff algorithm for time critical sporadic data in industrial wireless sensor networks," in *Proceedings of the International Conference on Advanced Technologies for Communications (ATC '12)*, pp. 255–258, 2012.

[12] R. A. Swartz, J. P. Lynch, and C.-H. Loh, "Near real-time system identification in a wireless sensor network for adaptive feedback control," in *Proceedings of the American Control Conference (ACC '09)*, pp. 3914–3919, June 2009.

[13] L. K. Au, A. A. T. Bui, M. A. Batalin, and W. J. Kaiser, "Energy-efficient context classification with dynamic sensor control," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 6, no. 2, pp. 167–178, 2012.

[14] W. Ren, L. Yu, L. Ma, and Y. Ren, "How to authenticate a device? Formal authentication models for M2M communications defending against ghost compromising attack," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 679450, 9 pages, 2013.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 852472, 10 pages
http://dx.doi.org/10.1155/2013/852472

Research Article

Data Dissemination in Mobile Wireless Sensor Network Using Trajectory-Based Network Coding

Lingzhi Li,[1] Shukui Zhang,[1,2] Zhe Yang,[1] and Yanqin Zhu[1]

[1] *School of Computer Science & Technology, Soochow University, Suzhou 215006, China*
[2] *State Key Laboratories for Novel Software Technology, Nanjing University, Nanjing 210093, China*

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

Received 7 June 2013; Accepted 14 September 2013

Academic Editor: Hongli Xu

With GPS devices spreading, more mobile nodes are guided by the navigation systems to select better paths. The performance of mobile networks can be improved with the navigation information in nodes. In this paper, we propose a trajectory-based network coding (TBNC) method to disseminate data in mobile wireless sensor network (MWSN). It is designed according to the characteristics of MWSN and is appropriate for the mobile nodes that share anonymously its GPS pretrajectory for the higher bandwidth. Data are disseminated in the topology that is predicted on the basis of the trajectory information. Network coding is used to adapt for the dynamics velocity of mobile nodes. Simulation results show that TBNC is able to cut down 1/2 overhead messages of PRoPHET when mobile nodes share GPS trajectory. It improves the reliability and scalability of MWSN.

1. Introduction

Wireless sensor network (WSN) is composed of a large number of sensor nodes, which collect the information of surrounding changes. Many researches on WSN are based on the assumption that the sensor node is static. However, the sensor node is mobile in many applications of WSN, such as target tracking, traffic monitoring, urban microclimate, and disaster monitoring [1]. Mobile WSN (MWSN) can simply be defined as the WSN in which the sensor nodes are mobile. It is a wireless ad hoc network that consists of many sensor nodes communicating with each other [2]. The nodes are either equipped with motors for active mobility or attached to mobile objects for passive mobility [3]. The nodes move so fast, and it is difficult to find a path between a pair of nodes using ordinary WSN routing protocol [4].

Mobile nodes can exchange data with each other through Ad-hoc communication networks. But the ordinary ad hoc routing protocols will produce a huge number of overhead messages, which can consume too much energy and add the network load [5]. With GPS devices spreading, more and more mobile nodes are guided by the navigation systems. The prepath and destination of every mobile node are created and stored in its navigation system [6]. The trajectory of mobile node is able to be predicted and shared in the networks. So every node can also predict the topology change of networks according to the pre-path and velocity of other mobile nodes. It doesn't need to send data to all nodes using broadcast such as Epidemic [7] and only forwards data to the nodes moving at the direction of destination.

The researchers have already paid attention to the works applying GPS devices to WSN [8–12]. Many applications of WSNs require the positions where data was sensed, and the former researches are only focused on locating the sensor node in GPS device. These proposed schemes do not require GPS on all sensor nodes, taking into the consideration that GPS units increase the costs of sensor nodes at that time. In this case, only a fraction of the nodes has GPS units, and every node transmits its coordinates to the rest of the sensors to localize themselves [13]. Nowadays, the GPS units are popular to the mobile devices, and they are upgraded from the positioning to navigation. Our work uses the pre-trajectory of mobile nodes to improve the routing protocol of MWSN, and it is different from the researches just positioning nodes.

In this paper, we present a trajectory-based network coding (TBNC) method to disseminate data in MWSN.

Mobile nodes are navigated by GPS devices and share their navigation information with the others. Data are divided into short fixed-size blocks in source node and disseminated to the other mobile nodes. The source implements network coding and sends the coding blocks to the destination according to the pretrajectory of other nodes. The blocks are recoded by the nodes on the cross roads. The experiments show that TBNC is able to reduce overhead messages, and enhance the reliability of transmission. TBNC is appropriate for the MWSN systems that mobile nodes are navigated by GPS devices.

The remainder of this paper is organized as follows. In Section 2, we discuss the related work. Section 3 presents the description and implementation of the TBNC. The evaluation and analysis of the TBNC are shown in Section 4. The paper is concluded in Section 5.

2. Related Works

The GPS devices are only used by mobile nodes of WSN to determinate their locations in many studies. Parts of mobile nodes are equipped with GPS devices to get their coordinates. GPS-equipped node is called a landmark, anchor, or seed [13]. It helps the other nodes without GPS devices to estimate their positions. The landmark moves throughout WSN providing sensor nodes with its location [14]. It either transmits its coordinates to help the other nodes to estimate their positions or receives messages sent by the other nodes and estimates their positions. In recent years, more and more studies have looked at positioning the nodes without any GPS devices in WSN [15, 16]. LIS [17] is a novel localization algorithm. As nodes run a fuzzy logic algorithm for processing the information, it was considered that nodes were provided with some kind of intelligence. They can implement in the area where GPS does not work correctly, such as the interior of buildings [18, 19]. GPS devices are not necessary to determin the location of mobile nodes.

A few studies have been attracted on utilizing GPS information for VANETs [20]. More and more vehicles are equipped with satellite navigation systems. With increased usage of smart phones like Apple iPhone or Android phones, navigation systems based on smart phones are popular. Vehicle trajectory information has become a valuable input for data forwarding. In [21], a protocol is proposed to enable efficient multihop routing capabilities. It fully supports two way communications between mobile vehicles and APs. Vehicular mobility is predicted from the information offered by the navigation system in terms of final destination and path. TSF [22] considers a reliable, efficient APs-to-vehicles data delivery by minimizing the packet delivery delay. Data delivery is performed through the computation of a target point based on the destination vehicle's trajectory information. Roadside APs can be selected as relays where destination vehicle is expected to pass by. Selecting AP optimally minimize the packet delivery delay. TBD [23] utilizes vehicle trajectory information to improve communication delay and delivery probability for vehicles-to-APs destination communications. A delay model of packets routing along roads is set up. The path with minimum delay can be found with the help of the real-time traffic condition information. VANETs can be regarded as a class of MWSN. Our work will extend the successful application of GPS trajectory in VANETs to MWSN.

It is not reliable to forward data among mobile nodes. Improving the reliability is another requirement of WMSN. Network coding has been proven to be a promising approach that can improve the reliability of communication and reduce overhead by combining packets [24]. The technique is well-suited for WSNs due to the broadcast nature of their communications. In [25], network coding is combined with multipath routing to deliver the message. The advantage of the algorithm is that the same reliability is guaranteed with significantly reduced energy consumption. In [26], an adaptive network coding is proposed to enhance reliability in WSN by considering redundancy. However, the algorithm is not suitable for the broadcast scenario. AdapCode [27] is proposed to reduce broadcast traffic in the process of code updates for the reliable data dissemination. In [28], a competitive approach to network coding is proposed to reduce the network traffic and prolong the lifetime of the network. In [29], network coding algorithm is combined with duty-cycling for saving energy in wireless sensor networks. In former researches, network coding is combined with few algorithms for MWSN. They achieve the improvement on reliability, reduce overhead, and save energy. Routing data according to the pre-trajectory can also improve the reliability. In the following work, we will combine network coding with the pre-trajectory of mobile nodes for the improvement of the scalability and reliability of MWSN.

3. Implementing TBNC on MWSN

In this section, we present the method of TBNC and its implementation on MWSN. It firstly composes navigating information and network coding in MWSN.

3.1. Conditions and Assumptions. In our work, mobile nodes have to satisfy the three restrictive conditions while using TBNC.

(1) The movements of nodes are constrained to roads. If ground node can go to any location without regard to the objects on surface, it is usually low-speed. Node's high-speed movement has a great influence on data dissemination of WSN, and it is also a characteristic of mobile WSN. The road is generically paved for the purpose of enhancing the speed of transportation. High-speed nodes on the ground usually move along the roads. Mobile nodes can go to anywhere in the specific area according to the definition of MWSN, but they only move along the roads in many applications of MWSN such as VANET, mobile IP, and passive localization. It is reasonable to constrain the high-speed nodes on roads in former applications.

(2) Mobile node installs GPS receiver for navigation. GPS became a very popular technology after being adopted for navigation. Many vehicles have installed GPS receivers. Navigation systems are also used

widely in smart phones like Apple iPhone or Android phones. Navigation system extends GPS to the usage of ordinary passenger. The interface of GPS has been reserved in the embedded systems such as ARM, PowerPC, and MIPS. New products will be developed easily to support the functions of position and navigation. It is also reasonable to navigate for mobile node using GPS.

(3) Mobile node shares its GPS navigation data to the others for reducing the delay of data dissemination. Passengers do not mind to disclose their destinations to the providers of transportation. Similarly, mobile node can share anonymously its pre-path for the higher bandwidth. The node must inform the other nodes near the path to prepare data dissemination and make sure that the navigation data is only used in communications. The information opened by the node is similar to P2P mode. The success of P2P software indicates that network users accept to share its IDs and routes to the others. User privacy can't be invaded when nodes share their pre-path. It is also reasonable to share the GPS navigation data in MWSN.

All of the former conditions are reasonable in part of applications of MWSN. In order to construct the problem model, we give the following two assumptions.

(1) All of nodes remain in a uniform linear motion. When the nodes i and j move in a same straight line with constant velocity, the packet can directly be transmit between them if their distance satisfies

$$\max\left(\left|\vec{d}_{ij}(t)\right|, \left|\vec{d}_{ij}(t+t_s)\right|\right) \le R,$$
$$\vec{d}_{ij}(t+t_s) = \vec{d}_{ij}(t) + \left(\vec{v}_i - \vec{v}_j\right) * t_s, \tag{1}$$

where R is the coverage radius of the node, $\vec{d}_{ij}(t)$ is the distance vector between nodes i and j at time t, $|\vec{d}_{ij}(t)|$ is the norm of $\vec{d}_{ij}(t)$, $\vec{v}_i$ and $\vec{v}_j$ are the velocities of nodes i and j. t_s is the time to send packet from i to j and can be computed by

$$t_s = \frac{L_p}{B}, \tag{2}$$

where L_p is the maximum length of the packets transmitting among nodes and B is the bandwidth between i and j.

(2) There are enough mobile nodes as the relay to forward data between the source and destination. We define the direction of data destination as positive and reverse direction as negative. There may be many nodes within the radio range of node. The node selects the farthest node in positive direction as its next hop. While data are forwarded completely from i to next hop, the distance between them $D(i, t + t_s)$ is computed as

$$D(i, t+t_s) = \max_{j=1}^{n_i}\left(\left|\vec{d}_{ij}(t+t_s)\right| * O(i,j)\right), \tag{3}$$

where n_i is the number of nodes within the radio range of i and $O(i, j)$ is the orientation of the node j in i. $O(i, j)$ can be denoted as

$$O(i,j) = \begin{cases} +1, & \vec{d}_{ij}(t+t_s) > 0, \\ -1, & \vec{d}_{ij}(t+t_s) < 0, \\ 0, & \min\left(\left|\vec{d}_{ij}(t+t_s)\right|, \left|\vec{d}_{ij}(t+t_s)\right|\right) > R. \end{cases} \tag{4}$$

The nodes continue to move when data are forwarded. The length of hop i can be computed by

$$l(i) = \vec{v}_i * t_s + D(i,t). \tag{5}$$

If data can arrive at destination through n hop, the distance from the source to destination L can be denoted as

$$L = \sum_{i=1}^{n} l(i). \tag{6}$$

The former two assumptions are not reasonable in any of the applications of MWSN. The following section will give the method to put away the assumptions in TBNC.

3.2. TBNC. In fact, the movement of any node is not uniform. Its velocity changes in a region. The node is only able to estimate the approximate location of neighbors. Neighbors may move out of their radio range while forwarding data. It is unreliable to forward data between two nodes moving with unpredictable velocity. TBNC forwards encoded data to multiple neighbors according to their trajectories. It improves the reliability of data dissemination using linear network coding.

3.2.1. Network Coding in the Source. Data are encoded in the source. K is its local encoding matrix with $\upsilon \times \omega$ dimension. ω is the number of rows in matrix K and also the dimension of encoding vectors. In order to reduce the complexity of decoding operation as ω is taken a fixed value. υ is the number of columns in matrix K and also the number of encoding vectors. υ can be computed by

$$\upsilon = \text{int}\left(\omega * \beta * \frac{\Delta v + 1}{Q} * 2^C\right), \tag{7}$$

where int() is the integral function, ω is the number of rows in matrix K, β is the constant adjusting υ to proper value, Δv is the velocity standard deviation of mobile nodes on the road, Q is the nodes density on this road, C is the pause frequency of data transmission, and Δv and Q are computed by the following equations:

$$\Delta v = \sqrt{\frac{1}{N} * \sum_{i=1}^{N}(v_i - \overline{v})^2},$$
$$Q = \frac{N}{(L/R)}, \tag{8}$$

where N is the number of mobile nodes and $\overline{v}$ is the average speed of mobile nodes on the road. If the distance between the node and its next hop is greater than the coverage radius R, data transmission will be paused until a node is in positive direction. C is the number of the pauses during a packet delivery to destination. The parameter C is estimated according to the detection fixed on the roadside (such as speed camera, traffic monitoring). Consider

$$C = \begin{cases} C + 1, & \text{if } \dfrac{R}{\overline{v}} < \Delta t < T_m,\ C < 5, \\ C, & \text{if } \dfrac{R}{\overline{v}} < \Delta t < T_m,\ C > 5, \\ \infty, & \text{if } \Delta t > T_m, \end{cases} \tag{9}$$

where Δt is the interval time between the two mobile nodes passing the detection. The available maximum value of C is 5. T_m is the specified maximum value of Δt. If $\Delta t > T_m$, $C = +\infty$, and $v = +\infty$, it represents that the path cannot reach the destination, and data transmission and encoding are terminated.

A row vector contains ω consecutive data blocks sent by the same source. The blocks IDs in the vector i are from $(i - 1) * \omega + 1$ to $i * \omega$, where $i = 1, 2, 3, \ldots$. All of the elements in the matrix K originate a pseudorandom sequence produced according to the vehicle ID. Seq$k$ denotes the pseudorandom sequence and is given by

$$\begin{aligned} \text{Seq}k_i &= \frac{Sk_i}{v}, \quad i = 1, 2, 3, \ldots, \\ Sk_i &= (Sk_{i-2} + Sk_{i-1}) \bmod v, \\ Sk_{-1} &= \text{ID}_{\text{source}} \bmod v, \\ Sk_0 &= \text{ID}_{\text{destination}} \bmod v, \end{aligned} \tag{10}$$

where mod is the modulus operator and ID is the unique identification of the mobile node. The element k_{ij} in the matrix K is gotten from Seq$k$ and given by

$$k_{ij} = \text{Seq}k_{i*\omega+j}. \tag{11}$$

The local coding matrix K with $v \times \omega$ dimension can be denoted as follows:

$$\begin{aligned} K &= \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1\omega} \\ k_{21} & k_{22} & \cdots & k_{2\omega} \\ \cdots & \cdots & \cdots & \cdots \\ k_{v1} & k_{v2} & \cdots & k_{v\omega} \end{bmatrix} \\ &= \begin{bmatrix} \text{Seq}k_1 & \text{Seq}k_2 & \cdots & \text{Seq}k_\omega \\ \text{Seq}k_{\omega+1} & \text{Seq}k_{\omega+2} & \cdots & \text{Seq}k_{2*\omega} \\ \cdots & \cdots & \cdots & \cdots \\ \text{Seq}k_{(v-1)*\omega+1} & \text{Seq}k_{(v-1)*\omega+2} & \cdots & \text{Seq}k_{v*\omega} \end{bmatrix}, \end{aligned} \tag{12}$$

where ω data blocks to the same destination are assembled and encoded together in the source. If x_i denotes data blocks i, ω data blocks are denoted by a data vector X:

$$X = (x_1, x_2, \ldots, x_\omega). \tag{13}$$

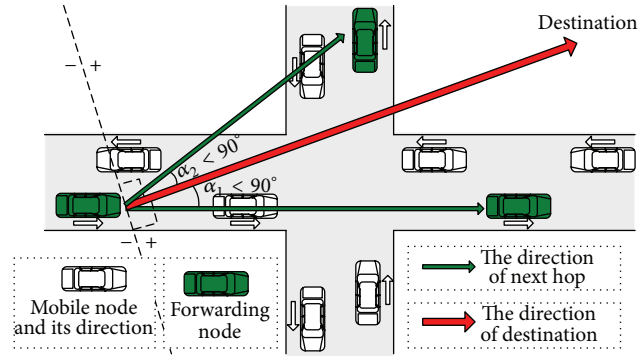


Figure 1: Forwarding data on the cross road.

Implementing linear network coding on the blocks from the source is formalized as computing the product of two matrixes. It is denoted by

$$Y = K \cdot X^T. \tag{14}$$

The encoded vector is denoted by Y, and its dimension is v. The elements in Y can be computed by

$$y_i = \sum_{j=1}^{\omega} k_{ij} * x_j = \sum_{j=1}^{\omega} \text{Seq}k_{i*\omega+j} * x_j, \tag{15}$$

where v encoded elements in Y are sent out from the source.

3.2.2. Recoding on the Cross Road. If the angle between destination and next hop directions is less than 90 degrees, the forwarding direction is positive. When data are forwarded to the mobile nodes on the cross road, there is more than one positive direction, as depicted in Figure 1. The node may select multiple next hops on different roads to forward data. It must measure the road status (such as the nodes flow and direction) to decide which nodes are next hops. The nodes flow on the road is computed by

$$f = \frac{N}{(\Delta v + 2^C) * L}. \tag{16}$$

If $f > 1/R$, the nodes on the road can deliver data to destination.

The angle α between destination and next hop directions is computed by

$$\begin{aligned} \alpha &= \left| \arccos\left(\frac{\Delta \text{la}_{\text{dst}} * \Delta \text{la}_{\text{nxt}} + \Delta \text{lo}_{\text{dst}} * \Delta \text{lo}_{\text{nxt}}}{\sqrt{\Delta \text{la}_{\text{dst}}^2 + \Delta \text{lo}_{\text{dst}}^2} * \sqrt{\Delta \text{la}_{\text{nxt}}^2 + \Delta \text{lo}_{\text{nxt}}^2}} \right) \right|, \\ \Delta \text{la}_{\text{dst}} &= \text{la}_{\text{dst}} - \text{la}_0; \quad \Delta \text{lo}_{\text{dst}} = \text{lo}_{\text{dst}} - \text{lo}_0, \\ \Delta \text{la}_{\text{nxt}} &= \text{la}_{\text{nxt}} - \text{la}_0; \quad \Delta \text{lo}_{\text{nxt}} = \text{lo}_{\text{nxt}} - \text{lo}_0, \end{aligned} \tag{17}$$

where la_0 and lo_0 are the latitude and longitude of mobile node position; la_{dst} and lo_{dst} are the destination positions; la_{nxt} and lo_{nxt} are the position of next hop.

We suppose that α_1 and α_2 are the angles between destination and next hops in the directions 1 and 2, respectively, and f_1 and f_2 are the nodes flows in the directions 1 and 2, respectively. While both α_1 and α_2 are less than 90 degrees, the next hops are selected as the following two rules:

(1) if $(\alpha_2 - \alpha_1) > 60^\circ$ or $(30^\circ \le (\alpha_2 - \alpha_1) \le 60^\circ$ and $f_1 \ge 1/R)$, data are only forwarded to the next hops in the direction 1;

(2) if $(30^\circ \le (\alpha_2 - \alpha_1) \le 60^\circ$ and $f_1 < 1/R)$ or $0^\circ \le (\alpha_2 - \alpha_1) < 30^\circ$, data are recoded and forward to the next hops in the directions 1 and 2, respectively.

The encoded vector Y is recoded with the global recoding matrix, denoted by P. The matrix P is a lower triangular matrix with $\mu \times \upsilon$ dimension, and its number of rows μ is given by

$$\mu = \text{int}\left(\upsilon * \frac{\gamma * (\alpha_1 + \alpha_2)}{(\alpha_1 f_1 + \alpha_2 f_2) * R}\right), \tag{18}$$

where γ is the constant adjusting μ to proper value and $\gamma > 2$. If the value of f_1 or f_2 is more than $1/R$, it is replaced by $1/R$ in (18).

The matrix P is unique and constant to the mobile node, need not be transferred to destinations. All of elements in P originate another pseudo-random sequence. Seq$p$ denotes the pseudo-random sequence and is given by

$$\begin{aligned} \text{Seq}p_i &= \frac{(\arcsin Sp_i)}{\pi} + 0.5, \quad i = 1, 2, 3, \ldots, \\ Sp_i &= 1 - 2 * Sp_{i-1}^2, \qquad Sp_{i-1} \in (-1, 0.5) \cup (0.5, 1), \end{aligned} \tag{19}$$

where Sp_i is an intermediate variable to produce the sequence. Sp_0 can be designated by the administrator, and its default value is equal to 0.3. $Sp_{-1} = 0$. The element p_{ij} in the matrix P is gotten from Seq$p$ and given by

$$\begin{aligned} p_{ij} &= \text{Seq}p_a; \\ a &= \begin{cases} -1 & j > i; \\ j + \sum_{k=1}^{i-1} k & j \le i. \end{cases} \end{aligned} \tag{20}$$

Implementing recoding on the coding blocks is formalized as computing the product of matrixes. It is denoted by

$$Z = P \cdot Y^T = P \cdot \left(K \cdot X^T\right)^T. \tag{21}$$

If mobile node in the range of the cross road needs forward data to multiple directions, it will buffer the encoded data, then recode them for delivery. We suppose that u is the number of buffered and recoded blocks and μ is its maximum number. u_1 and u_2 are the numbers of blocks sent to directions 1 and 2, respectively. $u = u_1 + u_2$. Pr is the proportion of the recoded blocks in two directions and can be computed by

$$\text{Pr} = \frac{u_1}{u_2} = \left(2^{(\alpha_2 - \alpha_1)/30}\right) * \frac{f_1}{f_2}. \tag{22}$$

The node clears the buffered data until it moves out of the range of the cross road.

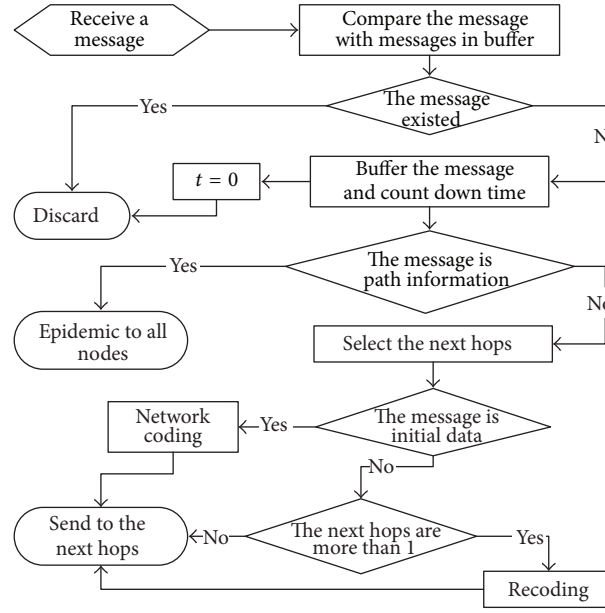


Figure 2: The process flowchart of mobile node on MWSN.

3.3. Implementing on MWSN. Mobile node presets the destination and automatically or manually pre-sets the path on GPS navigation. It shares the information of prepath to other nodes for the improvement of data dissemination. Then, the process of mobile node is formally shown in Figure 2. In this figure, initial data represent the uncoded data in the source nodes. If the number of next hops is more than 1, the mobile node is passing on the cross road and recoding the coded data for forwarding to different directions. Network coding algorithm on Figure 2 is described in Algorithm 1, and the recoding algorithm on Figure 2 is described in Algorithm 2.

Network coding is only implemented on the source node. Mobile node divides the packets into the short fixed-size blocks and generates the local coding matrix K, then encodes the blocks using linear network coding. It selects the farthest node within its radio range as the next hop in the time of a block to be forwarded. K is not sent out from the source. The destination can generate all elements of K for decoding according to source and its IDs.

The process of recoding on mobile node is described in Algorithm 2. The recoding algorithm is implemented by mobile node on the cross road, while the first element of encoded vector Y is received. Then, the mobile node is buffering the encoded blocks $y(i)$ and recoding them at the same time. The recoding matrix P is designed as a lower triangular matrix, and pseudo-random sequence is assigned to the elements whose column number is not less than row number. The recoded blocks are separated to the next hops in two directions according to the proportion of 1 : Pr. Because the matrix equations with triangular matrices are easier to solve, the decoding process is faster to implement. Destination node begins to decode initial data as long as it received ω recoded blocks.

```
Compute υ with (7)
Sk_{-1} = ID_source mod υ, Sk_0 = ID_destination mod υ
//Compute the local coding matrix K:
For i = 1 to υ
    For j = 1 to ω
        Sk = (Sk_{-1} + Sk_0) mod υ;
        Sk_{-1} = Sk_0; Sk_0 = Sk;
        k[i, j] = Sk/υ;
//Compute the encoded vector Y:
For i = 1 to υ
    sum = 0;
    For j = 1 to ω
        sum = sum + k[i, j] ∗ X[j];
    Y[i] = sum;
    Send Y[i] to the next hop
    If receiving Break Message exit.
```

Algorithm 1: Formal description of network coding on the source nodes.

```
Compute f, α_1 and α_2 with (16), (17)
Compute μ with (18)
Sp = 0.3;
//Compute the recoding matrix P:
For i = 1 to μ
    For j = 1 to υ
        If j < i then p[i, j] = 0;
        Else
            Sp = 1 − 2 ∗ Sp^2;
            p[i, j] = arcsin(Sm)/π + 0.5;
        End If
//Compute and send the recoded vector Z:
Compute Pr;
For i = 1 to μ
    For j = 1 to υ
        sum = 0;
        sum = sum + p[i, j] ∗ Y[j];
    Z[i] = sum;
    Pt = Pr/(1 + Pr);
    If i ∗ Pt − Int(i ∗ Pt) < Pt then
        Send Z[i] to the next hops in direction 1;
    Else
        Send Z[i] to the next hops in direction 2;
    End If
```

Algorithm 2: Formal description of recoding on mobile nodes.

4. Simulations

In this section, we generate the mobile nodes on the ONE simulator [30] and evaluate the performance of our TBNC method in contrast with Epidemic [31] and PRoPHET [32] routing and generic Wi-Fi.

4.1. Simulations Setting. We use part of Helsinki city map in ONE simulator as the region of simulation. The area with the range of 4000 m × 3000 m from the map is extracted as the scenarios of node movement. There are 127 cross roads within the area. The density of roads is higher than general city. The mobile nodes move on the map at the speed of 5 to 50 km/h, which covers the movement from the walk to the vehicle on the urban areas. The mobile nodes choose random from sources to destinations on the map and move there following the shortest path. The radio range is configured as 50 m, and the MAC protocol is IEEE 802.11.

We compare TBNC with Epidemic PRoPHET routing and generic Wi-Fi. Epidemic and PRoPHET are two typical routing protocols for ad-hoc networks and have been implemented by the users of the ONE simulator as add-ons. Epidemic replicates messages to all encountered nodes that do not have them yet. PRoPHET only sends messages to the

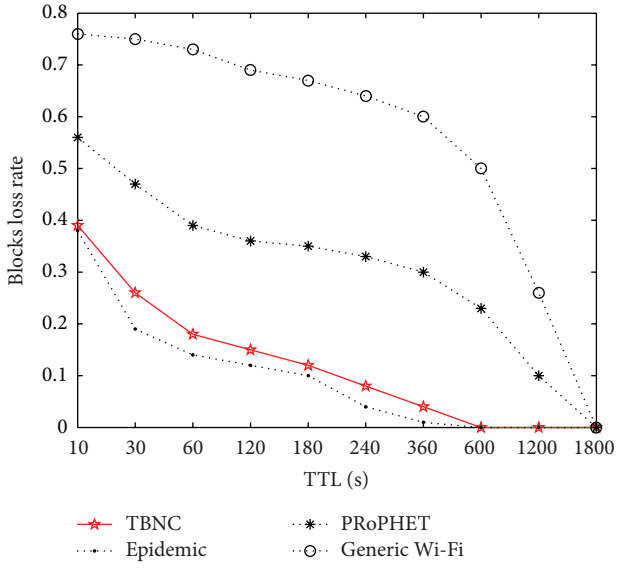


Figure 3: Blocks loss rate on the different TTLs.

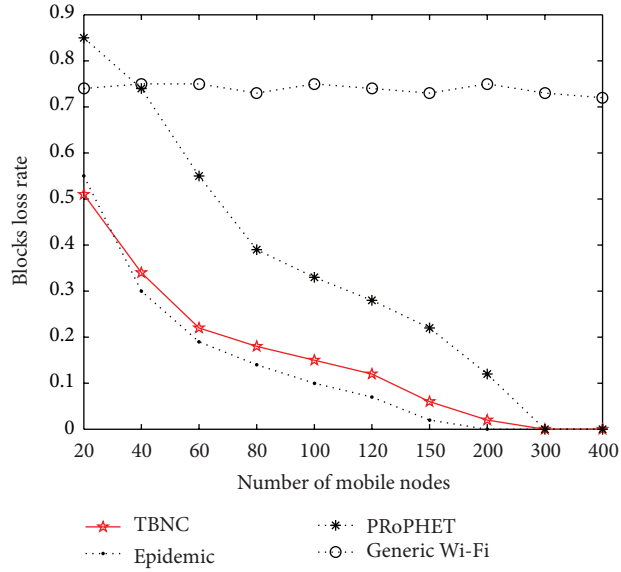


Figure 4: Blocks loss rate on the different number of mobile nodes.

nodes that have the highest chances to deliver the messages to the destination. In generic Wi-Fi, vehicles data must be delivered by access points (APs) in the radio range, and the routing capability of vehicles is disabled. Many APs need to be deployed intensively in the area.

The other parameters of TBNC method are listed as following. Maximum number of hops is 10. Maximum value of Δt is 180 seconds. The dimension of encoding vector $\omega = 10$. The number of APs in the area is 20.

We measure the block loss rate, transmission delay, and overhead messages as the parameters to reflect synthetically the performance of TBNC. The 10 sources generate flows continuously to the 10 destinations for the evaluation. The simulation lasts for 30 minutes. The number of mobile nodes and the TTL of blocks are variable at the every test, but the sources and destinations follow the respective prescribed trajectory to move in the range of test area.

4.2. Simulation Results. The block loss rate, transmission delay, and overhead messages of 10 flows are collected in the simulations. Block loss rate is the ratio of lost blocks to the sending blocks from sources. Transmission delay is the time of a block delivered from the source to all destinations. Overhead ratio is the effective blocks as the percentage of all forwarding blocks among vehicles. The average values of the 10 flows about the 3 parameters are presented as the basis for evaluating the performance of TBNC.

The average blocks loss rate of TBNC is contrasted by three protocols in Figures 3 and 4. TTLs are changed in four data plots of Figure 3, but the number of mobile nodes remains 80. Blocks loss rate decreases in all plots as the value of TTLs increases. When TTL is equal to the simulating time (1800 s), blocks loss rate is 0 for all methods. Only 20 APs cannot cover the whole area, while generic Wi-Fi has highest loss rate and cannot be used. The loss rate of PRoPHET is more than 0.3, while TTLs are less than 10 minutes. Too many blocks are lost and packets can't be recoverd in destinations. Epidemic sends copies to all encountered nodes, and transmits successively in the greatest probability, so it has lest loss rate in all methods. TBNC is close to Epidemic in blocks loss rate. When TTL is 1 minute, the blocks loss rate of TBNC is less than 0.2, and it can reach the target of transmission reliability. If network coding is not used, our method is only close to PRoPHET in blocks loss rate. Network coding played a critical role in the improvement of reliability.

The number of mobile nodes N is changed in four data plots in Figure 4, but TTL remains 60. The values of N have no effect on the loss rate of generic Wi-Fi. The other methods decrease their blocks loss rate rapidly as the mobile nodes became denser. TBNC are still closer to Epidemic than PRoPHET. When there are more than 150 nodes moving in the area, the blocks loss rate of TBNC decreases to less than 0.1. If more than 300 nodes are evenly distributed in the area, all blocks can be delivered to their destinations. The increase of N has more influence on the transmission reliability than the increase of TTL.

Figures 5 and 6 show the average delay of blocks that are transmitted successively from sources to destinations. In Figure 5, the test conditions are same as Figure 3. The average delay of all methods grows with TTLs increase. Generic Wi-Fi only can deliver few blocks to destinations. Its delay is determined by time that nodes move into the radio range of APs. PRoPHET only takes one of encountering node as the next hop. Although the node has the highest chance to destinations, it maybe moves more time to send block. So, the delay of PRoPHET is more than the other methods. TBNC is close to Epidemic in delay on the different TTLs. The process of network coding adds a little delay to delivery blocks. When TTL is 1 minute, the delay of TBNC is 7% more than Epidemic.

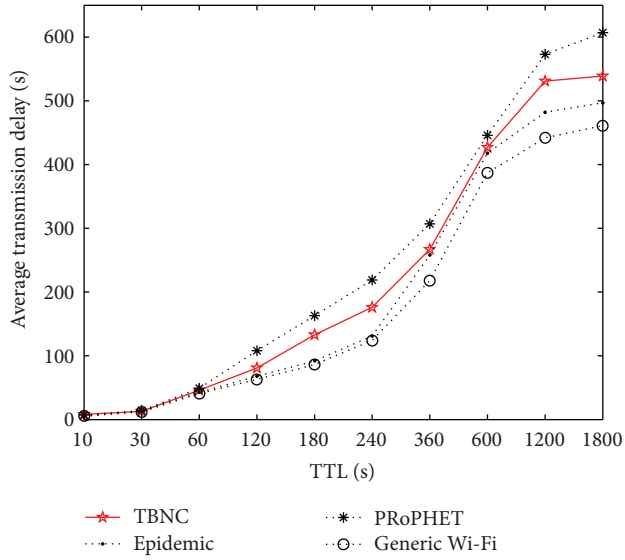


Figure 5: Average transmission delay on the different TTLs.

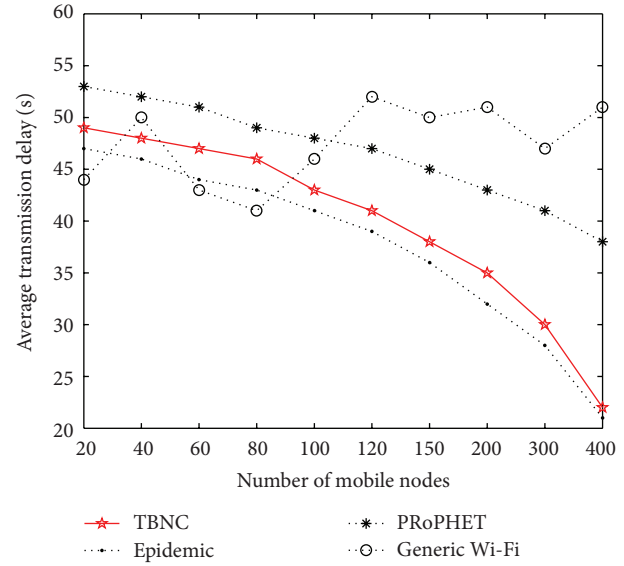


Figure 6: Transmission delay on the different number of mobile nodes.

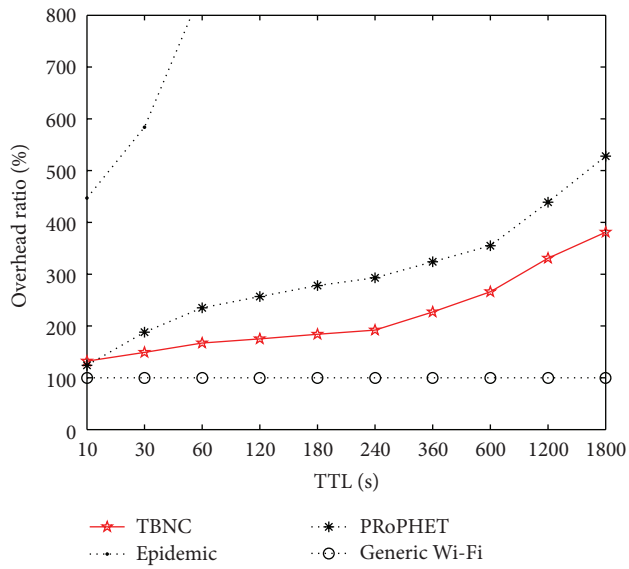


Figure 7: Overhead ratio on the different TTLs.

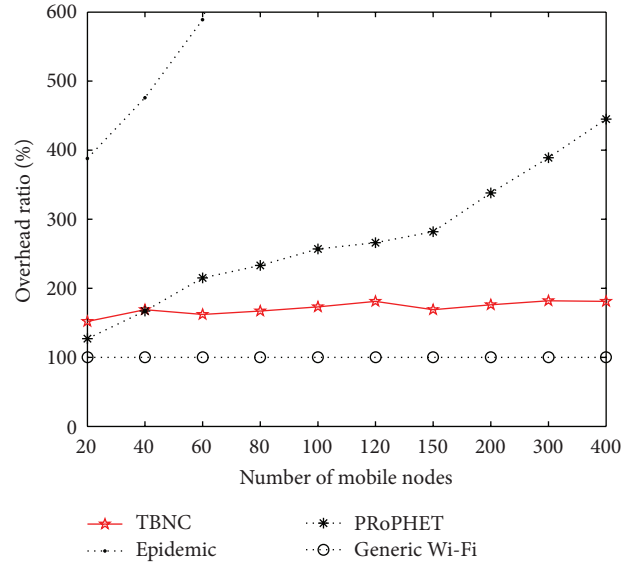


Figure 8: Overhead ratio on the different number of mobile nodes.

In Figure 6, the test conditions are same as Figure 4. The delay of generic Wi-Fi changes randomly and does not associate with the number of mobile nodes. All the other methods decrease the transmission delay with the increase of nodes. PRoPHET may select inaccurate next hops to delivery data. Its delay is more than TBNC and Epidemic. TBNC is still close to Epidemic in delay on the different N. The increase of N has also more influence on the delay than the increase of TTL. When the number of mobile nodes is more than 300, the delay of TBNC is less than 30 seconds, and it is only 7% more than Epidemic. It can reach the target of transmission real-time in the mobile networks.

The average overhead ratios of TBNC are contrasted by three protocols in Figures 7 and 8. In Figure 7, the test conditions are also same as Figure 3. Generic Wi-Fi does not generate overhead messages on any conditions, and its value is always 100% and the least in the four methods. Epidemic broadcasts data to any nodes and generates most overhead messages. Its overhead ratio rises rapidly with the increase of TTL. When TTL is 1 minute, its overhead ratio is more than 800%. Epidemic has too many overhead messages to be used in WMSN. PRoPHET produces overhead messages while it selects inaccurate next hops to delivery data. It also generates more overhead messages, while setting more TTLs. TBNC only produces overhead messages on the cross road. Its overhead messages also rise with the increase of TTL but are always less than PRoPHET. When TTL is 1 minute, TBNC has 68% less overhead ratios than PRoPHET.

In Figure 8, the test conditions are also same as Figure 4. The overhead ratio of Epidemic is too high to be used. PRoPHET grows in overhead message with the increase of N. The increase in the number of mobile nodes causes that more nodes select inaccurate next hops. Generic Wi-Fi still does not generate overhead messages. The TBNC does not change its overhead ratios with the increase of N. Overhead ratio relates to the number of cross roads rather than the number of mobile nodes. When N is 300, TBNC has 202% less overhead ratio than PRoPHET and it cuts down about 1/2 overhead messages. TBNC reduces the overhead message after network coding and improves the scalability of MWSN.

TBNC is able to achieve lower overhead ratios than both Epidemic and PRoPHET. Its blocks loss ratio and delay are tiny higher than Epidemic but lower than the other methods. TBNC reduces the overhead message after network coding and improves the scalability of MWSN.

5. Conclusion

MWSN can use the pre-trajectory of mobile nodes to improve its performance. We suppose that mobile nodes have to satisfy the three reasonable conditions. TBNC is implemented firstly on the source node. Encoded blocks are forwarded to multiple neighbors according to their trajectories. Then, they can be recoded on the mobile nodes on the cross road and delivered forwarded along multipath to the destination. TBNC combines network coding with the pre-trajectory of mobile nodes.

We evaluate the performance of our TBNC in contrast with Epidemic, PRoPHET routing, and generic Wi-Fi in the simulations. The results show that TBNC is able to cut down 1/2 overhead messages of PRoPHET when mobile nodes are shared by GPS trajectory. The blocks loss ratio and transmission delay of TBNC are tiny higher than Epidemic but lower than the other methods. Our work improves the reliability and scalability of MWSN when location and navigation systems are popular in mobile nodes.

Acknowledgments

The work is supported by the National Science Foundation of China under Grant nos. 61070169, 61202378, and 61070170, University Science Research Project of Jiangsu Province under Grant no. 11KJB520017, Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, Specialized Research Foundation for the Doctoral Program of Higher Education of China No. 20103201110018, and Application Foundation Research of Suzhou of China no. SYG201238, SYG201118, and SYG201034.

References

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

[2] S. Indu, A. Bhattacharyya, V. Kesham, and S. Chaudhury, "Self deployment of mobile sensor network for optimal coverage," *International Journal of Engineering Science and Technology*, vol. 2, no. 7, pp. 2968–2975, 2010.

[3] A. A. Ahmed, "An enhanced real-time routing protocol with load distribution for mobile wireless sensor networks," *Computer Networks*, vol. 57, pp. 1459–1473, 2013.

[4] L. Karim and N. Nasser, "Reliable location-aware routing protocol for mobile wireless sensor network," *Communications, IET*, vol. 6, no. 14, pp. 2149–2158, 2012.

[5] A. A. Ahmed and N. Fisal, "A real-time routing protocol with load distribution in wireless sensor networks," *Computer Communications*, vol. 31, no. 14, pp. 3190–3203, 2008.

[6] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.

[7] M. J. Khabbaz, C. M. Assi, and W. F. Fawaz, "Disruption-tolerant networking: a comprehensive survey on recent developments and persisting challenges," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 2, pp. 607–640, 2012.

[8] X. Cheng, A. Thaeler, G. Xue, and D. Chen, "TPS: a time-based positioning scheme for outdoor wireless sensor networks," in *Proceedings of the IEEE Conference on Computer Communications 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2685–2696, March 2004.

[9] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 81–95, September 2003.

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

[11] Y. Shang and W. Ruml, "Improved MDS-based localization," in *Proceedings of the IEEE Conference on Computer Communications 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, pp. 2640–2651, March 2004.

[12] P. N. Pathirana, N. Bulusu, A. V. Savkin, and S. Jha, "Node localization using mobile robots in delay-tolerant sensor networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 285–296, 2005.

[13] D. Koutsonikolas, S. M. Das, and Y. C. Hu, "Path planning of mobile landmarks for localization in wireless sensor networks," *Computer Communications*, vol. 30, no. 13, pp. 2577–2592, 2007.

[14] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed online localization in sensor networks using a moving target," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN '04)*, pp. 61–70, April 2004.

[15] L. Liu, R. Wang, and F. Xiao, "Topology control algorithm for underwater wireless sensor networks using GPS-free mobile sensor nodes," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1953–1963, 2012.

[16] V. K. Chaurasiya, N. Jain, and G. C. Nandi, "A novel distance estimation approach for 3D localization in wireless sensor network using multi dimensional scaling," *Information Fusion*, vol. 15, pp. 5–18, 2014.

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

[18] W. Ren, "A rapid acquisition algorithm of WSN-aided GPS location," in *Proceedings of the 2nd International Symposium on Intelligent Information Technology and Security Informatics (IITSI '09)*, pp. 42–46, Moscow, Russia, January 2009.

[19] 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, July 2001.

[20] L. Z. Li, S. K. Zhang, Y. Q. Zhu, and Z. Yang, "MCNC: data aggregation and dissemination in vehicular Ad-hoc networks using multicast network coding," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 853014, 11 pages, 2013.

[21] I. Leontiadis, P. Costa, and C. Mascolo, "Extending access point connectivity through opportunistic routing in vehicular networks," in *Proceedings of the IEEE (INFOCOM '10)*, San Diego, Calif, USA, March 2010.

[22] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "Trajectory-based statistical forwarding for multihop infrastructure-to-vehicle data delivery," *IEEE Transactions on Mobile Computing*, vol. 11, no. 10, pp. 1523–1537, 2012.

[23] J. Jeong, S. Guo, Y. Gu, T. He, and D. H. C. Du, "Trajectory-based data forwarding for light-traffic vehicular Ad Hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 743–757, 2011.

[24] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[25] L. Shan-Shan, Z. Pei-Dong, L. Xiang-Ke, C. Wei-Fang, and P. Shao-Liang, "Energy efficient multipath routing using network coding in wireless sensor networks," in *Ad-Hoc, Mobile, and Wireless Networks*, vol. 4104, pp. 114–127, Springer, Berlin, Germany, 2006.

[26] T.-G. Li, C.-C. Hsu, and C.-F. Chou, "On reliable transmission by adaptive network coding in wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, Dresden, Germany, June 2009.

[27] I.-H. Hou, Y.-E. Tsai, T. F. Abdelzaher, and I. Gupta, "AdapCode: adaptive network coding for code updates in wireless sensor networks," in *Proceedings of the 27th IEEE Communications Society Conference on Computer Communications (INFOCOM '08)*, pp. 1517–1525, Phoenix, Ariz, USA, April 2008.

[28] L. Miao, K. Djouani, A. Kurien, and G. Noel, "Network coding and competitive approach for gradient based routing in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 6, pp. 990–1008, 2012.

[29] R. Chandanala, W. Zhang, R. Stoleru, and M. Won, "On combining network coding with duty-cycling in flood-based wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 1, pp. 490–507, 2013.

[30] "The ONE," http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

[31] W. Vogels, R. Renesse, and K. Birman, "The power of epidemics: robust communication for large-scale distributed systems," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 131–135, 2003.

[32] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 505710, 8 pages
http://dx.doi.org/10.1155/2013/505710

Research Article

A Node Deployment Algorithm Based on Van Der Waals Force in Wireless Sensor Networks

Xiangyu Yu,[1] Ninghao Liu,[1] Weipeng Huang,[2] Xin Qian,[3] and Tao Zhang[4]

[1] *School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China*
[2] *China Mobile Group Corporation, Guangdong Co., Ltd., Dongguan Branch, Dongguan, Guangdong 523008, China*
[3] *Microsoft Corporation, One Microsoft Way Redmond, WA 98052, USA*
[4] *School of Electronic and Communication Engineering, Guiyang University, Guiyang 550005, China*

Correspondence should be addressed to Xin Qian; xinqian@microsoft.com

Received 7 June 2013; Accepted 4 September 2013

Academic Editor: Shukui Zhang

The effectiveness of wireless sensor networks (WSN) depends on the regional coverage provided by node deployment, which is one of the key topics in WSN. Virtual force-based algorithms (VFA) are popular approaches for this problem. In VFA, all nodes are seen as points subject to repulsive and attractive force exerted among them and can move according to the calculated force. In this paper, a sensor deployment algorithm for mobile WSN based on van der Waals force is proposed. Friction force is introduced into the equation of force, the relationship of adjacency of nodes is defined by Delaunay triangulation, and the force calculated produce acceleration for nodes to move. An evaluation metric called pair correlation function is introduced here to evaluate the uniformity of the node distribution. Simulation results and comparisons have showed that the proposed approach has higher coverage rate, more uniformity in configuration, and moderate convergence time compared to some other virtual force algorithms.

1. Introduction

Wireless sensor networks (WSN), with its advanced abilities in sensing and communication, is an emerging technology that promises a wide range of potential application in both civilian and military areas due to its low power consumption, low cost, distributed, and selforganization property. A wireless sensor network typically consists of a large number of low-cost, low-power, and multifunctional sensor nodes that are deployed in a region of interest [1, 2]. These nodes are equipped with sensors, microprocessors, and mutual communication devices, so that they have sensing ability as well as data processing and communication. WSN can be used for target tracking, temperature and environmental monitoring, security surveillance, data collection, smart homes and offices, health care, and industrial diagnosis, and so forth, thus it is an active research area of interest recently.

In the applications of WSN, energy saving, connectivity, and configuration uniformity are some of the key respects of interest, which are also related with coverage rate of the whole network. Because of this, coverage becomes an important issue in WSN. It mainly addresses how to deploy the sensor nodes to achieve sufficient coverage of the service area, so that each position in the service area is monitored at least by one sensor node.

The coverage ratio of the WSN is calculated by [3]

$$\mathrm{CR} = \frac{\bigcup c_i}{A}, \quad i \in S, \tag{1}$$

where c_i is the coverage of sensor i, S is the set of nodes, and A is the total size of the area of interest. The aim of the optimization technique is to maximize the coverage rate of the network.

A good coverage is indispensable for the effectiveness of wireless sensor networks. An efficient deployment of sensor nodes will reduce the configuration and communication cost of the network and improve the resource management, thus node deployment becomes a challenging work.

Node deployment algorithms can be divided into deterministic or movement-assisted ones. In movement-assisted deployment algorithms, each sensor knows its position; the

mobile sensors can communicate with others and can move to new positions accordingly. In some cases such as remote and unmanned environments, mobile wireless sensor networks are initially distributed randomly, and only movement-assisted deployment can be applied.

Many approaches have been proposed for movement-assisted node deployment [4, 5], such as virtual force-based [3, 6–12], swarm intelligence [13–17], and computational geometry [18], and so forth, or some combination of the above approaches [19, 20], among which, the kind of virtual force-based strategies has emerged as one of the effective solutions. In this paper, a virtual force-based node self-deployment algorithm using a force model based on van der Waals force is proposed. The relationship of adjacency of nodes is defined by Delaunay triangulation, and the force calculated produces acceleration for the nodes to move. A new metric called pair correlation is introduced to evaluate the uniformity of the node distribution. Simulation results showed that the proposed approach is better than the original virtual force algorithm in convergence time, coverage rate, and more uniformity in configuration.

The rest of this paper is organized as follows. Section 2 gives a brief introduction of virtual force-based approach. Section 3 introduces the proposed algorithm. A few simulation results are given in Section 4 to verify the effectiveness of the proposed algorithm. Finally, with several improvements discussed for our future work, we conclude the paper in Section 5.

2. Virtual Force Based Node Deployment Algorithms

Virtual force-based algorithm is a popular approach for node deployment. In this kind of algorithm, the sensor nodes, the obstacles, and the preferential areas are modeled as points subject to attractive or repulsive force among them. By setting a threshold of the desired distances among sensors, each sensor moves in accordance with the summation of the force vectors, and eventually a uniform deployment is achieved.

Some assumptions are made in the virtual force algorithm [6]: first, an individual node should be able to acquire the relative position of other nodes within its communication range; second, all the nodes can move according to the calculation results of the algorithm effectively; third, all the nodes are homogeneous with omnidirectional sensors, which means that for each node, the sensing range is identical for all nodes and the sensing areas they sensed are circles with the node at its center, so is the communication range.

Virtual force-based node deployment approach is inspired by the artificial potential field-based techniques in the field of robotic obstacle-avoidance [21, 22]. Based on disc packing and virtual potential theory, Zou and Chakrabarty designed a VFA algorithm [6], in which each node s_i is subjected to three kinds of forces: (1) a repulsive force $\mathbf{F}_{iR}$, exerted by obstacles, (2) an attractive force $\mathbf{F}_{iA}$, exerted by areas of preferential coverage (sensitive areas where a high degree of coverage is required), and (3) an attractive or repulsive force $\mathbf{F}_{ij}$, by another node s_j depending on its distance and orientation from s_i. A threshold distance d_{th} is defined between two nodes to control how close they can get to each other. The net force on a sensor s_i is the vector sum of all the above three forces:

$$\mathbf{F}_i = \mathbf{F}_{iR} + \mathbf{F}_{iA} + \sum_{j=1, j\neq i}^{k} \mathbf{F}_{ij}. \tag{2}$$

Two-sensor detection models can be applied on VFA algorithms: the binary detection model and the probabilistic detection model [6]. Then, Heo and Varshney add some restrictions on the function of force [3]. Kribi et al. improved original VFA and proposed Serialized VFA, Lmax_Serialized_VFA, and Dth_Lmax_Serialized_VFA [7]. Garetto et al. proposed a distributed sensor relocation scheme based on virtual force and made a restriction that there are at most only six nodes that can exert force on current node, and it has good coverage rate and can response to the event quickly [8]. Yu et al. introduced the idea of Delaunay triangulation to define the adjacent relation to propose a virtual force approach of better convergence time and coverage rate [9] and introduced van der Waals force into this problem [10]. An expression of exponential function for the relationship of virtual force is proposed to converge rapidly in [11]. Li et al. proposed a sensor deployment optimization strategy based on target-involved virtual force algorithm (TIVFA) [12]. Kukunuru et al. proposed a virtual force-directed particle swarm optimization algorithm, combining particle swarm optimization algorithm with virtual force to reach at a better coverage rate [13].

Once the vector of force is determined, there are various attempts to map the force to moving strategies. Some made the moving vector in the next time slot directly proportional to the calculated force or its modification [9, 11], others use the force to generate acceleration to guide motion just as the physical world [23]. In this paper, we use the original physical meaning of the force and use the force calculated to produce acceleration on nodes.

3. The Van Der Waals Force Based Node Deployment Model

The total virtual force received by a sensor node may be the composition of various types of force, such as the force resulting from interaction among the nodes themselves, the friction force hindering the motion of sensor nodes, and the force exerted by the event of interest. It is worth noting that friction plays a critical role in the performance of a VFA. In [21], potential energy and kinetic energy are mutually convertible. In order to achieve the steady state of node deployment, friction is essential for consuming the potential and kinetic energy, thus stopping the motion of whole system. In this paper, the effect of environmental events in the region of interest is not taken into consideration, and the $\mathbf{F}_i(t)$ used can be defined as the sum of two different components as follows:

$$\mathbf{F}_i(t) = \mathbf{F}_i^e(t) + \mathbf{F}_i^f(t), \tag{3}$$

where $\mathbf{F}_i^e(t)$ is the exchange forces among sensor nodes and $\mathbf{F}_i^f(t)$ denotes the friction force. In this paper, we take van der

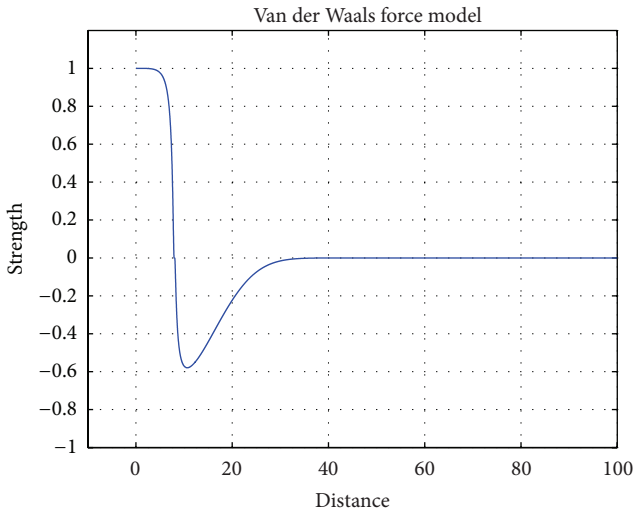


Figure 1: The van der Waals force model used in this paper.

Waals force as the exchange force. In physical chemistry, the van der Waals force is the sum of the attractive or repulsive forces between molecules (or between parts of the same molecule) other than those due to covalent bonds or to the electrostatic interaction of ions with one another or with neutral molecules [10]. The term consists of force between two permanent dipoles (Keesom force) and force between a permanent dipole and a corresponding induced dipole (Debye force), force between two instantaneously induced dipoles (London dispersion force) [24].

The van der Waals force can be modeled as

$$F(d) = \frac{A}{d^a} - \frac{B}{d^b}, \tag{4}$$

where d is the distance between two neighboring nodes and $a > b$. The force is the composition of two terms: the first term represents the repulsive force, and the second term denotes the attractive force. When $d < D_m$, $F(d)$ appears to be repulsive; when $d > D_m$, it turns out to be attractive. In (4), only one formula is used to describe both forces; the key problem here is to choose appropriate values of a and b. After simulating with various values of a and b, we found that with larger a, the repulsive force plays a dominant role, and when b is larger, the maximum value of attractive force is smaller. After repeated simulations, we tested out that $A = B = 160$, $a = 4$, and $b = 3$ are satisfactory for computer simulation. The illustration of applied force model is shown in Figure 1, from which we know that when the mutual distance is less than the horizontal ordinate of zero point (threshold), the force is repulsive. When the mutual distance is larger than the threshold value, the force turns out to be attractive.

Since van der Waals force is the interaction only between adjacent molecules, we use Delaunay triangulation [9] here to determine the "adjacent relationship". A Delaunay triangulation for a set P of points in a plane is a triangulation such that no point in P is inside the circumcircle of any triangle in it. Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation [25]. If two nodes form a side of triangle together in Delaunay triangulation, then they are defined as adjacent nodes to each other. If two close nodes are not directly connected by a triangle side in the triangulation diagram, there must be some other nodes between them, so the in-between nodes will obstruct the force between these two nodes. And if there are many nodes in the same side of the current node, only some nearest nodes are considered to exert force. In conclusion, only adjacent nodes within the communication range of each other can exert force mutually.

Friction force is indispensable for the construction equilibrium configuration and for preventing chaotic motion of sensor nodes. In real life, static friction and viscous friction are two basic forms of friction forces. For a static sensor node to move, it needs to overcome the static friction, while the motion of a moving node will also be impeded by viscous friction proportional to the instant velocity of it. During the simulation of network deployment, the friction force must be set large enough to stabilize the whole system.

Although van der Waals force has been introduced into node deployment in [10], the vector of force results in moving speed directly in the next time slot, which is inconsistent with laws of motion in the physical world. In this paper, time is divided into slot sequence, and, in each time slot dt, the motion of sensor nodes conforms to Newton's second law of motion in (5)

$$m\frac{d^2\mathbf{x}_i(t)}{dt^2} = \mathbf{F}_i(t), \tag{5}$$

where m denotes the "virtual" mass of a sensor node, $\mathbf{x}_i(t)$ denotes its position in the coordinate, and $\mathbf{F}_i(t)$ denotes the total "virtual" force exerted to a sensor node. A second-order leap-frog scheme [26] is applied here for the numerical solution of differential equation.

4. Simulation Results and Analysis

In order to verify the effectiveness of the proposed approaches, some simulations are proceeded. Binary detection model is used here and the nodes are distributed random initially. First is the simulation for a regular indoor case, which means that there is bound at each side.

To begin with, we set the length of square room as 100 and the total number of sensor nodes 213. By referring to [8], we get that

$$Dm^2 = \frac{1}{\rho_m \sin 60^\circ} = \frac{2}{\sqrt{3}\rho_m}. \tag{6}$$

So that the theoretical equilibrium distance D_m between two nodes is 7.4. However, due to the edge effect of walls constraint, the actual D_m for simulation is 8.1, which is the distance of two adjacent nodes when equilibrium state is reached. Then, the sensing range of sensor node r is $D_m/\sqrt{3} = 4.67$, and the communication range is 14.7. The virtual mass does not play an important role here, and we set it to 1 to all the nodes for convenience. Then, through repeated computer simulation tests, we finally determine the coefficient friction force as 0.3.

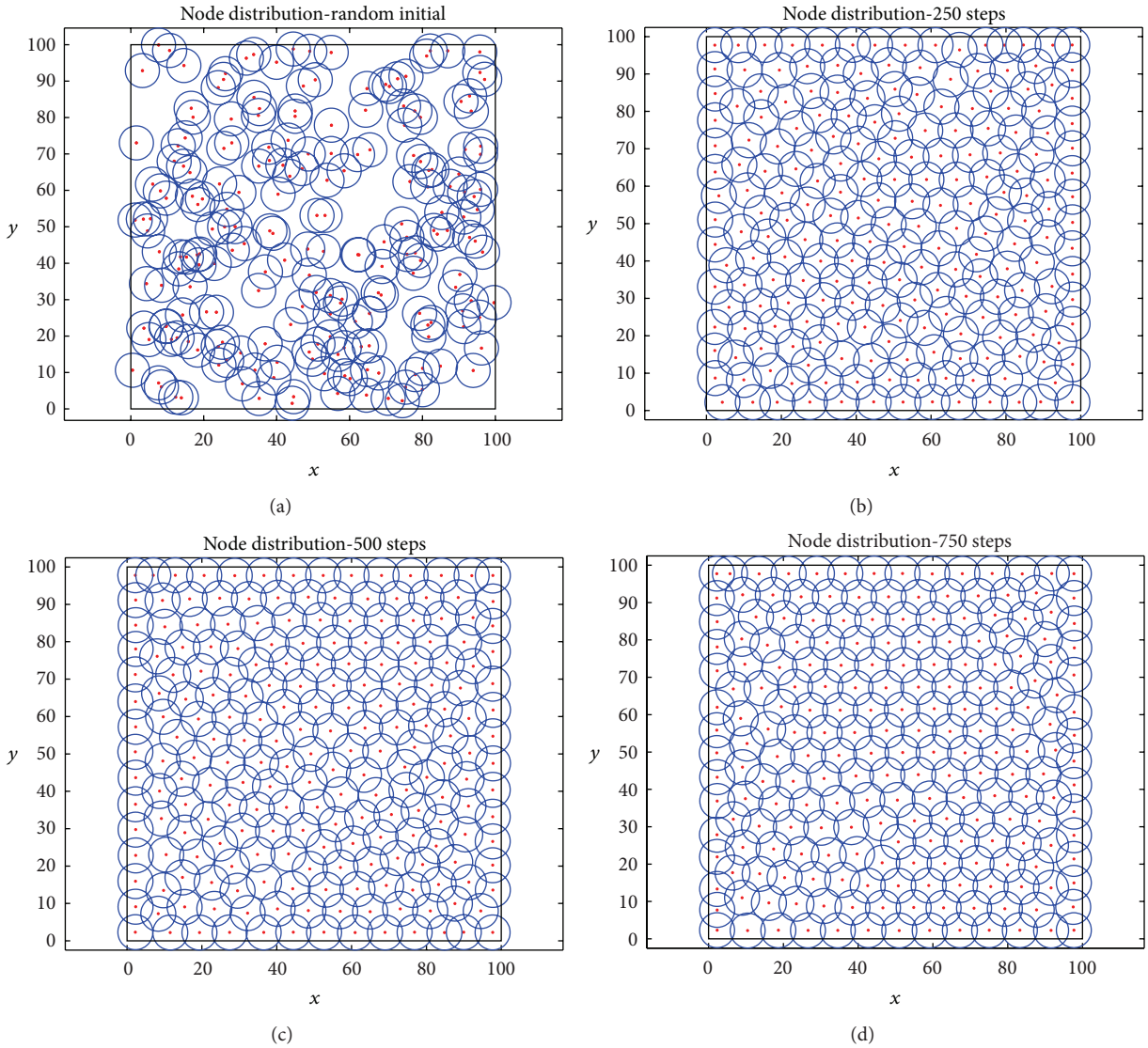


FIGURE 2: The distribution of the node at some step: (a) random initial; (b) after 250 steps; (c) after 500 steps; (d) after 750 steps.

4.1. Indoor Case. Figure 2 shows the results of node distribution at initial, 250, 500, and 750 steps, respectively. The red dots represent the positions of the nodes, and the blue circles indicate the sensing range of each node.

Figure 3 gives the comparison of the coverage rate versus the steps of the proposed approach, the original VFA, and the algorithms proposed in [8, 10]. From this figure, we can find that after the node reaches stable distribution, the proposed algorithm has better coverage rate than the other three approaches. This priority results from the feature of van der Waals force model shown in Figure 1, which can be summarized as follows.

(1) When the distance between two adjacent nodes increases over D_m, the magnitude of attractive force grows at first and then vanishes gradually to zero; when the mutual distance between two nodes decreases from D_m, the magnitude of repulsive force grows and converges to a certain value. The magnitude of force is bounded. In comparison, however, the force model of algorithm in [8] is not bounded. The magnitude of force will be infinitely larger, when the mutual distance between two nodes equals zero or R_c, in which cases of unreasonable large forces will be exerted on both nodes.

(2) An equilibrium point or threshold is indispensable in a well-performed force model. The force exerted on a sensor node at equilibrium point should be zero. Moreover, the force function should be continuous. The proposed model meets all these requirements and gives rise to the desired deployment results. However, in the original VFA, the function of force model is discontinuous at threshold point, which could explain the incompact and unstable resultant network topology.

(3) The slope of the curve in Figure 1 is large, when mutual distance falls near D_m, which insures convergence of the process of node deployment. Comparatively,

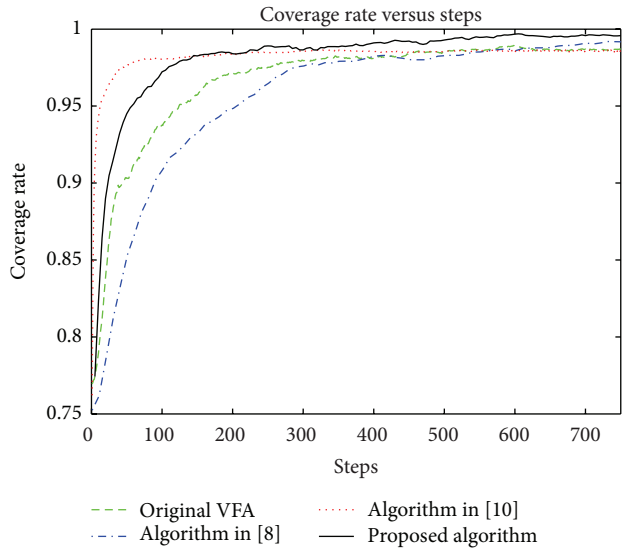


Figure 3: Comparison of the coverage rate versus steps.

the slope of force curve in algorithm [8] and original VFA is much smaller than that in the proposed algorithm, which serves as the reason accounting for the better performance of the proposed algorithm, especially in the later stage of computer simulation (steps > 400).

4.2. Outdoor Case. For indoor case, there is a bound on each side preventing sensor nodes from moving outside. In order to test the effectiveness of the proposed algorithm in more general case, another simulation for outdoor case is analyzed. All nodes are initially random, but there is no bound at each side. Figure 4 gives the initial and final distribution of the four algorithms mentioned above. The center is at (0, 0) instead to illustrate the off-centered degree effectively.

In indoor case, coverage rate is used as the metric to evaluate the performance of deployment, however, it is not effective in outdoor case anymore, since there is no boundary on any side; if we compare the coverage area under different algorithms, we can find that the one with the highest value must be the one in which all the blue circles in above figures are tangent to each other, which will leave many sensing holes and is not desired. So for outdoor case, new metric is introduced here.

Standard deviation is widely used as a metric for evaluating the uniformity of VFA. A smaller standard deviation value corresponds to a better deployment configuration. The average distance between two adjacent sensors and the standard deviation of mutual distance are listed in Table 1. From this table, we can find that the proposed algorithm is closer to perfect distribution in both average distance and standard deviation than others.

The deployment of sensor nodes under a well-performed force model should give rise to a network topology with good uniformity. Through repeated simulation, we find that a better uniformity means more resemblance to a perfect hexagon topology. Figure 5 illustrates the perfect hexagon deployment.

Table 1: Comparison of the average and standard deviation of the distance between two adjacent nodes.

Algorithm	Average distance	Standard deviation
Perfect distribution	8.1	0
Proposed algorithm	9.0859	1.3448
Algorithm in [8]	9.1232	1.7287
Original VFA	9.4950	1.7102
Algorithm in [10]	9.1104	1.5629

Table 2: Time elapse comparisons.

Algorithm	Time elapse
The proposed	237 s
Original VFA	174.26 s
Algorithm in [8]	325.39 s
Algorithm in [10]	116.74 s

An evaluation metric called pair correlation function [27], which compare the similarity of the configuration of the node distribution with the perfect hexagon topology, is also introduced here to evaluate the uniformity of the final distribution of each algorithms, which is shown in Figure 6. From this figure, we can find that the pair correlation of the proposed approach is closer to the perfect hexagon than the other algorithms, which shows that it has better uniformity than the other three methods. The "peaks" in the curve of the proposed algorithm is corresponding with the "impulses" which depict the pair correlation function of perfect hexagon. In the curves of other algorithms, the correspondences are not that obvious. The reason for the better performance of proposed algorithm is similar as what we have explained in Section 4.1.

We also compare the running time of all algorithms; the simulation environment is a PC with CPU 2.1 GHz, 8 GB RAM and all simulation are run in Matlab 2012a. The elapse time for both indoor and outdoor is almost the same, and the time elapse of each algorithm for 1000 steps is list in Table 2. From this table, we can find that the proposed approach has moderate computational complexity compared to others. The reason why the time elapse of the algorithm in [10] is the shortest is that the amplitude of force calculated is mapped into moving distance directly instead of through Newton's formula. There are no intermediate variables of speed and acceleration.

5. Conclusions

In this paper, a van der Waals force-based deployment algorithm is proposed to provide an opinion on node deployment in WSN. The relationship of adjacency of nodes is defined by Delaunay triangulation, and the force considering friction is calculated to produce acceleration for nodes to move. A new metric called pair correlation is introduced to evaluate the uniformity of the node distribution. Simulation results and

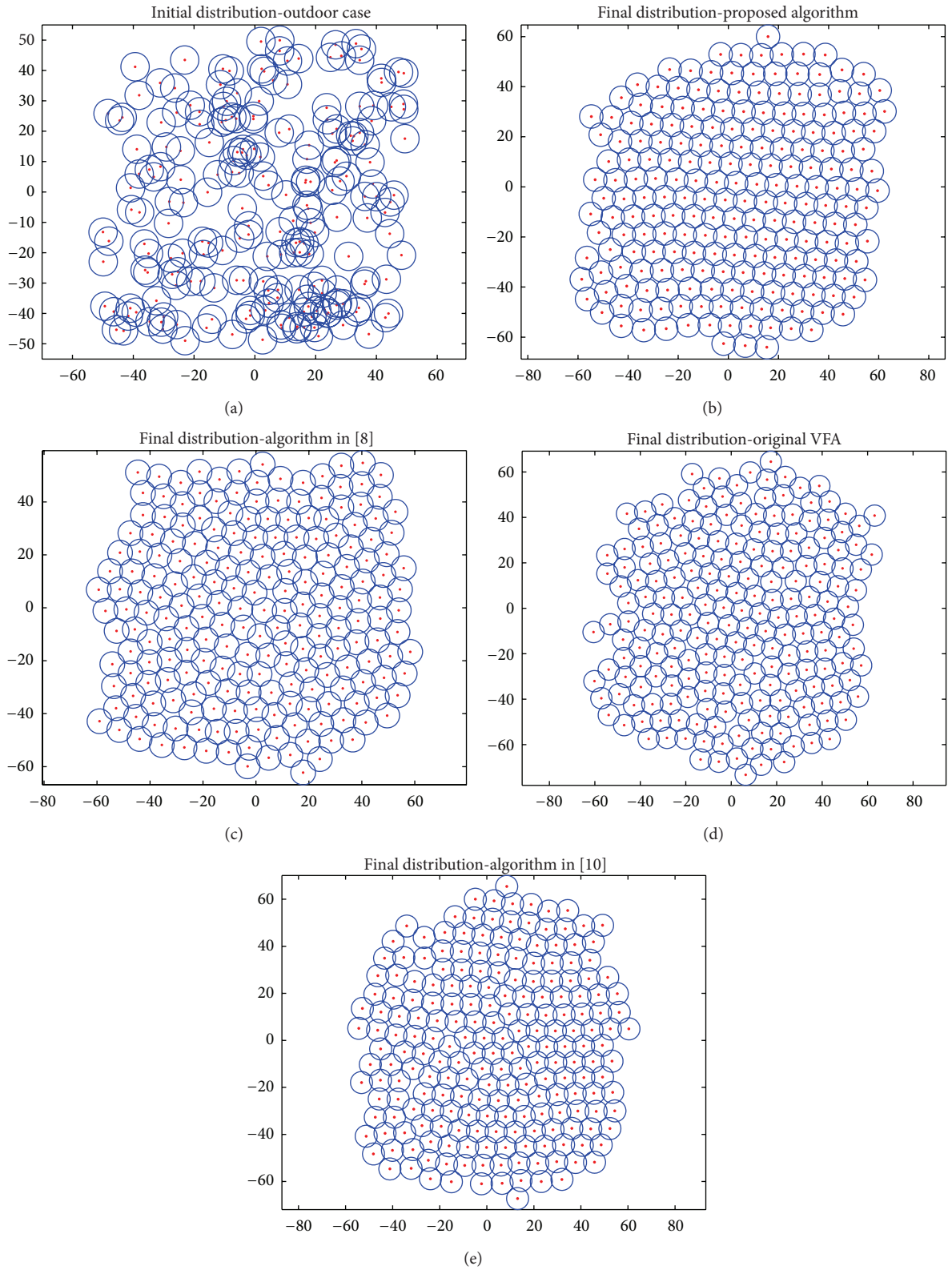


FIGURE 4: The initial random distribution (a) and the distribution of the node at step 1000 for different algorithms, (b) the proposed approach, (c) the algorithm in [8], (d) the original VFA, and (e) the algorithm in [10].

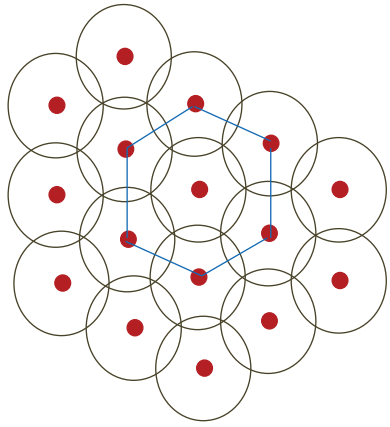


Figure 5: Perfect hexagon topology.

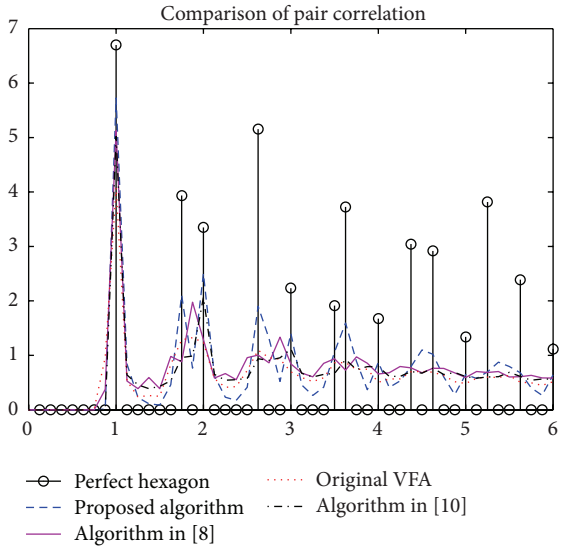


Figure 6: Comparison of the function pair correlation among different algorithms.

comparisons have verified that the proposed approach has higher coverage rate, more uniformity in configuration, and faster convergence time than traditional VFAs. Irregular terrain, obstacles, and other associated issues will be considered in our further work.

Conflict of Interests

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

Acknowledgments

This work was supported by the Foundation for Distinguished Young Talents in Higher Education Guangdong, China (Grant no. LYM10011), the National Natural Science Foundation of China (Grant no. 61201178), the Scientific Project of Guangdong (no. 2010B010600019), and, United Foundation Project of Guiyang University Guizhou (Grant no. QianKeHeJ-LKG[2013]36).

References

[1] J. Zheng and A. Jamalipour, *Wireless Sensor Networks: A Network Perspective*, IEEE Press, Piscataway, NJ, USA, 2009.

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

[3] N. Heo and P. K. Varshney, "A distributed self-spreading algorithm for mobile wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 1597–1602, New Orleans, La, USA, 2003.

[4] J. Chen, E. Shen, and Y. Sun, "The deployment algorithms in wireless sensor net works: a survey," *Information Technology Journal*, vol. 8, no. 3, pp. 293–301, 2009.

[5] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[6] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1293–1303, New York, NY, USA, April 2003.

[7] F. Kribi, P. Minet, and A. Laouiti, "Redeploying mobile wireless sensor networks with virtual forces," in *Proceedings of the 2nd IFIP Wireless Days (WD '09)*, pp. 1–6, December 2009.

[8] M. Garetto, M. Gribaudo, C.-F. Chiasserini, and E. Leonardi, "A distributed sensor relocation scheme for environmental control," in *Proceedings of the IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS '07)*, pp. 1–10, October 2007.

[9] X. Yu, W. Huang, J. Lan, and X. Qian, "A novel virtual force approach for node deployment in wireless sensor network," in *Proceedings of IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '12)*, pp. 359–363, Hangzhou, China, 2012.

[10] X. Yu, W. Huang, J. Lan, and X. Qian, "A van der Waals force-like node deployment algorithm for wireless sensor network," in *Proceedings of the 8th International Conference on Mobile Ad-hoc and Sensor Networks*, pp. 218–221, Chengdu, China, 2012.

[11] J. Chen, S. Li, and Y. Sun, "Novel deployment schemes for mobile sensor networks," *Sensors*, vol. 7, no. 11, pp. 2907–2919, 2007.

[12] S. Li, C. Xu, W. Pan, and Y. Pan, "Sensor deployment optimization for detecting maneuvering targets," in *Proceedings of the 7th International Conference on Information Fusion (FUSION '05)*, pp. 1629–1635, Philadelphia, Pa, USA, July 2005.

[13] N. Kukunuru, B. Thella, and R. Davuluri, "Sensor deployment using particle swarm optimization," *International Journal of Engineering Science and Technology*, vol. 2, no. 10, pp. 5395–5401, 2010.

[14] Z. Li and L. Lei, "Sensor node deployment in wireless sensor networks based on improved particle swarm optimization," in *Proceedings of the International Conference on Applied Superconductivity and Electromagnetic Devices (ASEMD '09)*, pp. 215–217, Chengdu, China, September 2009.

[15] W.-H. Liao, Y. Kao, and Y.-S. Li, "A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12180–12188, 2011.

[16] C. Ozturk, D. Karaboga, and B. Gorkemli, "Artificial bee colony algorithm for dynamic deployment of wireless sensor networks," *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 20, no. 2, pp. 255–262, 2012.

[17] C. Ozturk, D. Karaboga, and B. Gorkemli, "Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm," *Sensors*, vol. 11, no. 6, pp. 6056–6065, 2011.

[18] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

[19] X. Wang, S. Wang, and J.-J. Ma, "An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment," *Sensors*, vol. 7, no. 3, pp. 354–370, 2007.

[20] Y. Han, Y. Kim, W. Kim, and Y. Jeong, "An energy-efficient self-deployment with the centroid-directed virtual force in mobile sensor networks," *Simulation*, vol. 88, no. 10, pp. 1152–1165, 2012.

[21] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," in *Proceedings of the 6th International Conference on Distributed Autonomous Robotic Systems*, pp. 299–308, Fukuoka, Japan, 2002.

[22] S. Poduri and G. S. Sukhatme, "Constrained coverage for mobile sensor networks," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '04)*, pp. 165–171, Los Angeles, Calif, USA, May 2004.

[23] J. Li, B. H. Zhang, L. G. Cui, and S. C. Chai, "An extended virtual force-based approach to distributed self-deployment in mobile sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 417307, 14 pages, 2012.

[24] J. N. Israelachvili, *Intermolecular and Surface Forces*, Academic Press, 3rd edition, 2011.

[25] J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, New York, NY, USA, 2nd edition, 1998.

[26] C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulations*, McGraw-Hill Book Company, 1985.

[27] J.-J. Hua, Y.-H. Liu, M.-F. Ye, L. Wang, and Z.-H. Zhang, "Structural and dynamical analysis of a two-dimensional dusty plasma lattice," *Chinese Physics Letters*, vol. 20, no. 1, pp. 155–157, 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 141480, 9 pages
http://dx.doi.org/10.1155/2013/141480

Research Article

Spectrum Sensing for Cognitive Coexistent Heterogeneous Networks

Bingxuan Zhao and Shigenobu Sasaki

Department of Electrical and Electronic Engineering, Niigata University, 8050 Ikarashi 2-no-cho, Nishi-ku, Niigata 950-2181, Japan

Correspondence should be addressed to Bingxuan Zhao; bxzhao@ieee.org

Received 7 June 2013; Accepted 4 September 2013

Academic Editor: Hongli Xu

This paper proposes a noncoherent spectrum sensing scheme for the cognitive coexistent heterogeneous networks with the assistance of geolocation information of primary and secondary nodes. Different from the conventional networks with single secondary network, the spectrum sensing scheme in the coexistence scenario should not only detect the primary signal but also detect the secondary signals to avoid the interference with both the primary network and the operating coexistent secondary networks. Therefore, the sensing scheme in this case should be able to differentiate the primary signal from each kind of the secondary signals. However, in the coexistent heterogeneous scenarios, the secondary signals may exploit different PHY modes (some of them may be the same as the primary PHY mode), which imposes difficulties in the coherent signal detection schemes. Aiming to tackle this problem, the parallel detection of both primary signal and each kind of secondary signals is implemented through a proposed noncoherent power decomposition scheme. In this scheme, the power decomposition is formulated into a problem of solving a nonhomogeneous linear equation matrix. During the signal detection, the characteristics of both primary and secondary signals are not required. Both the analysis and the simulation results show the feasibility and efficiency of the proposed scheme.

1. Introduction

With the rapid growth of the wireless applications, more and more wireless spectrum bands are demanded. However, the wireless spectrum resources have almost been allocated to special applications. This results in the problem between the growing demand and the limited supply of the spectrum. On the other hand, measurements on different cities all over the world show that most of the allocated spectra are significantly underutilized for different locations and times [1, 2]. Cognitive radio is considered as one of the most promising techniques to provide the dynamic spectrum access for the unlicensed users, that is, secondary users (SUs), on the underutilized licensed frequency bands when the licensed users, that is, the primary users (PUs), are not using them at a specific location and a specific time period [3].

One can envision that multiple heterogeneous secondary networks will try to utilize the same spectrum holes in the licensed frequency band due to the lack of coordination. In this case, packets collision and the interference cannot be avoided. Therefore, in addition to the coexistence between the primary network and the secondary networks, the coexistence mechanism among the secondary networks should also be considered in order to prevent the performance degradation due to the harmful interference. A lot of effort has been put into the creation of such a coexistence mechanism. For example, the IEEE 802.19.1 Task Group (TG) has been created to develop high level radio-technology-independent standard for coexistence in the TV white spaces (TVWS) [4]. The open access of TVWS has motivated several standardization efforts such as IEEE 802.22, IEEE 802.11af, the European Computer Manufacturers Association (ECMA) Technical Committee 48 Task Group 1 (TC48-TG1), to develop the PHY and MAC standards to support the operation in this band. Obviously, such coexistent different standard-compliant devices lead to the heterogeneity. Heterogeneity and coexistence are the characteristics of all the open access frequency bands not unique to the TVWS. The

coexistence of heterogeneous secondary networks coupled with the PUs protection poses new and subtle challenges to the cognitive coexistent heterogeneous networks.

We can mainly classify the coexistence issues into two categories: (1) detection of available spectrum bands to protect the PUs and mitigate the interference among SUs and (2) spectrum sharing among the overlapped secondary networks to achieve better Quality of Service (QoS). The former one can be implemented by spectrum sensing or database indicating the availability of each channel. The latter one can be implemented by many coexistent mechanisms such as the Transmission Power Control (TPC) [5], Dynamic Frequency Selection (DFS) [6], Time-Division Multiple Access (TDMA), and Code-Division Multiple Access (CDMA).

The objective of this paper is to present a spectrum sensing scheme for the cognitive coexistent heterogeneous networks. Different from the traditional spectrum sensing schemes which only detect the primary signal, the proposed sensing scheme with the assistance of geolocation information is able to detect both the primary signal and the coexistent secondary signals simultaneously. The detection of primary signal refers to identifying the available spectrum to prevent causing interference to the licensed PUs. The detection of secondary signals can be used to enable optimized decisions for spectrum sharing, especially for TPC and DFS. Specifically, the proposed scheme has the following features.

(1) It can differentiate the primary signal and each kind of secondary signals individually. Therefore, such signals can be detected simultaneously.

(2) The strength of the secondary signals operating in the adjacent channels can be obtained so that they can be subtracted from the total collected energy when constructing the test statistic at each sensor. As a result, the noise uncertainty range can be narrowed down.

(3) The proposed scheme avoids the widely used *quiet periods* in traditional energy detection schemes. This can lower the synchronization requirement in the coexistent networks and improve the QoS of the secondary networks.

(4) It is a noncoherent scheme and does not require any prior knowledge of both primary and secondary signals.

The reminder of this paper is organized as follows. In Section 2, we will analyze the problem of spectrum availability detection. In Section 3, we will present the system model. In Section 4, we will analyze the noise uncertainty problem in the cognitive coexistent networks. In Section 5, we will describe the proposed spectrum sensing scheme for cognitive coexistent heterogeneous networks. The simulation-based performance evaluation will be illustrated in Section 6. We conclude this paper in Section 7.

2. Spectrum Availability Detection

In the cognitive coexistent heterogeneous networks (a typical coexistent scenario, considering multiple secondary networks, is shown in Figure 1), the spectrum availability detection is twofold: first, the primary signals should be detected to protect the incumbent Pus; second, the secondary signals on the primary band should also be detected in order to enable optimized decisions when selecting operating channels, for example, using DFS to avoid operating in the same channel in the interference range or using the TPC to restrain the transmission power to decrease the interference when operating in the same channel.

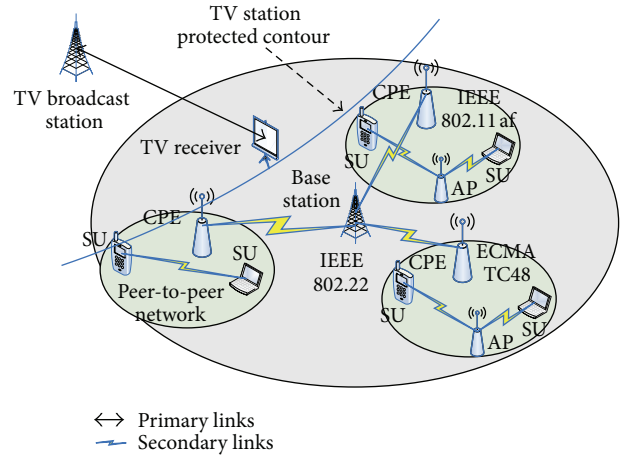


Figure 1: A typical cognitive coexistent heterogeneous scenario.

2.1. Primary Signal Detection. Since the precondition of using dynamic spectrum access is to protect licensed PUs from interference, the detection of primary signals should be a very strict requirement. This can be seen from an example of the FCC rule. According to the rule, the Advanced Television Standard Committee (ATSC) signal higher than −114 dBm over a 6 MHz frequency band should be detected with a probability of detection (P_D) higher than 90% as well as a probability of false alarm (P_{FA}) lower than 10%, and the signal of the part 74 devices (e.g., wireless microphone) higher than −107 dBm over 200 kHz frequency band should be detected with the same P_D and P_{FA}. This is a very challenging requirement. In order to detect the primary signal under the challenging constraint, both database approach and different kinds of spectrum sensing schemes have been proposed in standards and the literature.

Both of these two approaches have their pros and cons. Compared with the geolocation-based database which is more reliable but sometimes cannot be reachable, the spectrum sensing is more flexible but faces difficulties in detecting signals in such a low level mentioned above. Sensing techniques of primary signals to date can mainly be classified into noncoherent detection and coherent detection. Compared with the former one, coherent detection, for example, matched filter detection [7] and cyclostationarity detection [8], requires the prior knowledge of the primary signals such as the operating frequency, pulse shape, bandwidth, and modulation type and order. Energy detection is a kind of noncoherent detection. It detects a signal by collecting the power samples in an observing period and compares it

with a predefined threshold based on the noise floor. Energy detection is widely used, especially in cooperative spectrum sensing, because of its simplicity and no requirement of any prior knowledge of the primary signal. However, the performance of the energy detection is significantly compromised by the noise uncertainty including the device noise uncertainty and the environmental interference uncertainty. As it is pointed out in [9], the environmental interference dominates the noise uncertainty. Therefore, when performing the energy detection, the environmental interference should be deducted from the test statistic so as to greatly narrow down the uncertainty range.

2.2. Secondary Signal Detection. In the scenario of multiple secondary networks coexisting together (see Figure 1), secondary signals operating in the cochannel and adjacent channels should also be detected. While detection of cochannel secondary signals can be used for DFS and TPC, detection of the signals operating in adjacent channels can be exploited to narrow down the noise uncertainty range, which significantly affects the sensing performance. Similar to the case of primary signal detection, the detection of secondary signals can also be implemented by either the database or the spectrum sensing. However, since the white spaces of the primary band are opened to access, it is almost impossible to use the centralized database to control all the accesses of the secondary networks. For example, it is difficult to control the noncompliant networks (e.g., the peer-to-peer network) because they do not obey a standard access mechanism. In this case, the spectrum sensing becomes necessary no matter whether the secondary database exists or not.

Due to the heterogeneity of the secondary networks, the transmission power, signal characteristics, and the used protocols may be different. For example, the transmission Equivalent Isotropically Radiated Power (EIRP) of the fixed and the mode II devices in the TVWS can be as high as 36 dBm, while the maximum transmission EIRP of the mode I devices is 20 dBm. Owing to the difference of the air interfaces and protocols, it is difficult to use the coherent detection to differentiate the primary signals and each kind of secondary signals. Another challenging issue in cognitive heterogeneous networks is the synchronization. It is the precondition of the traditional sensing schemes using quiet periods during which all the SUs should stop data transmission to decrease the interference. Therefore, the detection of both primary signal and secondary signals simultaneously is much more challenging than solely detecting the primary signal. On the other hand, failure to detect or ineffectively detecting the secondary signals may lead to the following problems: (1) performance degradation may be resulted from the interference within the overlapping regions; (2) missed detection may lead to packet loss and impact the communication effectiveness among the secondary networks: for example, the loss of Coexistence Beacon Protocol (CBP) due to the packet collision will prevent the convergence of the self-coexistent IEEE 802.22 networks [10]; (3) loss of synchronization beacons will cause difficulties in scheduling *quiet periods*, which then leads to performance degradation of traditional sensing schemes.

Analyses above show that spectrum sensing for coexistent heterogeneous networks has to solve two challenging tasks at the same time: primary signal detection and secondary signal detection. Most of proposed spectrum sensing techniques in the literature so far focus only on the primary signal detection exploiting the *quiet period*; that is, all the coexistent secondary networks should stop data transmission in this period [10–12]. In such techniques, the scheduling of *quiet period* requires tight synchronization of the secondary networks, which can be easily implemented by coordination within the identical type of secondary network. However, it will become much challenging when different types of secondary networks coexist together. In addition, the previously proposed coherent spectrum sensing techniques cannot work well when the primary network takes the same PHY mode as one or more secondary networks. Aiming at such challenges and characteristics of the cognitive coexistent heterogeneous networks, this paper proposes an energy detection scheme to detect the primary signal and each kind of secondary signals simultaneously. In addition, we try to narrow down the noise uncertainty range by mitigating the environmental interference to improve the sensing performance.

3. System Model

In this section, we will create a model to describe the cognitive coexistent heterogeneous networks. To simplify the description, as it is shown in Figure 1, we consider a coexistent network that consists of a primary transmitter and multiple secondary networks such as the IEEE 802.22 network, IEEE 802.11af network, and ECMA TC48 network. The primary spectrum band is divided into M channels. We assume that the primary transmitter, the fusion center (FC), and the K sensors $(S_1, \ldots, S_K)$ are working on channel j, $j \in (1, \ldots, m)$. In channel i, $i = 1, \ldots, m$, there are c_i SUs, $(\mathrm{SU}_{i,1}, \ldots, \mathrm{SU}_{i,c_i})$. The SUs operating on the adjacent channels except channel j are used to model the adjacent channel interference (ACI) to the PUs, while the SUs operating on channel j are the source of cochannel interference (CCI) for primary detection. Noticing the mandatory requirement of geographical information by FCC and Ofcom, we assume that the geographical locations of all the nodes including the PUs, SUs, and the sensors are known so that the distances among them can be obtained. In addition, we assume that the device noise power is varying but identical for each sensor and the exchanging of control messages is free of error similar to [13].

The detection of primary signal and each kind of secondary signals can be formulated into binary hypothesis testing problems. For each kind of signal, we use H_0 and H_1 to denote the absence and presence of the signal, respectively. The probability of false alarm $P_{\mathrm{FA}} = \mathrm{prob}\{E_t > S_t \mid H_0\}$ and the probability of missed detection $P_{\mathrm{MD}} = \mathrm{prob}\{E_t < S_t \mid H_1\}$ are used to evaluate the sensing performance, where E_t and S_t

are the test statistic and the predefined threshold, respectively. Under the *Neyman-Pearson* criterion, S_t can be obtained by

$$S_t = \frac{\sigma_n^2}{\sqrt{N_s}}\left(1 + Q^{-1}\left(P_{\mathrm{FA}}\right)\right), \tag{1}$$

where σ_n^2 and N_s are the device noise power and number of samples in the sensing period, respectively [14]. $Q(x)$ can be given by

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{+\infty} \exp\left(\frac{t^2}{2}\right) dt. \tag{2}$$

The process of data exchanging in the proposed sensing scheme can be described as follows.

(1) Each sensor performs local energy collection and reports the collected energy to the FC.

(2) The FC decomposes the total received power into the primary signal power, each kind of secondary signal powers, and the device noise power. According to either decision fusion or data fusion [14], the FC exploits the decomposed primary signal power and the secondary signal powers to make final decision whether the primary signal and each kind of secondary signals are present or not. In addition, the received power at each SU, produced by each of other SUs, can also be computed by the FC for network optimization. The detailed scheme of this step will be described in Section 5.

(3) The FC distributes the computed secondary signal powers and the decisions on the primary signal and secondary signals to the related SUs.

4. Noise Uncertainty Problem

Many energy detection schemes assume the noise power is known a priori in order to form the test statistic and/or the decision threshold [14]. In fact, its power level varies over time and is very difficult to be exactly measured, which yields the noise uncertainty problem. Mathematically, if we use σ_{act}^2, σ_n^2, and α to denote the actual noise power, nominal noise power, and the uncertainty factor in dB, respectively, the actual noise uncertainty zone can then be given by

$$\sigma_{\mathrm{act}}^2 \in \left[\frac{\sigma_n^2}{\beta}, \beta\sigma_n^2\right], \quad \beta = 10^{\alpha/10}. \tag{3}$$

Energy detection employs the summation of the received target signal and the actual noise to construct the test statistic to compare with a given threshold. Obviously, if the target signal is so weak that the test statistic falls into the noise uncertainty zone, it is impossible to infer the presence or absence of the signal. The wider the noise uncertainty zone is, the worse the detection performance is. Therefore, in order to improve the performance of energy detection, we can try to narrow down the noise uncertainty zone. Generally, the noise consists of the device noise and the interference including CCI and ACI. According to [9], the device noise uncertainty is generally less than 2 dB, while the uncertainty of ACI can be as wide as 58 dB. Obviously, the uncertainty of CCI would be much wider than ACI when the transmission power is the same. Therefore, it is considerably important to differentiate the primary signal, each kind of secondary signal in the sensing channel, the ACI from the SUs in the adjacent channels, and the device noise from total received energy. When constructing the test statistic for the detection of primary and secondary signals, the interference and the device noise should be cancelled so as to improve the energy detection performance.

5. Parallel Detection of Primary and Secondary Signals

In this section, we present how to detect the primary signal and secondary signals in parallel. Then, the presence or absence of the primary signal and secondary signals as well as the power level of the secondary signals at each SU can be determined. In addition, the ACI can also be mitigated in signal detection after the power decomposition.

According to [15], when the two-ray ground propagation model is used, the interference power with the receiver rx, operating on channel v produced by the transmitter tx, operating on channel u, can be given by

$$P_I(u, v) = d_{tx,rx}^{-\beta} P_{tx} I(u, v)\, \phi, \tag{4}$$

where $d_{tx,rx}$ is the distance between tx and rx; P_{tx} is the transmission power of tx; ϕ is constant and related to the antenna gains and heights of the transmitter and receiver; β is the path loss exponent and is typically between 2 and 4; $I(u, v)$ is the interference factor and can be obtained through the power mask requirement [15, 16]. $I(u, v)$ is not required when performing the power decomposition scheme, although it is utilized in the derivation process. Obviously, $I(u, v) = 1$ when tx and rx are operating on the same channel.

Equation (4) illustrates that each primary and secondary transmitter contributes a part of power to the energy collecting sensor. As a result, the received power at each sensor is a composite power consisting of several parts: the primary signal power, the cochannel secondary signal powers, and the adjacent channel secondary signal powers. From the perspective of primary signal detection, both cochannel and adjacent channel secondary power are interference and should be removed when constructing the test statistic of energy detection. When the primary signal is detected, all the secondary networks should vacate the sensing channel and switch to other backup channels or terminate the data transmission when there are no available backup channels to protect the PUs. When the primary signal is not detected, the sensed channel can be utilized by the secondary networks. In this case, the coexistence mechanism becomes greatly important to avoid performance degradation because of the interference. To enhance the interference management, it is important to detect secondary signal and to determine the power level of each kind of secondary signals.

Obviously, the composite received power of a sensor S_k operating on channel j can be expressed by

$$P_r[S_k] = P_{pr,r}[S_k] + P_{se,r}[S_k] + P_{se,\text{int}}[S_k] + \sigma_n^2, \quad (5)$$

where $P_{pr,r}[S_k]$ and $P_{se,r}[S_k]$ are the received power of S_k produced by the target primary signal and the secondary signals operating in the channel j, respectively. $P_{se,\text{int}}[S_k]$ is the caused interference power at S_k by the SUs operating in the adjacent channels. σ_n^2 is the device noise power. According to (4), $P_{pr,r}[S_k]$, $P_{se,r}[S_k]$, and $P_{se,\text{int}}[S_k]$ can be expressed by respectively,

$$P_{pr,r}[S_k] = \phi d_{tx,S_k}^{-\beta} I(j,j) P_{tx}, \quad (6)$$

$$P_{se,r}[S_k] = \sum_{l=1}^{c_j} \phi d_{\text{SU}_{j,l},S_k}^{-\beta} I(j,j) P_{\text{SU}_{j,l}}, \quad (7)$$

$$P_{se,\text{int}}[S_k] = \sum_{i=1,i\neq j}^{M} \phi I(i,j) \sum_{l=1}^{c_i} d_{\text{SU}_{i,l},S_k}^{-\beta} P_{\text{SU}_{i,l}}. \quad (8)$$

In (6)–(8), P_{tx} and $P_{\text{SU}_{i,l}}$ are the transmission power of the primary transmitter and the lth SU operating in the channel i (i.e., $\text{SU}_{i,l}$), respectively; d_{tx,S_k} and $d_{\text{SU}_{i,l},S_k}$ are the distances from the sensor S_k to the primary transmitter and the $\text{SU}_{i,l}$, respectively.

Let

$$D(S_k) = \left(d_{tx,S_k}^{-\beta}, d_{se}(1), \ldots, d_{se}(M)\right),$$
$$P = \left(\phi P_{tx}, P_{se}(1), \ldots, P_{se}(M)\right), \quad (9)$$

where

$$d_{se}(i) = \left(d_{\text{SU}_{i,1}}^{-\beta}, \ldots, d_{\text{SU}_{i,c_i}}^{-\beta}\right),$$
$$P_{se}(i) = \phi I(i,j)\left(P_{tx}, P_{\text{SU}_{i,1}}, \ldots, P_{\text{SU}_{i,c_i}}\right), \quad (10)$$
$$i = 1, \ldots, M;$$

then, (5) can be rewritten as (11) by noticing the fact that dimensions of the vectors $D(S_k)$ and P are the same:

$$D(S_k) \cdot P^\sharp = P_r(S_k) - \sigma_n^2, \quad (11)$$

where $(\cdot)$ and $(\,)^\sharp$ are the matrix multiplication operator and the transpose operator, respectively.

For all the other sensors, we can obtain the same results as the sensor S_k. By combining them together using the matrix equation, we can obtain

$$D \cdot P^\sharp = P_r - \sigma_n^2 \mathbf{e}, \quad (12)$$

where

$$D = [D(S_1), \ldots, D(S_K)]^\sharp,$$
$$P_r = [P_r(S_1), \ldots, P_r(S_K)]^\sharp, \quad (13)$$
$$\mathbf{e} = (1, \ldots, 1)^\sharp.$$

Obviously, D is a $K \times T$ matrix, where $T = \sum_{i=1}^{M} c_i + 1$. $\mathbf{e}$ is a column vector with a size of K.

By finely adjusting the geolocations of the sensors, the rank of coefficient matrix D can always satisfy $\text{rank}(D) = \min(K, T)$. In order to solve the vector P and σ_n^2 from (12), K should satisfy the condition: $K \geq T' = T + 1$. When $K = T'$, if we use σ_0^2 to denote the solution of the device noise power, the solution of the vector P can be obtained by $P^\sharp = D^{-1}(P_r - \sigma_0^2 \mathbf{e})$. When $K > T'$, any T' sensors can be selected to construct a new coefficient matrix D' and a new vector of received power P_r' as long as both the rows and columns of D' are linearly independent, and the solution set can be obtained similarly from $P^\sharp = D'^{-1}(P_r' - \sigma_0'^2 \mathbf{e})$.

The solved vector P can be used to obtain the received primary signal power, each kind of secondary signal powers in the sensing channel, and the ACI. The received primary signal power at the sensor S_k and the ACI can be expressed by respectively,

$$P_{pr,r}(S_k) = d_{tx,S_k}^{-\beta} P(1), \quad (14)$$

$$P_{se,\text{int}}(S_k) = \sum_{i=1,i\neq j}^{M} \sum_{l=1}^{c_i} d_{\text{SU}_{i,l},S_k}^{-\beta} P(l'), \quad (15)$$

where

$$l' = \sum_{\omega=1}^{i-1} c_\omega + l + 1, \quad 1 < i \leq M,\ c_0 = 0. \quad (16)$$

Next, we should consider the secondary signals. If we assume there are Θ kinds of secondary heterogeneous networks, each of which consists of h_θ SUs, where $\theta = 1, \ldots, \Theta$ and $\sum_{\theta=1}^{\Theta} h_\theta = c_j$, the received signal power at the sensor S_k produced by the θth kind of secondary network can be expressed by

$$P_{se,\theta}(S_k) = \sum_{t=1}^{h_\theta} d_{\text{SU}_{j,t'},S_k}^{-\beta} P(t'), \quad (17)$$

where

$$t' = \sum_{\omega=1}^{j-1} c_\omega + \sum_{\tau=1}^{\theta-1} h_\tau + t + 1, \quad h_0 = 0. \quad (18)$$

Up to now, we have obtained the device noise power σ_0^2, the received primary signal power, each kind of secondary signal powers, and the ACI. The device noise power, the given P_{FA}, and sample number N_s can be used to determine the threshold for local energy detection in the coexistent network by using (1). Then, the primary signal detection by each local sensor can be implemented by comparing the obtained primary signal power from (14) with the threshold. After that, the FC can make the final decision according to different decision fusion rules such as *and*, *or*, or *majority* [17]. Similarly, the detection of each secondary signal can also be obtained by using the same device noise power and its designated P_{FA}. Note that the ACI can be excluded in

the detection of both primary signal and secondary signals in the proposed scheme. As a result, the noise uncertainty range can be greatly narrowed down [9]; thus, the detection performance can be further improved.

In coexistent heterogeneous networks, the detection of the secondary signals is not enough. In some cases, the signal power level is also necessary in order to optimize the spectrum sharing, for example, TPC. In the proposed scheme, the caused CCI power to each SU by other SUs sharing the same channel can also be achieved. For example, the CCI power at the θ^*th SU, SU_{j,θ^*}, of the θth kind of secondary network can be obtained by

$$P_{cci}\left(\mathrm{SU}_{j,\theta^*}\right) = d_{tx,\mathrm{SU}_{j,\theta^*}}^{-\beta} P(1) + \sum_{t=1, t\neq\theta^*}^{h_\theta} d_{\mathrm{SU}_{j,t'},\mathrm{SU}_{j,\theta^*}}^{-\beta} P\left(t'\right). \tag{19}$$

Such information is beneficial for the SUs with given tolerant interference. When the total interference exceeds the tolerant value, the coexistent SUs can be requested to decrease their transmission power to achieve better coexistent performance.

As we can see, the proposed parallel detection scheme is actually a real-time spectrum sensing scheme. Since the received signal power varies with the variation of the transmission power of both PU and SUs, the proposed scheme is applicable in the adaptive transmission power control. Moreover, it can be used in the scenario with ON/OFF status of the secondary signals, in which the OFF status is treated as the absence of this secondary signal by noticing the fact that the actual received power at each sensor does not contain the resulted part from this secondary signal.

Due to the successful avoidance of the widely used quiet periods in the conventional energy detections, the proposed parallel sensing scheme has at least two advantages. First, it lowers the synchronization requirement, which is difficult to be implemented in the coexistent heterogeneous networks. Second, it can improve the QoS (e.g., the capacity and the packet delay) of the secondary networks because the secondary transmissions do not need to stop during the spectrum sensing. In addition, no prior knowledge of both primary signal and the different kinds of secondary signals is required, which avoids the difficulties in acquiring such information due to the heterogeneity.

6. Performance Evaluation

6.1. Simulation Results. After the power decomposition, as we can see from Section 5, the detection processes of primary signal and secondary signals are essentially the same except using different P_{FA} to determine the corresponding decision thresholds. Therefore, when evaluating the detection performance, we do not make differentiation between the primary signal and the secondary signals, instead, we use *signal* to stand for both of them (i.e., it can be either primary signal or secondary signal depending on the application scenario). Since few similar spectrum sensing techniques with the proposed one can be found in the literature, we evaluate our scheme by comparing its achieved performance with the FCC requirements.

In the simulation, the primary spectrum band is divided into 3 consecutive channels (i.e., ch1, ch2, ch3) with 6 MHz bandwidth, which is the TV channel bandwidth in US and many other countries [18]. The average power spectral density (PSD) of the device noise is set to be −174 dBm/Hz, and no noise figure is considered for each sensor. In each channel, we deploy 2 SUs. The primary transmitter, the FC, and the sensors are deployed in ch2. As it is analyzed in Section 5, the number of deployed sensors should equal the sum of the SUs and PU; that is, 7 sensors are deployed in ch2. The geographic locations of all the nodes including PU, SUs, and sensors are randomly generated so as to average the achieved results. The path loss exponent β is set to be 3. In order to effectively show the role of the proposed scheme, we do not consider the diversity gain in the simulation. The *or* rule is used for decision fusion in the FC.

Figure 2 shows the achieved P_{MD} with a given $P_{\mathrm{FA}} = 10\%$ for different ACI caused by the SUs operating in the adjacent channels. We take 1200 samples (corresponding to 200 μs) as the sensing duration. It shows that the detection performance degrades very fast with the increase of the ACI. By using the proposed scheme, after the ACI is decomposed and removed from the test statistic (corresponding to the case of ACI = 0 dB in the figure), the detection performance becomes much better. In addition, Figure 2 illustrates the P_{MD} can be lower than 7% when the target primary signal is −114 dBm over the 6 MHz channel and the ACI equals 3 dB; that is, the probability of detection is higher than 93% in this case. Therefore, the proposed scheme can achieve 3 dB margin for adverse factors (e.g., fading) when satisfying the sensing requirement of FCC.

Figure 3 indicates the sensing performance for different sensing durations. The P_{FA} is also set to be 10%, and no ACI is used. As it is shown in this figure, the performance of the proposed scheme can be improved by increasing the sensing time. In other words, longer sensing time can be used to detect weaker target signal. Obviously, it leads to higher sensing complexity. Therefore, there exists a tradeoff between the sensing performance and the sensing duration.

Figure 4 evaluates the detection performance with different device noise uncertainties. P_{FA} and N_s are set to be 10% and 1200 (200 μs), respectively. Although the proposed scheme is able to mitigate the CCI and ACI, Figure 4 shows that it still suffers from the device noise uncertainty. This figure implies that the performance of the proposed scheme degrades with the widening uncertainty range. Fortunately, compared with the environmental interference, the uncertainty range of device noise is much narrower and is usually less than 2 dB [9]. In addition, such a performance degradation can be compensated by the left interference margin and sensing time to some extent.

6.2. Sensitivity Evaluation. As it is shown in Section 5, the performance of the proposed scheme depends on the accuracy of the path loss exponent and the distances among

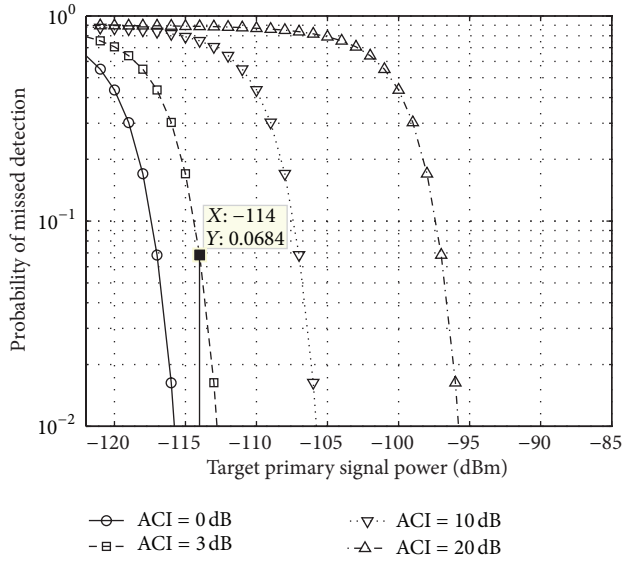


Figure 2: Probability of missed detection for different ACI. P_{FA} and N_s are set to be 10% and 1200 (corresponding to 200 μs), respectively. The ACI values in this figure are the relative value to the average device noise power. No device noise uncertainty is considered.

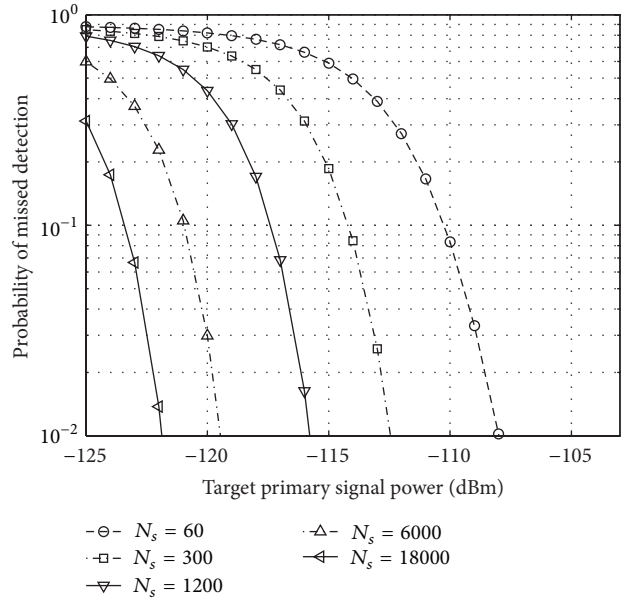


Figure 3: Probability of missed detection for different number of samples (sensing times). Sixty samples correspond to 10 μs. P_{FA} are set to be 10%, and no ACI is used. No device noise uncertainty is considered.

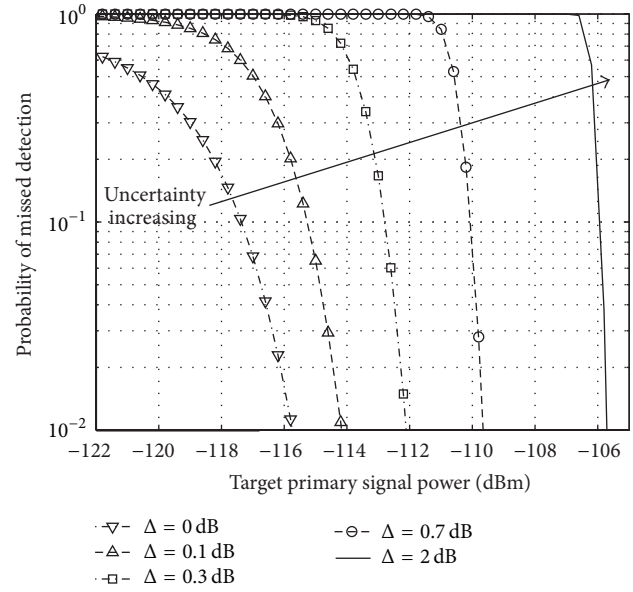


Figure 4: Probability of missed detection for different device noise uncertainties. No ACI is used. P_{FA} and N_s are set to be 10% and 1200 (200 μs), respectively.

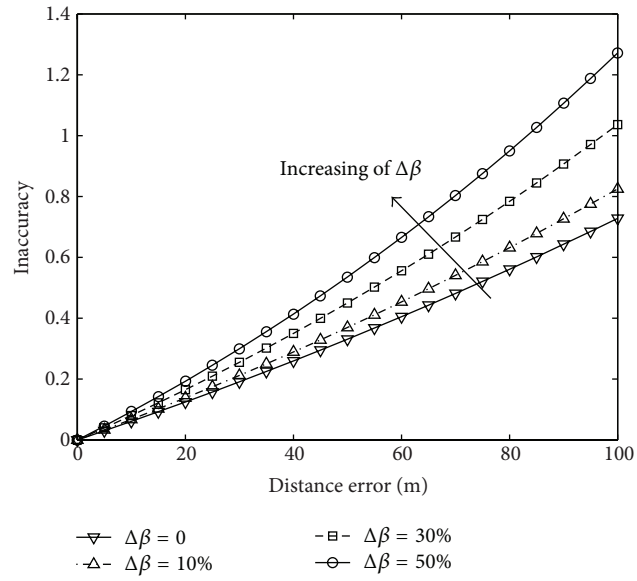


Figure 5: Inaccuracy (defined by $|\widehat{P^t} - P^t|/P^t$) of the solved transmission power of the primary transmitter due to the distance error and β. The actual distance between the primary transmitter and the S_k is 500 m, and the actual β is 3.

the primary transmitter, SUs, and sensors. It also shows that the distance errors result in the similar impact on the performance and that the sensing performance is determined by the accuracy of the solved transmission power of each node. Therefore, to simplify the analysis, let us take a case study on the geolocation error of the primary transmitter, which leads to the distance errors between the primary transmitter and other nodes. We use the inaccuracy of the solved transmission power of the primary transmitter to measure the sensing performance and define it as $|\widehat{P^t} - P^t|/P^t$, where $\widehat{P^t}$ denotes the resulted transmission power of the primary transmitter from the error of the distance and the path loss exponent (denoted by $\Delta\beta$ in Figure 5). Figure 5 illustrates that the solved transmission power becomes inaccurate with the increase in the error of distance and β. It also shows that the inaccuracy becomes serious with the increase of the

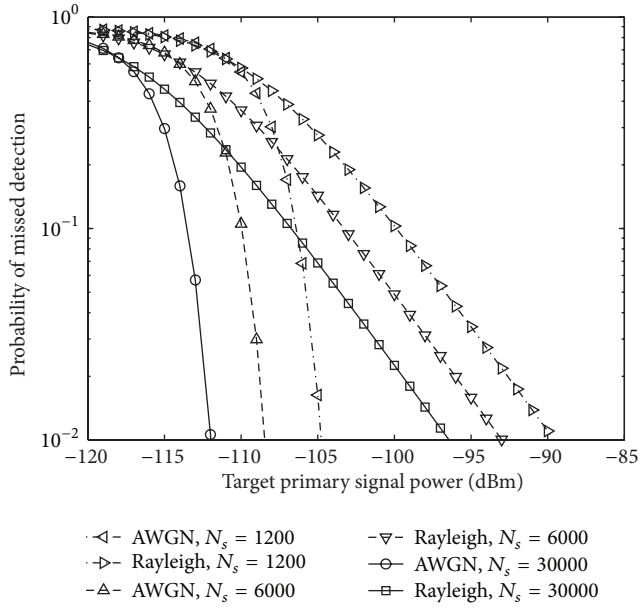


Figure 6: Impact on the performance of the proposed sensing scheme by Rayleigh fading. No device noise uncertainty is considered.

distance error even when β is accurate. When the distance error is fixed, the inaccuracy becomes worse with the increase in the error of β. Moreover, the resulted inaccuracy from the path loss error becomes smaller with the decrease in the distance error. This simulation shows that the sensing performance of the proposed scheme is sensitive to the errors of distance and β. Therefore, the accuracy of the distance and the path loss exponent is important to assure the efficiency of the proposed spectrum sensing scheme. Fortunately, there are many approaches to improve the distance accuracy in the fixed, portable, or mobile environment, for example, the widely used round trip time of flight (RTT) approach in the IEEE 802.11 WLAN; however, they are beyond the scope of this paper.

In the proposed scheme and the previous simulations, we assume the channels are additive white Gaussian noise (AWGN) channels and consider the large scale path loss for cochannel and interchannels by using the interference model. Next, let us evaluate the impact of Rayleigh fading channels.

Figure 6 illustrates the achieved average P_{MD} for AWGN and Rayleigh fading channels. It shows that Rayleigh fading degrades the performance of the proposed sensing scheme. That is to say, the proposed scheme is sensitive to the Rayleigh fading. In fact, we can also expect that the proposed scheme is sensitive to other types of small scale fading because our scheme is essentially based on the large scale path loss. However, Figure 6 shows that the performance in Rayleigh fading can also be improved by using longer observation time, which is similar to the case in AWGN channel. Note that the performance cannot be further improved by lengthening the observation time when the summation of the signal and device noise falls in the uncertainty range of the device noise, as it is analyzed in Section 4.

7. Conclusion

In this paper, we proposed a geolocation assisted spectrum sensing scheme for cognitive coexistent heterogeneous networks. The proposed scheme is able to detect the primary signals and each kind of secondary signals simultaneously. With the assistance of the geographic locations of the PUs, SUs, and the sensors, the detection of primary and secondary signals is formulated into a problem of solving a homogeneously linear equation matrix, the coefficient matrix of which depends only on the distances among the PUs, SUs, and sensors. The proposed scheme is a noncoherent spectrum sensing scheme, which avoids the difficulties in acquiring the characteristics of primary and secondary signals in the heterogeneous networks. In addition, the proposed scheme does not exploit the quiet periods, which not only greatly relaxes the requirement on the network synchronization but also improves the QoS of the secondary networks. Simulation results verify the feasibility and the efficiency of the proposed scheme and show that it can satisfy the sensing requirement of FCC. We also pointed out that the proposed scheme is sensitive to fading and the parameter errors of the employed interference model.

Acknowledgment

This work was conducted under a contract of R&D on intelligent resource utilization technology for future broadband access systems in whitespace with the Ministry of Internal Affairs and Communications, Japan.

References

[1] FCC, "Spectrum policy task force Report," ET Docket No. 02-135, 2002.

[2] M. A. McHenry, E. Livsics, T. Nguyen, and N. Majumdar, "XG dynamic spectrum access field test results," *IEEE Communications Magazine*, vol. 45, no. 6, pp. 51–57, 2007.

[3] R. Yu, Y. Zhang, L. Yi, S. Xie, L. Song, and M. Guizani, "Secondary users cooperation in cognitive radio networks: balancing sensing accuracy and efficiency," *IEEE Wireless Communications*, vol. 19, no. 2, pp. 30–37, 2012.

[4] G. P. Villardi, Y. D. Alemseged, C. Sun et al., "Enabling coexistence of multiple cognitive networks in TV white space," *IEEE Wireless Communications*, vol. 18, no. 4, pp. 32–40, 2011.

[5] A. T. Hoang, Y.-C. Liang, and M. H. Islam, "Power control and channel allocation in cognitive radio networks with primary users' cooperation," *IEEE Transactions on Mobile Computing*, vol. 9, no. 3, pp. 348–360, 2010.

[6] H. Khaleel, F. Penna, C. Pastrone, R. Tomasi, and M. A. Spirito, "Frequency agile wireless sensor networks: design and implementation," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1599–1608, 2012.

[7] H.-S. Chen, W. Gao, and D. G. Daut, "Signature based spectrum sensing algorithms for IEEE 802.22 WRAN," in *Proceedings of the IEEE International Conference on Communications (ICC '07)*, pp. 6487–6492, Glasgow, Scotland, June 2007.

[8] T. Yücek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[9] B. Zhao and S. Sasaki, "A blind power differentiation scheme for energy detection-based spectrum sensing," in *Proceedings of the 18th IEEE International Conference on Networks*, pp. 320–325, Singapore, 2012.

[10] C. Ghosh, S. Roy, and D. Cavalcanti, "Coexistence challenges for heterogeneous cognitive wireless networks in TV white spaces," *IEEE Wireless Communications*, vol. 18, no. 4, pp. 22–31, 2011.

[11] J. Luo, J. Wang, Q. Li, and S. Li, "GLRT based cooperative spectrum sensing with soft combination in heterogeneous networks," in *Proceedings of the IEEE International Symposium on Dynamic Spectrum Access Networks*, pp. 389–396, Washington, DC, USA, 2012.

[12] S. Maharjan, C. Yuen, Y. H. Chew, Y. Zhang, and S. Gjessing, "Distributed spectrum sensing for cognitive radio networks with heterogeneous traffic," in *Proceedings of the 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies (ISABEL '10)*, pp. 1–5, Rome, Italy, November 2010.

[13] J. Huang, R. A. Berry, and M. L. Honig, "Auction-based spectrum sharing," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 405–418, 2006.

[14] Y. Zeng, Y.-C. Liang, A. T. Hoang, and R. Zhang, "A review on spectrumsensing for cognitive radio: challenges and solutions," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, Article ID 381465, pp. 1–15, 2010.

[15] A. Mishra, V. Shrivastava, S. Banerjee, and W. Arbaugh, "Partially overlapped channels not considered harmful," *SIGMETRICS Performance Evaluation Review*, vol. 34, no. 1, pp. 63–74, 2006.

[16] K. R. Chowdhury and I. F. Akyildiz, "Cognitive wireless mesh networks with dynamic spectrum access," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 168–181, 2008.

[17] W. Zhang, R. K. Mallik, and K. Ben Letaief, "Cooperative spectrum sensing optimization in cognitive radio networks," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 3411–3415, Beijing, China, May 2008.

[18] IEEE Std. 802.22-2011, "Part 22: Cognitive Wireless RAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Policies and Procedures for Operation in the TV Bands," IEEE, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 890737, 7 pages
http://dx.doi.org/10.1155/2013/890737

Research Article

Face Recognition in Mobile Wireless Sensor Networks

Qiao-min Lin,[1,2] Jin-wen Yang,[1] Ning Ye,[1,2] Ru-chuan Wang,[1,2,3] and Bin Zhang[4]

[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 210003, China*
[3] *Key Lab of Broadband Wireless Communication and Sensor Network Technology Nanjing University of Posts and Telecommunications, Ministry of Education Jiangsu Province, Nanjing 210003, China*
[4] *EQ Smartech Co. Ltd., Nanjing 211100, China*

Correspondence should be addressed to Qiao-min Lin; lqm@njupt.edu.cn

Received 7 June 2013; Accepted 4 August 2013

Academic Editor: Shukui Zhang

A new wireless sensing network paradigm is presented for face recognition applications. In addition to the flexibility the face recognition system gains by integrating into a wireless sensor network, we take it further by introducing mobility into the network to improve the sensing coverage area and cost efficiency. To implement these goals, a multilayered network structure and Gauss-Markov mobility model are proposed. Furthermore, analysis of the sensing coverage area is given. Besides, some of the potential application scenarios based on the proposed paradigm are also presented. According to the simulation, the whole system achieves high recognition rate and energy efficiency compared to stationary network.

1. Introduction

As one of the most important biometric techniques, face recognition (FR) has advantages of being natural and passive over other biometric techniques requiring cooperative subjects, such as fingerprint recognition and iris recognition [1]. A normal framework of FR system is shown in Figure 1, including procedures of enrollment and identification. It has been widely used in access control, identification systems, and surveillance [2].

Nevertheless, in traditional FR systems, the biometric information is stored remotely in a central database, and the identification device communicates with the database based on traditional wired network [3]. In recent years, some researchers add a wireless dimension to FR systems by combining it with wireless sensor networks (WSNs) [4], which is another hot research topic. For example, Zaeri et al. propose to apply face recognition for wireless surveillance systems [5]. Kim et al. implement a wireless face recognition system based on ZigBee protocol and principle components analysis (PCA) method with low energy consumption [6]. Chang and Aghajan focus on recovering face orientation for more robust face recognition in wireless image sensor networks [7]. Muraleedharan et al. propose to use a specific evolutionary algorithm to optimize routing in distributed time varying network for face recognition [8]. In this work, we take it further by introducing mobility in WSNs for face recognition. To this end, a multilayered structure and Gauss-Markov based mobility model for wireless sensor network are provided. To our best knowledge, it is the first research work proposing an integrated framework of employing face recognition in mobile WSNs (MWSNs). The main contributions of this paper are summarized as follows.

(i) We propose a novel prototype of FR system in the environment of mobile WSNs. The FR system gains flexibility and cost efficiency while being integrated into a wireless work. Meanwhile, the face recognition technique enhances the functionality and security of the wireless network.

(ii) According to the functionality of different components in the network, a multilayered network architecture is built, which introduces a Gauss-Markov mobility model into each level to increase the covering area.

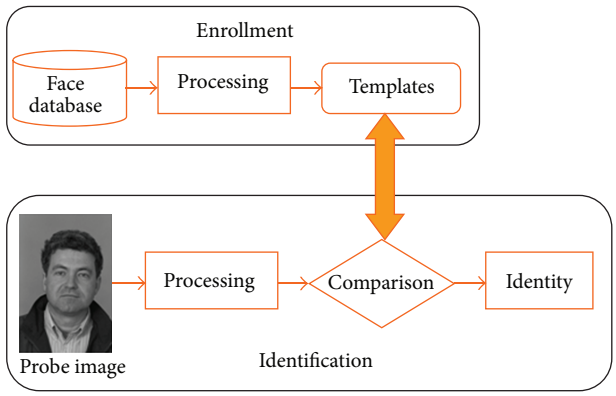


Figure 1: Framework of FR system.

(iii) Last but not least, several potential applications based on the paradigm in this work are presented.

The remainder of this work is organized as follows. In Section 2, background on mobile wireless sensor networks and face recognition is summarized. The structure of proposed face recognition paradigm and the mobility model applied in the sensor network are presented in Section 3. Potential applications based on the mechanism in this work are also investigated in this section. Empirical results and simulation analysis are given in Section 4, and the paper is concluded finally in Section 5.

2. Background

2.1. Mobile Wireless Sensor Networks. Due to recent technological advancement in low power wireless communications and low cost multifunctional sensors, a lot of research efforts have been put in the field of wireless sensor network (WSN), in which a large number of sensor nodes collaborate by means of wireless transmissions for remote monitoring, tracking, and so on [9]. As illustrated in Figure 2(a), a wireless sensor network is usually composed of densely deployed sensor nodes that have been spatially scattered inside the sensing filed. These self-organized sensor nodes will collect the information in interest and deliver it back to the sink node for further transmission via power efficient multihop routing protocols. In traditional static WSNs, the topologies are generally assumed to be invariant, which will subsequently lead to two principal problems: firstly, the network connectivity and coverage may not be guaranteed, in case of sensor failure or malfunctioning (e.g., due to obstacles or energy depletion), which will cause a disconnection between nodes. Secondly, the nearer a sensor node lies relative to the sink node, the faster it will deplete its energy. In fact, the nonuniformity of energy consumption among the sensor nodes will significantly reduce the lifetime of the whole network [10]. To address these problems (connectivity and lifetime), similar to the research trends in mobile ad hoc networks (MANETs) [11] and delay tolerant networks (DTNs) [12], mobile elements (MEs) have also been introduced to WSNs, forming mobile wireless sensor networks (MWSNs)

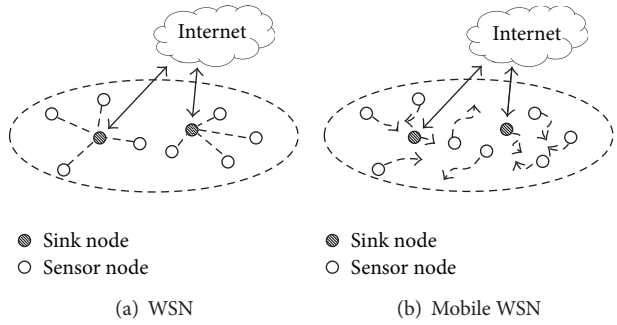


Figure 2: Topology of (a) WSN and (b) mobile WSN.

[13]. MEs play the role of either sinks or sensor nodes in the network, as shown in Figure 2(b).

Due to the fact that MEs can cope with isolated regions, a sparse WSN architecture becomes a feasible option. Besides, by using mobile elements, the battery lifetime of individual sensors can be increased by shifting the energy consumption burden for communication to MEs. Furthermore, reliability can also be achieved owing to the single-hop transmission between MEs and sensors. Finally, in contrast to conventional static WSNs, targets that might never have been detected in a static WSN can now be detected by MEs due to sensors' (or sink's) mobility [10]. Based on these significant advantages, this work concentrates on integrating the application of face recognition into MWSNs.

2.2. Face Recognition. Automatic recognition of faces is considered as one of the fundamental problems in computer vision and pattern analysis, and many scientists from different areas have proposed many approaches to address it [2], including knowledge-based methods which encode facial features according to rules applied based on the typical face [14]; template matching methods which match images to those from a category of stored facial images or features [15]; appearance-based methods which develop models for comparison based on training images [16]; and invariant models which use algorithms to discover facial features even though the view and pose of the subjects and/or lighting conditions change [17]. Among these methods, one of appearance-based methods termed as eigenfaces [16], which is based on PCA, has been applied most extensively. This approach transforms the holistic face image into a small set of eigen vectors, which are the principle components of the initial training set of face images. Recognition is performed by projecting a new probe face image into the subspace spanned by the eigen vectors and then classifying the face by comparing its positions in face space with known individuals in the face database. PCA as well as other variant methods based on it has yielded promising results on frontal face recognition. Hence, in this work, PCA is applied for the task of face recognition.

3. Paradigm of Face Recognition in MWSNs

In this section, we present the architecture and design of the MWSNs. We firstly give an overview of the network

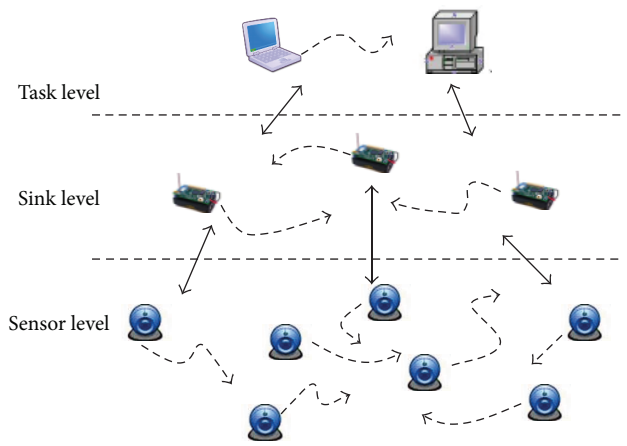


Figure 3: Proposed multilayered structure of face recognition network.

organization at three levels and then describe the mobility model applied in this paradigm. Furthermore, the analysis of coverage is presented. Lastly, we explore the possible applications based on the paradigm.

3.1. Overview. Mobility might be involved at the different network components, as shown in the architecture illustrated in Figure 3.

Depending on the rules that MEs are playing in the network, the structure of the mobile network is divided into three layers.

(i) Sensor level nodes at this level are the sources of information. Such nodes perform sensing as their main responsibility. For the task of face recognition, sensor nodes with cameras are devoted to capturing face images. They may also forward or relay the probe face images in this network, depending on the adopted communication paradigm. The mobile nodes at this level are mainly used to address the problem of sensing coverage. These mobile sensor nodes change their location to better characterize the sensing area or to forward data from the source nodes to the sink. In this case, the primary concern is avoiding coverage holes—areas where the sparse distribution of nodes is not dense enough to properly characterize a phenomenon or detect a probe face.

(ii) Sink level nodes at this level are the destinations of information. They collect face images sensed by sensor nodes either directly (i.e., by visiting sensors and collecting data from each of them) or indirectly (i.e., through intermediate nodes). They can either autonomously process collected facial image data for their own purposes through related image processing algorithms or make them available to remote users by using a long range wireless Internet connection. As a consequence of introducing mobility to this level, the WSN topology can actively change in order to improve both communication reliability and energy efficiency.

(iii) Task level nodes at this level implement the related algorithms for the task of recognition or identification. To implement this, adequate resources of computing, storage, and power should be assigned to the task nodes. For example, the related nodes need to store the eigenvectors of gallery images in the face database. In addition, they are also responsible for measuring the similarity between probe and galley images in the projected eigenspaces. With mobility, the recognition procedure can be implemented with high efficiency.

3.2. Mobility Model in MWSNs

3.2.1. Initial Configuration. In this work, we consider a network consisting of a great number of sensors placed in a vast two-dimensional geographical sensing field R. For the initial configuration, the locations of these sensors are uniformly and independently distributed at random locations in the region. According to our assumption, the sensor locations can be modeled by a stationary two-dimensional Poisson point process. Denote the density of the underlying Poisson point process by λ. The total number of sensors in the region R is $N(R)$, which follows a Poisson distribution with parameter $\lambda\|R\|$, where $\|R\|$ represents the area of the region:

$$P(N(R) = k) = \frac{e^{-\lambda\|R\|}(\lambda\|R\|)^k}{k!}. \quad (1)$$

In a stationary sensor network, sensors do not move after being deployed and network coverage remains the same as that of the initial configuration. While for a mobile sensor network, depending on the application scenario, sensors can choose from a wide variety of mobility strategies, from passive movement to highly coordinated and complicated motion [18], for example, random walk mobility model [19] based on random directions and speeds, boundless simulation area mobility model [20] that converts a 2D rectangular simulation area into a torus-shaped simulation area, Gauss-Markov mobility model that uses one tuning parameter to vary degree of randomness in the mobility pattern [21]. In this work, the Gauss-Markov Model is used to simulate the nodes' motion in the network. We assume sensors move independently of each other and with coordination among them. To characterize the movement model of a sensor, the speed S and direction θ are utilized. For every constant time period, a node calculates the speed and direction of movement based on the speed and direction during the previous time period, along with a certain degree of randomness incorporated in the calculation. The node is assumed to move with the calculated speed and in the calculated direction during the time period. Details of the mobility model will be described in the next section.

3.2.2. Mobility Model. The Gauss-Markov mobility model has been used for the simulation of an ad hoc network protocol [21]. In this section, we shall describe how the model is implemented in mobile WSNs.

Firstly, each ME in the network is assigned an initial speed and direction. At fixed intervals of time, n, movement occurs by updating the speed and direction of each mobile node. Specifically, the value of speed and direction at the nth instance is calculated based upon the value of speed and direction at the $(n+1)$th instance and a random variable using the following equations:

$$\begin{aligned} S_n &= \alpha S_{n-1} + (1-\alpha)\,\overline{S} + \sqrt{(1-\alpha^2)\,S_{n-1}^G}, \\ \theta_n &= \alpha \theta_{n-1} + (1-\alpha)\,\overline{\theta} + \sqrt{(1-\alpha^2)\,\theta_{n-1}^G}, \end{aligned} \tag{2}$$

where S_n and θ_n are the new speed and direction of the MEs at time interval n; α, where $\alpha \in [0, 1]$, is the tuning parameter used to vary the randomness; $\overline{S}$ and $\overline{\theta}$ are constants representing the mean value of speed and direction as $n \to \infty$; and S_{n-1}^G and θ_{n-1}^G are random variables chosen independently by each node from a Gaussian distribution with mean 0 and standard deviation 1. The parameter α is used to incorporate the degree of randomness while calculating the speed and direction of movement for a time period. Totally random values are obtained by setting $\alpha = 0$, and linear motion is obtained by setting $\alpha = 1$. Intermediate levels of randomness are obtained by varying the value of α between 0 and 1. When α is closer to 0, the degree of randomness is high, which may result in sharper turns. When α is closer to 1, the speed and direction during the previous time period are given more importance (i.e., the model is more temporally dependent), and the node prefers to move in a speed and direction closer to what it has been using so far. Thus, the movement of a node gets more linear as the value of α approaches unity. At each time interval, the next location is calculated based on the current location, speed, and direction of movement. Specifically, at time interval n, an ME's position is given by the following equations:

$$\begin{aligned} x_n &= x_{n-1} + S_{n-1} \cos \theta_{n-1}, \\ y_n &= y_{n-1} + S_{n-1} \sin \theta_{n-1}, \end{aligned} \tag{3}$$

where (x_n, y_n) and (x_{n-1}, y_{n-1}) are the x and y coordinates of the ME's position at the nth and $(n-1)$th time intervals, respectively, and S_{n-1} and θ_{n-1} are the speed and direction of the ME, respectively, at the $(n-1)$th time interval.

3.3. Sensing Coverage Area Analysis. In this section, we study and compare the area coverage of both stationary and mobile sensor networks for face recognition.

(1) Stationary Network. We assume that each sensor node in the network can cover a panoramic view of 360 degrees and a disk with radius r. Consider that $\beta \in (0, 1]$, termed as effective recognition factor, denotes the effective range where the sensors can capture face images with resolution high enough for the task of identification. Consider a bounded region R; the vacancy V within R is defined to be the area in R not covered by sensors:

$$V = \int_R \chi(\eta)\, dx, \tag{4}$$

where

$$\chi(\eta) = \begin{cases} 1 & \eta \text{ is not covered} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

According to Tonelli's theorem [22], we have

$$E(V) = \int_R E\{\chi(\eta)\}\, d\eta. \tag{6}$$

Consider an arbitrary point η in region R and denote the number of sensors which cover the point as N. Point η is covered by sensors located within distance r. It follows immediately from the Poisson point process assumption that N has a Poisson distribution with parameters $\lambda\pi(r\beta)^2$. Therefore, we have

$$E\{\chi(\eta)\} = P(N(R) = 0) = e^{-\lambda\pi(r\beta)^2}. \tag{7}$$

Note that the above derivation is independent of R. The area coverage can thus be obtained as follows:

$$f_a = 1 - \frac{E(V)}{\|R\|} = 1 - e^{-\lambda\pi(r\beta)^2}. \tag{8}$$

This formula characterizes the dependence of area coverage on the network properties, that is, the density of the nodes and the coverage area for each node in the sensing field.

In a stationary sensor network, a location always remains either covered or not. The area coverage does not change over time.

(2) Mobile Network. Given the initial node placement and the random mobility model, at any time instant t, the locations of the sensors still form a two-dimensional Poisson point process of the same density. Therefore, the fraction of the area covered at time t remains the same as in the initial configuration, $f_a = 1 - e^{-\lambda\pi r^2}$.

During time interval $[0, t]$, each sensor covers a shape of a track whose expected area is

$$\alpha = E\left[\pi(r\beta)^2 + 2r\beta St\right] = \pi(r\beta)^2 + 2r\beta E[S]\, t, \tag{9}$$

where $E[S] = \int_0^{S_{\max}} G(S) dS$ represents the expected sensor speed. The function $G(S)$ represents a Gaussian distribution, more specifically, a normal distribution in this case.

As pointed out in [23], area coverage depends on the distribution of the random shapes only through its expected area. Therefore, we have

$$f_i(t) = 1 - e^{-\alpha\lambda} = 1 - e^{-\lambda(\pi(r\beta)^2 + 2r\beta E[S]t)}. \tag{10}$$

While an uncovered location will be covered when a sensor moves within distance r of the location, a covered location becomes uncovered as sensors covering it move away. As a result, a location is only covered part of the time. More specifically, a location alternates between being covered and not being covered, which can be modeled as an alternating renewal process. We use the fraction of time when a location is covered to measure this effect. The fraction of

time that a location is covered equals the probability that it is covered at any given time instant, $f_t = 1 - e^{-\lambda\pi(r\beta)^2}$.

At any specific time instant, the fraction of the area being covered in a mobile sensor network model described above is the same as in a stationary sensor network. This is because at any time instant, the positions of the sensor are still described by a Poisson Boolean model with the same parameters as in the initial configuration. Unlike in a stationary sensor network, areas initially not covered can now be covered as sensors move around in a mobile sensor network. Consequently, facial targets in the initially uncovered areas can be detected by the moving sensors. According to (10), the more sensors exist, the more expected coverage area will be achieved, which is the same as stationary network. Furthermore, the fraction of the area that has ever been covered increases as time goes by, and the growth rate depends on the movement speed of sensors involved. The faster the sensors move, the more quickly the area is covered. Therefore, sensor mobility can be exploited to compensate for the lack of sensors to improve the area coverage over an interval of time.

3.4. Potential Applications. In this section, we have listed possible application scenarios for face recognition in MWSNs, with the goals of achieving social interaction, remote surveillance, and security.

3.4.1. Mobile Network Establishment. In this scenario, we consider using mobile smart phones as the mobile elements in the network. All the current smart phones have camera sensors for shooting face images and can be connected to a specific network through cellular network or other protocols such as Bluetooth or ZigBee. Using such mobile devices as sensors has a significant advantage over unattended wireless sensor networks in deploying the sensing hardware and providing it with network. Secondly, mobile phones can provide coverage where static sensors are hard to deploy and maintain. Thirdly, each mobile device is associated with a human user, whose assistance can sometimes be used to enhance application functionality. For instance, a human user may help by pointing the camera appropriately at the target face to be sensed.

In one embodiment, in connection with making comparisons of a captured image with those cataloged in a database, the portable devices used to capture the images may page or multicast signals to other devices within range, sending a query requesting confirmation of the identification. In connection with an affirmative response, further identifying information may be exchanged allowing further messaging such as SMS messages, video or email messaging, and other wireless communication among devices. Based on this identification, one or more mobile networks may be established or extended.

3.4.2. Social Interaction. One other possible scenario of face recognition in WMSNs is that of social interaction. There is a rapid increase in number of mobile subscribers all over the world. With the efficient identification scheme, mobile phones can act as sinks to have a "social interaction" among peers who share the common interest. By entering into a "session" with existing sensors or WSN in a particular area through mobile face recognition, the mobile phone user can get the necessary information on his mobile phone, like the location of his friends/relatives, the time table/schedule of the events taking place, environmental conditions, and so forth. With the help of little initial information about the user, it is also possible to enter into any area, shop around, and buy digital tickets, all with electronic billing.

3.4.3. Remote Wireless Surveillance. In this scenario, our goal is to enhance the utility of the existing surveillance system. In case of monitoring geographically inaccessible or dangerous areas, mobile robots equipped with camera sensors can be deployed for effective coverage. Moreover, if the target to be detected by the sensor network is of time-critical nature, the coverage of the network should be sufficiently high to be able to respond to the detected event in a timely manner, such as monitoring in battle filed or liveliness rescue in case of an earthquake, where the emergency personnel work against the clock.

4. Simulation

In this section, performance comparison of stationary and mobile network in terms of target recognition rate and energy efficiency is provided via Monte Carlo simulations, where each simulation setting is computed over 500 different runs. It is assumed that the range of the mobile nodes, r, is 500 m. The simulation area is circle-shaped with a radius of 4000 m. Initially, mobile nodes are randomly distributed according to the Poisson point process described above in the simulation area. The speed of the mobile nodes is assumed to be 5 m/s. The mobility tuning parameter α in Gauss-Markov is set as 0.5. For the validation of recognition, 200 subjects in FERET dataset [24] are selected. For each subject, one frontal image with regular expression is chosen as the gallery image and one image with alternative expression as the probe image.

4.1. Recognition Rate. In this work, we investigate the recognition of target with a linear motion model. We assume that the target starts its motion from a randomly selected point in one boundary of the simulation area and moves toward a randomly selected point in the opposite boundary following a line. This scenario could be considered an example of border monitoring, where the target tries to cross the border without being detected.

Figure 4 shows the recognition performance of the mobile and stationary WSNs versus number of nodes. It is obvious that the mobility in the network highly improves the detection rate for recognition, which is mainly due to the increase of the coverage area. Furthermore, it can be concluded that the recognition rate has a positive correlation with the number of sensors in the network. To take it further, the impact of motion speed on the recognition rate is examined by modifying the parameter S, as illustrated in Figure 5 (the number of sensor nodes is 50). According

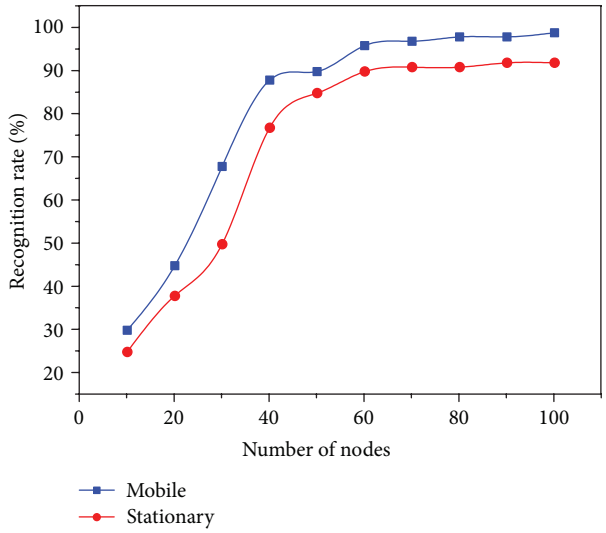


Figure 4: The comparison of recognition rate for stationary and mobile WSNs.

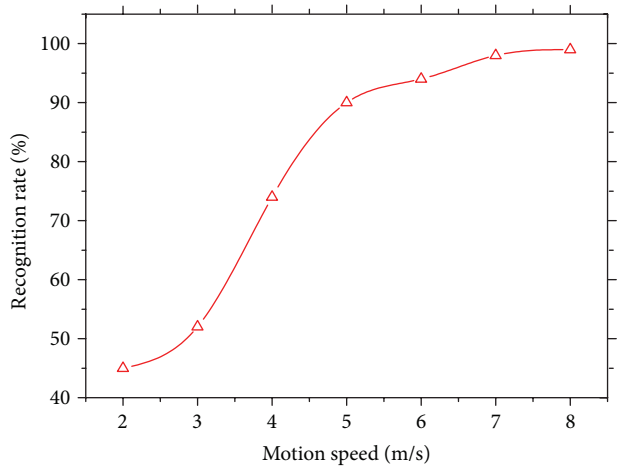


Figure 5: The impact of nodes' motion speed on recognition rate.

to (10), as the motion speed of components grows, the coverage area in the network over an interval increases, thus leading to a higher recognition rate.

4.2. Energy Efficiency. To further analyze the energy efficiency, we use the following energy model [25]. To transmit an k-bit data to a distance d, the radio expends energy:

$$E_{\mathrm{TX}}(k,d) = E_{\mathrm{TX\text{-}elec}}(k) + E_{\mathrm{TX\text{-}amp}}(k,d) = \begin{cases} E_{\mathrm{elec}} * k + k * \varepsilon_{\mathrm{fs}} * d^2, d < d_0 \\ E_{\mathrm{elec}} * k + k * \varepsilon_{\mathrm{mp}} * d^4, d \geq d_0 \end{cases}. \quad (11)$$

The first item denotes the energy consumption of radio dissipation, while the second denotes the energy consumption for amplifying radio. Depending on the distance between the transmitter and receiver, both the free space $\varepsilon_{\mathrm{fs}}$ (d^2 power loss) and the multipath fading $\varepsilon_{\mathrm{mp}}$ (d^4 power loss) channel

Table 1: Parameters for the energy model.

Parameter	Value
Radius R of the sensor field	4000 m
Effective sensing range r of a sensor	500 m
Number N of Nodes in the sensor field	100
E_{elec}	50 nJ/bit
$\varepsilon_{\mathrm{fs}}$, $\varepsilon_{\mathrm{mp}}$	100 pJ/bit/m^2

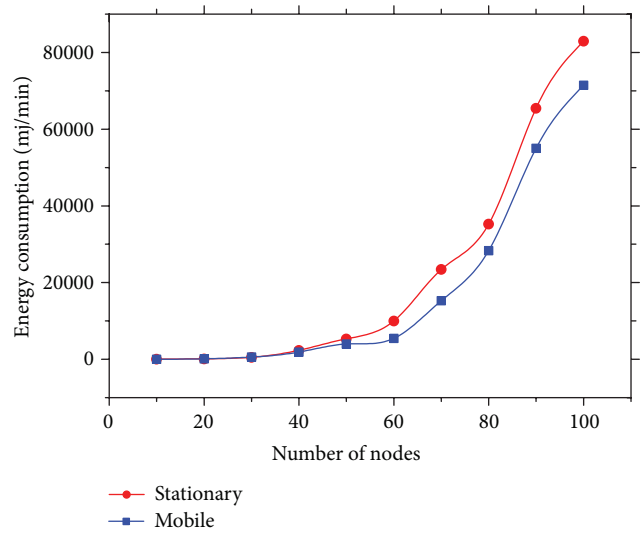


Figure 6: Comparison of the energy consumption between stationary and mobile networks.

models are used. When receiving this data, the radio expends energy:

$$E_{\mathrm{RX}}(k) = E_{\mathrm{RX\text{-}elec}}(k) = E_{\mathrm{elec}} * k. \quad (12)$$

In this work, the related parameters are listed in Table 1.

The comparison of the energy consumption between stationary and mobile network is shown in Figure 6. By introducing mobility into the network components, high energy efficiency can be achieved, and the network lifetime will be prolonged. The reason to explain this is that the related network components, including senor nodes, sink nodes and task nodes, move within the simulated field; energy consumption does not concentrate on several nodes, which means that the distribution of energy consumption is fair over the whole network.

5. Conclusion

In this paper, we have presented a multilayer architecture for the mobile wireless sensor network as a key element of face recognition paradigm. Firstly, the overview of the network's structure is given. Then, the detailed mobility model is presented in this paper, followed by a theoretical analysis of impact of mobility on the sensing coverage area compared to stationary networks. We have further discussed some of the future potential application scenarios for this face recognition paradigm in the environment of mobile WSNs. According to

the simulation validation, by introducing the Gauss-Markov mobility model to different levels in the network, the coverage area and energy efficiency can be improved at the same time.

Acknowledgments

The research is supported by the National Natural Science Foundation of China (Grants nos. 61170065, 61171053, 61003039, 61003236, 61103195, and 61203217), the Natural Science Foundation of Jiangsu Province (Grants nos. BK2011755, BK2012436, and BK20130882), Scientific & Technological Support Project of Jiangsu Province (BE2011844 and BE2011189), Peak of Six Major Talent in Jiangsu Province (Grant no. 2010DZXX026), Project sponsored by Jiangsu provincial research scheme of natural science for higher education institutions (Grant no. 12KJB520009), Science & Technology Innovation Fund for higher education institutions of Jiangsu Province (Grant no. CXZZ11_0405), and Students' Innovative Training Program of Nanjing University of Posts and Telecommunications (SZDG2013042).

References

[1] X. Zhang and Y. Gao, "Face recognition across pose: a review," *Pattern Recognition*, vol. 42, no. 11, pp. 2876–2896, 2009.

[2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: a literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[3] M. Turk and A. P. Pentland, Face Recognition System. US Patent: 5164992, 1992.

[4] 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.

[5] N. Zaeri, F. Mokhtarian, and A. Cherri, "Efficient face recognition for wireless surveillance systems," in *Proceedings of the 9th IASTED International Conference on Computer Graphics and Imaging (CGIM '07)*, E. Gobbetti, Ed., pp. 132–137, OACTA Press, Innsbruck, Austria, February 2007.

[6] I. Kim, J. Shim, J. Schlessman, and W. Wolf, "Remote wireless face recognition employing zigbee," in *Workshop on Distributed Smart Cameras*, 2006.

[7] C. C. Chang and H. Aghajan, "Collaborative face orientation detection in wireless image sensor networks," in *Proceedings of the ACM SenSys Workshop on Distributed Smart Cameras (DSC)*, 2006.

[8] R. Muraleedharan, Y. Yan, and L. A. Osadciw, "Increased efficiency of face recognition system using wireless sensor network," *Systemics, Cybernetics and Informatics*, vol. 4, no. 1, pp. 38–46, 2006.

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

[10] M. D. Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: a survey," *ACM Transactions on Sensor Networks*, vol. 8, no. 1, article 7, 2011.

[11] W. Zhao and M. Ammar, "Message ferrying: proactive routing in highly-partitioned wireless ad hoc networks," in *Proceedings of the 9th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS)*, pp. 308–314, 2003.

[12] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '03)*, pp. 27–34, August 2003.

[13] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: modeling a three-tier architecture for sparse sensor networks," in *Proceedings of the 2nd ACM International Workshop on Wireless Sensor Networks and Applications (SNPA '03)*, pp. 30–41, 2003.

[14] S. Y. Lee, Y. K. Ham, and R. H. Park, "Recognition of human front faces using knowledge-based feature extraction and neuro-fuzzy algorithm," *Pattern Recognition*, vol. 29, no. 11, pp. 1863–1876, 1996.

[15] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042–1052, 1993.

[16] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[17] X. Zhang and Y. Gao, "Face recognition across pose: a review," *Pattern Recognition*, vol. 42, no. 11, pp. 2876–2896, 2009.

[18] 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.

[19] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile users: to update or not to update?" in *Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '94)*, pp. 570–576, June 1994.

[20] Z. J. Haas, "New routing protocol for the reconfigurable wireless networks," in *Proceedings of the 6th IEEE International Conference on Universal Personal Communications (ICUPC '97)*, pp. 562–566, October 1997.

[21] V. Tolety, *Load reduction in ad hoc networks using mobile servers [M.S. thesis]*, Colorado School of Mines, 1999.

[22] C. Freiling, "Axioms of symmetry: throwing darts at the real number line," *The Journal of Symbolic Logic*, vol. 51, no. 1, pp. 190–200, 1986.

[23] P. Hall, *Introduction to the Theory of Coverage Processes*, John Wiley & Sons, 1988.

[24] P. J. Phillips, H. Moon, P. J. Rauss, and S. A. Rizvi, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

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

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 632398, 7 pages
http://dx.doi.org/10.1155/2013/632398

Research Article

Congestion Control Based on Consensus in the Wireless Sensor Network

Xinhao Yang,[1] Juncheng Jia,[2] Shukui Zhang,[2] and Ze Li[3]

[1] *Department of Mechanical and Electrical Engineering, Soochow University, Suzhou 215006, China*
[2] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[3] *College of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China*

Correspondence should be addressed to Xinhao Yang; yangxinhao@163.com

Received 7 June 2013; Accepted 15 August 2013

Academic Editor: Yong Sun

The congestion control algorithm based on the weighted directed graph is designed for the network congestion over the wireless sensor network. The congestion problem is modeled as a distributed dynamic system with time-varying delay, and it can be proven that the sent rate for all nodes converges to the available bandwidth of the sink by the proposed congestion control algorithm. Via Lyapunov function, the validity of the proposed algorithm is shown under the varying network topologies. Ns simulation results indicate that the proposed algorithm restrains the congestion over the wireless sensor network, maintains a high throughput and a low delay time, and also improves the quality of service for the whole network.

1. Introduction

Over the last decades, there have been widely researches in the area of the wireless sensor networks (WSNs) [1]. WSNs are being deployed for several mission-critical tasks, such as habitat monitoring [2], structural health monitoring [3, 4], image sensing [5], and physical game [6]. Typically, a sensor network may contain thousands of nodes, which are cheap, and small size sensors; otherwise, in many applications, the sensor nodes will be deployed in the remote area, such as the high mountain area, and the satellite in the outer space, in which case recharging is not feasible. Thus, the main focus for WSNs is on the low energy use within the autonomous, cooperative nodes which may be constrained in terms of a small memory and a low computing capability.

A wireless sensor network consists of the distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, or pollutants. The physical parameters and the information of the interactions, in conjunction with the variable wireless network conditions, may result in unpredictable behavior in terms of the traffic load variations and the link capacity fluctuations [7]. The network condition is worsened by link bit errors [8], medium contention [9], or potential handoff operations in wireless networks. These hostile factors are likely to occur in WSN environments, thus increasing the possibility of congestion. In data networking and queuing theory, network congestion occurs when a link or node is carrying so much data that its quality of service deteriorates. In the context of WSNs, data is passed using multihop routes between sensors until they reached the sink, so the convergent (many-to-one) nature of WSN, especially in single-sink WSNs, increases the susceptibility to congestion. When a network is congested, it has settled into a stable state where the traffic demand is high, but little useful throughput is available. The high levels of the time delay and packet loss caused by congestion may deteriorate the quality of service (QoS). Especially, packet loss may activate the time-out retransmission scheme of TCP, and the retransmitted packet may worsen the congestion and even cause more retransmission request. The vicious circle may consume more energy in the retransmitted packet repeatedly and ineffectively. Consequently, congestion in WSN causes radical decrease in the delivery ratio and an increase in per-packet energy consumption.

The resource limited and unpredictable characteristics of WSNs necessitate decentralized, robust, self-adaptive, and

scalable mechanisms [7]. The novel congestion control for WSNs should be simple to implement at an individual node level with minimal energy usage. Queue-based congestion control schemes and rate-based ones are the most popular congestion control schemes to solve the congestion problem. The drawback [10] of the queue-based schemes is that a backlog is inherently necessitated; on the other hand, the rate-based schemes [11] can provide early feedback for congestion. So the rate-based scheme is chosen in this paper to solve the congestion problem for WSNs. As the wireless nodes can self-organize into a wireless sensor network, in this architecture, the wireless sensor network is considered as a distributed dynamic system. Natural designs have inherent powerful characteristics and are often more effective and simpler than man-made designs [12]. And the consensus analysis of the complex network theory, such as fish swarming and bird flocking, is used in our paper. With the help of the graph theory, a congestion control algorithm based on consensus analysis (CC-CA) is proposed in this paper, considering the sink as a leader. Ns simulation results indicate that our CC-CA could restrain the congestion over wireless sensor network and maintain a high throughput and a low delay time. It can also improve the QoS for the whole network. As can be seen from the conclusion, CC-CA designed for WSN is superior from throughput, drop ratio, and average delay time of the conventional congestion control.

The rest of the paper is structured as follows: Section 2 presents the previous works on congestion control for WSNs. Section 3 presents the congestion model based on the graph theory for WSN. In Section 4, the theoretical results for congestion control algorithm based on a leader are provided. Section 5 contains a performance evaluation of the proposed scheme and a comparison with TCP protocol. The conclusion of this paper is presented in Section 6.

2. Related Works

The conventional congestion control algorithms are almost end-to-end schemes [14], which follow the TCP's end-to-end mechanism. A centralized congestion control (CC) approach cannot be generally applied since it provokes several serious drawbacks [15]. Firstly, such approach leads to excessive communication load in the network which rapidly depletes the batteries [1]. Secondly, the time-varying nature of the radio channel and the asymmetry in communication links make it harder for even regulated traffic to reach the sink [16]. Thirdly, hop-to-hop schemes [17, 18] could result in a better performance and a faster reaction than the end-to-end mechanisms. Finally, WSNs permit simple processing and a decision making by individual nodes [19] rather than a centralized approach.

Early studies in the area of the sensor networks were mostly focused on fundamental networking problems, for example, topology [20], routing [21], and energy efficiency [22], largely ignoring network performance assurances.

As the queue length in buffer suggested the current network condition, fusion [23] is a congestion mitigation technique that uses queue lengths to detect the congestion. Three different techniques have been adopted to alleviate the congestion in fusion, which included hop-by-hop flow control, rate limiting, and a prioritized MAC. IFRC [24] is a distributed rate allocation scheme that uses queue sizes to detect the congestion and further shares the congestion state through overhearing.

Rate-based congestion control may react more rapidly than queue-based scheme, so a large number of congestion control based on data rate emerges in WSNs. The bandwidth for the sensor node in ARC [25] is split proportionally between route-through and locally generated traffic, by estimating the number of upstream nodes. Congestion control and fairness for many-to-one routing in sensor networks [13] is another rate assignment scheme that uses a different congestion detection mechanism from IFRC.

QoS technique is also widely used in congestion control of the wireless sensor networks. In [26] a dual-path QoS routing protocol is designed to increase the network lifetime and reduce packet delay. MQOSR [27] is another QoS-enabled multipath routing protocol, assuming that the base stations are typically many orders of magnitude more powerful than sensor nodes. Global positioning systemss (GPS) [28, 29], which is providing the location information, is used to discover the congestion regions, while GPS receivers are expensive and not suitable in the construction of the small cheap sensor nodes.

3. Congestion Model Description

The self-organization and multihop characteristic of the WSNs indicate that the wireless sensor network is modeled as a distributed dynamic system based on the directed graph theory. The sink is considered as a "leader" node, in which the mass of information is gathered and computed. We consider that $G = (V, E, \mathbf{A})$ is a weighted directed graph with $n + 1$ nodes, where V denotes the set of vertices v_i ($i \in \mathscr{L} = \{0, 1, 2, \ldots, n\}$), E denotes the set of edges $e_{ij} = (i, j)$, $i, j \in \mathscr{L}$ of the graph G, and $\mathbf{A} = [a_{ij}] \in R^{n\times n}$ for $1, 2, \ldots, n$ is a weighted adjacency matrix.

This paper considers that the vertex indexed by 0 is assigned as the "leader," which is the sink in WSN. The other vertices of the graph G indexed by $1, 2, \ldots, n$ are referred to as "follower agents," which are the autonomous sensor nodes in WSN. When there is data transmission between v_i and v_j, then we consider that there is a path between the two nodes; otherwise $e_{ij} \notin E$. Define the weight matrix $\mathbf{A}$ for the graph G as follows:

$$a_{ij} = \begin{cases} 0.5 & e_{ij} \in E, \\ 0 & e_{ij} \notin E, \end{cases} \tag{1}$$

where $x_i \in R (i \in \mathscr{L})$ is denoted as the data bulk sent to node I, then the differential of x_i denotes the data sent rate. If there is no data communication between node i and others, $x_i = 0$. $G_x = (V, E, \mathbf{A}, x)$ with $x = (x_1, \ldots, x_n)^T$ is referred to an algebraic graph, and then we can say that the algebraic graph $G_x = (V, E, \mathbf{A}, x)$ denotes the WMN's topology.

To study a leader-following problem, the connection weight between nodes i and the leader, denoted by b_i, is shown

as follows. The sink, as the last hop in the monitor area of WSN, is assigned with the largest weight:

$$b_i = \begin{cases} 0.75 & v_i \text{ connected to the leader } v_0, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

In WSN, the sensor may begin to transmit packet suddenly or perform the backoff algorithm due to mutual interference. Namely, for the weighted directed graph $\mathbf{G}$ under consideration, the relationships between neighbors (and the interconnection topology) change over time. Define $\overline{G} = \{\overline{G}_1, \overline{G}_2, \ldots, \overline{G}_N\}$ as a set of the graphs with all possible topologies, which includes all possible interconnection graphs (involving n nodes and a leader), and denote by $S = \{1, 2, \ldots, N\}$ its index set. To describe the variable interconnection topology, we define a switching signal $\varepsilon(t) : [0, \infty) \to S$, which is piecewise constant. Therefore, the connection weights a_{ij} and b_i are time varying, and, moreover, Laplacian $\mathbf{L}_s$ ($s \in S$) associated with the switching interconnection graph is also time varying (switched at t_i, $i = 0, 1, \ldots$).

Remark 1. The topology of WSN is time varying. And this topology is considered unchanged in any interval $[t_i, t_{i+1})$, which is reasonable for the wireless sensor network in this paper. As the network condition is not very bad, the topology of the event-driven network should remain unchanged during a data packet transmitting from one router to another. In other words, if the interval $[t_i, t_{i+1})$ approximates the propagation delay in the network, the topology will remain unchanged.

4. Congestion Control Based on the Consensus Analysis

It is indicated in [30] that, when the offered load exceeds the available capacity in the link, the packet will accumulate in the router buffer, which will induce the congestion. The congestion can be avoided, if the data bulk exchange of all nodes for one task converges to the same equilibrium point in the network. Then the congestion control problem can be attributed to the consensus problem of the complex network. Furthermore, in our simulation, all the nodes are considered the same as each other, and they split the bandwidth fairly.

Here, the entire considered data rate accelerates in a rule:

$$\begin{aligned} \dot{x}_i &= v_i, \\ \dot{v}_i &= \sum_{j \in N_i(t)} a_{ij}(t) \left(x_i(t-r) - x_j(t-r) \right) \\ &\quad + b_i(t) \left(x_0(t-r) - x_i(t-r) \right) + k \left(v_0 - v_i(t) \right), \end{aligned} \quad (3)$$

where N_i is the set of neighbors of node i ($i \in \mathscr{L}$), which is denoted by $N_i = \{v_j \in V : (i, j) \in E_r\}$ and the time-varying delay $r(t) > 0$ is a stochastic function with upper bound τ, which can be determined by the retransmission timeout in TCP.

Formula (3) can be written in a matrix form during the interval $[t_i, t_{i+1})$:

$$\begin{aligned} \dot{x}_i &= v_i, \\ \dot{v}_i &= -\left(\mathbf{L_s} + \mathbf{B_s} \right) x(t-r) - k \left(v - \mathbf{1} \otimes v_0 \right) \\ &\quad + \mathbf{B_s} \mathbf{1} \otimes x_0(t-r), \end{aligned} \quad (4)$$

where $\mathbf{L_s}$ is Laplacian of G_s, $\mathbf{B_s}$ is the leader adjacency matrix whose ith diagonal element is $b_i(t)$, and $\mathbf{1} = [1, 1, \ldots, 1]^T$.

Lemma 2. *Let G be a graph on n vertices with Laplacian $\mathbf{L}$. Denote the eigenvalues of $\mathbf{L}$ by $\lambda_1(\mathbf{L}), \ldots, \lambda_n(\mathbf{L})$; then $\lambda_1(\mathbf{L}) = 0$, and $\mathbf{1} = [1, 1, \ldots, 1]^T$ is its eigenvector.*

Lemma 2 was obtained in [31].

Denote $\widetilde{x} = x - \mathbf{1} \otimes x_0$, and $\widetilde{v} = v - \mathbf{1} \otimes v_0$. Noting that

$$-(L + B) x(t-r) + B1 \otimes x_0(t-r) = -(L + B) \widetilde{x}(t-r), \quad (5)$$

system (4) can be rewritten as

$$\dot{\varepsilon} = C\varepsilon(t) + E\varepsilon(t-r), \quad (6)$$

where

$$\begin{aligned} \varepsilon &= \begin{pmatrix} \widetilde{x} \\ \widetilde{v} \end{pmatrix}, \qquad C = \begin{bmatrix} 0 & I_n \\ 0 & -kI_n \end{bmatrix}, \\ E &= \begin{bmatrix} 0 & 0 \\ -H & 0 \end{bmatrix}, \qquad H = L + B. \end{aligned} \quad (7)$$

Before the discussion of the consensus problem, we introduce Lemma 3 for time-delay system (6). Consider the following system:

$$\begin{aligned} \dot{x} &= f(x_t), \quad t > 0, \\ x(\theta) &= \varphi(\theta), \quad \theta \in [-\tau, 0]. \end{aligned} \quad (8)$$

Lemma 3 (Lyapunov-Razumikhin theorem). *Let Φ_1, Φ_2, and Φ_3 be continuous, nonnegative, and nondecreasing functions with $\Phi_1(s) > 0$, $\Phi_2(s) > 0$, and $\Phi_3(s) > 0$ for $s > 0$, and $\Phi_1(0) = 0$ and $\Phi_2(0) = 0$. For system (8), suppose that the function $f : C([-\tau, 0], R^n) \to R$ takes bounded sets of $C([-\tau, 0], R^n)$ in bounded sets of R^n. If there is a continuous function $V(t, x)$ such that*

$$\Phi_1(\|x\|) \le V(t, x) \le \Phi_2(\|x\|), \quad t \in R,\ x \in R^n. \quad (9)$$

In addition, there exists a continuous nondecreasing function $\Phi(s) > s$, $s > 0$, such that

$$\begin{aligned} &\dot{V}(t, x) \le -\Phi_3(\|x\|), \\ &\text{If } V(t + \theta, x(t + \theta)) < \Phi(V(t, x(t))),\ \theta \in [-\tau, 0], \end{aligned} \quad (10)$$

then the solution $x = 0$ is uniformly asymptotically stable.

Lemma 4. *The matrix $\mathbf{H} = \mathbf{L} + \mathbf{B}$ is positively stable if and only if node 0 is globally reachable in G.*

The proof of Lemma 4 can be looked up in [32].

Theorem 5. *For system* (6)*, take*

$$k > \frac{\overline{\mu}}{2\overline{\lambda}} + 1, \tag{11}$$

where $\overline{\mu} = \max\{\text{eigenvalue of } \overline{P}HH^T\overline{P}\}$, $\overline{\lambda}$ *is the smallest eigenvalue of* $\overline{P}$*, and* $\overline{P}$ *is a positive definite matrix. Then, when delay upper bound* τ *is sufficiently small,*

$$\lim_{t\to\infty} \varepsilon(t) = 0, \tag{12}$$

if node 0 is globally reachable in $\overline{G}$*.*

Proof. Since node 0 is globally reachable in $\overline{G}$, from Lyapunov theorem, there exists a positive definite matrix $\overline{P}$ satisfying

$$\overline{P}H + H^T\overline{P} = I_n. \tag{13}$$

Take a Lyapunov-Razumikhin function $V(\varepsilon) = \varepsilon^T P\varepsilon$, where

$$P = \begin{bmatrix} k\overline{P} & \overline{P} \\ \overline{P} & \overline{P} \end{bmatrix} \tag{14}$$

is positive definite.

By Leibniz-Newton formula,

$$\begin{aligned} \varepsilon(t-r) &= \varepsilon(t) - \int_{-r}^{0} \dot{\varepsilon}(t+s)\,ds \\ &= \varepsilon(t) - C\int_{-r}^{0} \varepsilon(t+s)\,ds - E\int_{-2r}^{-r} \varepsilon(t+s)\,ds. \end{aligned} \tag{15}$$

From $E^2 = 0$, the delayed differential equation (6) can be rewritten as

$$\dot{\varepsilon} = F\varepsilon - EC\int_{-r}^{0} \dot{\varepsilon}(t+s)\,ds, \tag{16}$$

where $F = C + E$. As $2a^Tb \le a^T\psi a + b^T\psi^{-1}b$ holds for any appropriate positive definite matrix ψ, we have

$$\begin{aligned} \dot{V}(\varepsilon) &= \varepsilon^T\left(F^TP + PF\right)\varepsilon - 2\varepsilon^T PEC\int_{-r}^{0} \varepsilon(t+s)\,ds \\ &\le \varepsilon^T\left(F^TP + PF\right)\varepsilon + r\varepsilon^T PECP^{-1}C^TE^TP\varepsilon \\ &\quad + \int_{-r}^{0} \varepsilon^T(t+s)P\varepsilon(t+s)\,ds. \end{aligned} \tag{17}$$

Take $\Phi(s) = qs$ for $q > 1$. In the case of

$$V(\varepsilon(t+\theta)) < qV(\varepsilon(t)), \quad -\tau \le \theta \le 0, \tag{18}$$

we have

$$\dot{V}(\varepsilon) \le \varepsilon^T\left(F^TP + PF\right)\varepsilon + r\varepsilon^T\left(PECP^{-1}C^TE^TP + qP\right)\varepsilon. \tag{19}$$

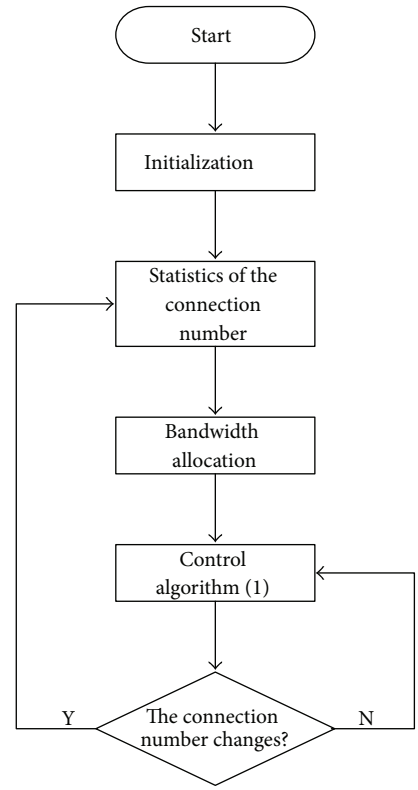


Figure 1: Program flow chart.

If k satisfies (11), $F^TP + PF > 0$ according to Lemma 4 and Schur complements theorem. Let $\lambda_{\min}$ denote the minimum eigenvalue of $F^TP + PF$. If we take

$$r < \frac{\lambda_{\min}}{\left\|PECP^{-1}C^TE^TP\right\| + q\|P\|}, \tag{20}$$

then $\dot{V}(\varepsilon) \le -\eta\varepsilon^T\varepsilon$ for some $\eta > 0$. Therefore, the conclusion follows from Lemma 3. □

Remark 6. There are two limiting conditions in Theorem 5. Firstly, the delay upper bound τ is sufficiently small. The timeout time in the retransmission timeout is small enough with initial value 3 s. Secondly, node 0 is globally reachable in $\overline{G}$. The leader (node 0) is the sink of WSN, through which all the information collected by the sensors is transmitted. So the sink is globally reachable in WSN topology.

Theorem 5 proves that the CC-CA guarantees the convergence of the system error; in other words, the data sent rates for all nodes converge equally to the sink's. The proposed algorithm maintains an equilibrium state for the whole WSN, avoiding congestion. Figure 1 shows the program flow chart of the CC-CA.

5. Simulation

This section studies the performance of the proposed CC-CA under a general wireless sensor network configuration.

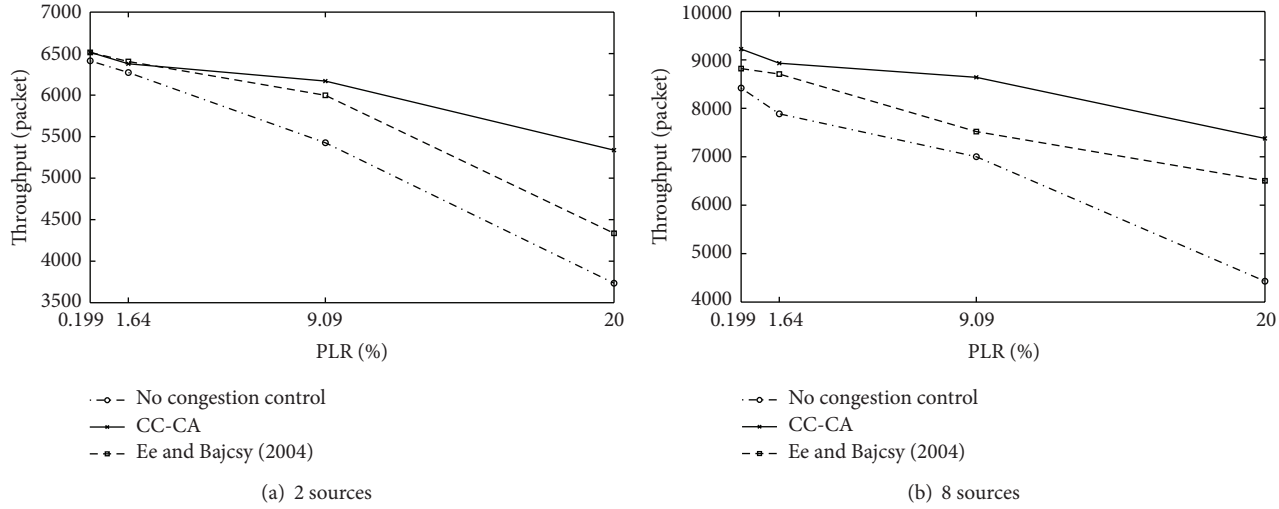


Figure 2: Throughput in different PLRs.

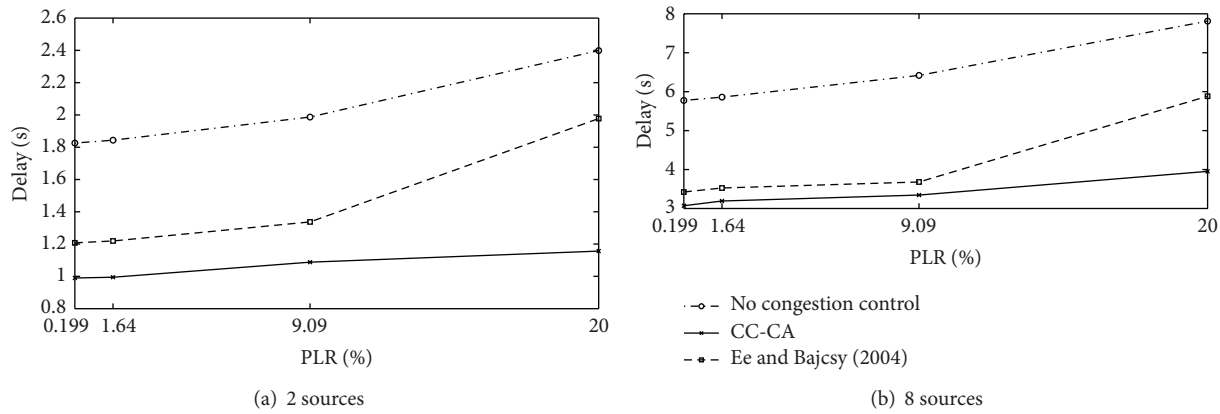


Figure 3: Average delay in different PLRs.

The simulation environment models a sensor network with 200 nodes deployed randomly over an area of 300 × 300 m. The coordinate of the sink is (144, 168). Simulations are conduced using ns-2 network simulator, and the simulation time is 300 sec.

Test 1. In order to verify the validity of the proposed algorithm in variable condition, the number of the data source is set to 2, 5, 8, and 10, respectively. The throughput and the drop ratio are shown in Table 1 for the different connection numbers. The suitable rate for the sink is assigned to all the other sensors, reducing the packet cumulate in the sensor buffer. As discussed in Section 1, the energy usage is the key factor in WSN. The lost parameter η is used to measure the energy efficiency of the whole network, which is defined in [33]. The lost parameter η at distinct rate is shown in Table 2.

Test 2. Test 2 studies the performance of CC-CA by the channel error. The Gilbert model [34] is used to structure the packet loss ratio (PLR) in wireless link. The PLRs involved in Test 2 are provided in Table 3, where p means the probability from "good" state (0% PLR) to "bad" state (100% PLR); q is the probability from "bad" state to "good" one. The throughput and the average delay with different PLRs are shown in Figures 2 and 3, respectively. The performance of CC-CA is tested in varying degrees of congestion by changing the number of the data sources. In our test, most of the packet drop is caused by the link error; by contrast to the traditional wired network, most of the packet losses (sometimes represented as duplicate acknowledgements) are suggested as network congestion notifications, and the end host reduces the transmit rate.

Simulation results indicate that our CC-CA is able to utilize the network resources more efficiently with low drop ratio and low delay time.

6. Conclusion

The congestion problem is unavoidable because of the many-to-one characteristic in the wireless sensor network, which causes the channel quality deterioration and the loss ratio

Table 1: Throughput and drop ratio within different connection numbers.

	Throughput (packet)				Packet drop ratio (%)			
No.	2	5	8	10	2	5	8	10
No congestion control	7047	6457	8860	12926	0.09	0.3	0.38	0.45
Reference [13]	7052	6621	9023	13454	0.08	0.25	0.33	0.39
CC-CA	6932	6655	9707	16598	0.08	0.26	0.27	0.32

Table 2: Loss parameter η at distinct rate.

Time interval(s)	0.02	0.05	0.08	0.1
No Congestion control	1.857	0.923	0.667	0.235
CC-CA	1.101	0.614	0.558	0.222

Table 3: Different PLRs in simulation.

P	q	Packet loss rate (PLR)
0.001	0.5	0.199%
0.01	0.6	1.64%
0.05	0.5	9.09%
0.1	0.4	20%

rise. It leads to packets drops at the buffers, increased delays, wasted energy and requires retransmissions. Based on the consensus problem of the complex network, a novel congestion control algorithm (CC-CA) is introduced in this paper, which provides a better performance. At the same time, only the single-sink topology is discussed in the paper, so the multisink sensor network is the further research. On the other hand, all nodes are considered to be sharing the link capacity fairly in the simulation, so an efficient bandwidth allocation protocol is needed which will improve our algorithm.

Acknowledgments

The authors are indebted to the National Natural Science Foundation of China (61070169, 61203048, and 61201212), the Natural Science Foundation of Jiangsu Province of China (BK2011376), the Application Foundation Research of Suzhou of China (SYG201118, SYG201239), and Specialized Research Foundation for the Doctoral Program of Higher Education of China (no. 20103201110018) for financial support.

References

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

[2] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 214–226, November 2004.

[3] K. Mechitov, W. Y. Kim, G. Agha, and T. Nagayama, "High-frequency distributed sensing for structure monitoring," in *Proceedings of the 1st International Workshop on Networked Sensing Systems (INSS '04)*, 2004.

[4] J. Paek, K. Chintalapudi, R. Govindan, J. Caffrey, and S. Masri, "A wireless sensor network for structural health monitoring: performance and experience," in *Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors (EmNetS '05)*, pp. 1–10, May 2005.

[5] M. Rahimi, D. Estrin, R. Baer, H. Uyeno, and J. Warrior, "Cyclops, image sensing and interpretation in wireless networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, p. 311, November 2004.

[6] W. H. Tan, W. J. Li, Y. Z. Zheng, and X. C. Zhou, "ePet: a physical game based on wireless sensor nerworks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 1, p. 68, 2009.

[7] P. Antoniou, A. Pitsillides, T. Blackwell, A. Engelbrecht, and L. Michael, "Congestion control in wireless sensor networks based on bird flockong behavior," *Computer Networks*, vol. 57, pp. 1167–1191, 2013.

[8] C.-H. Lim and J.-W. Jang, "Robust end-to-end loss differentiation scheme for transport control protocol over wired/wireless networks," *IET Communications*, vol. 2, no. 2, pp. 284–291, 2008.

[9] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP throughput and loss," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, pp. 1744–1753, San Francisco, Calif, USA, April 2003.

[10] T. Nakashima, "Properties of the correlation between queue length and congestion window size under self-similar traffics," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 11, pp. 4373–4381, 2009.

[11] S. S. Kunniyur and R. Srikant, "An Adaptive Virtual Queue (AVQ) algorithm for Active Queue Management," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 286–299, 2004.

[12] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence*, From Natural to Artificial Systems, Oxford, UK, 1999.

[13] C. T. Ee and R. Bajcsy, "Congestion control and fairness for many-to-one routing in sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 148–161, November 2004.

[14] A. Moarefianpour and V. J. Majd, "Input-to-state stability in congestion control problem of computer networks with nonlinear links," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 8, pp. 2091–2105, 2009.

[15] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, John Wiley & Sons, 2005.

[16] S. Brahma, M. Chatterjee, K. Kwiat, and P. K. Varshney, "Traffic management in wireless sensor networks: decoupling congestion control and fairness," *Computer Communications*, vol. 35, no. 6, pp. 670–681, 2012.

[17] S. Sarkar and L. Tassiulas, "Back pressure based multicast scheduling for fair bandwidth allocation," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1279–1290, 2005.

[18] Y. Yi and S. Shakkottai, "Hop-by-hop congestion control over a wireless multi-hop network," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 133–144, 2007.

[19] T. E. Daniel, R. M. Newman, E. I. Gaura, and S. N. Mount, "Complex query processing in wireless sensor networks," in *Proceedings of the 2nd ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks (PMHWN '07)*, pp. 53–60, Chania, Greece, 2007.

[20] S. Jardosh and P. Ranjan, "A survey: topology control for wireless sensor networks," in *Proceedings of the International Conference on Signal Processing Communications and Networking (ICSCN '08)*, pp. 422–427, January 2008.

[21] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–27, 2004.

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

[23] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 134–147, November 2004.

[24] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware fair rate control in wireless sensor networks," in *Proceedings of the ACM SIGCOMM Symposium on Network Architecture and Protocols*, 2006.

[25] A. Woo and D. E. Culler, "A transmission control scheme for media access in sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 221–235, July 2001.

[26] A. Mahapatra, K. Anand, and D. P. Agrawal, "QoS and energy aware routing for real-time traffic in wireless sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 437–445, 2006.

[27] Y. Chen, N. Nasser, T. El Salti, and H. Zhang, "A multipath QoS routing protocol in wireless sensor networks," *International Journal of Sensor Networks*, vol. 7, no. 4, pp. 207–216, 2010.

[28] L. Popa, C. Raiciu, I. Stoica, and D. S. Rosenblum, "Reducing congestion effects in wireless networks by multipath routing," in *Proceedings of the 14th IEEE International Conference on Network Protocols (ICNP '06)*, pp. 96–105, November 2006.

[29] R. Kumar, H. Rowaihy, G. Cao, F. Anjum, A. Yener, and T. L. Porta, "Congestion aware routing in sensor networks," Tech. Rep. 0036, Department of Computer Science and Engineering, Pennsylvania State University, 2006.

[30] S. Li, X. Liao, P. Zhu, and N. Xiao, "Congestion avoidance, detection and mitigation in wireless sensor networks," *Journal of Computer Research and Development*, vol. 44, no. 8, pp. 1347–1356, 2007.

[31] Z. Lin, B. Francis, and M. Maggiore, "Necessary and sufficient graphical conditions for formation control of unicycles," *IEEE Transactions on Automatic Control*, vol. 50, no. 1, pp. 121–127, 2005.

[32] J. Hu and Y. Hong, "Leader-following coordination of multi-agent systems with coupling time delays," *Physica A*, vol. 374, no. 2, pp. 853–863, 2007.

[33] O. Ying, L. Chuang, R. Fengyuan, Y. Hongkun, H. Xiaomeng, and L. Ting, "Design and analysis of a backpressure congestion control algorithm in wireless sensor network," in *Proceedings of the 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT '07)*, pp. 413–420, December 2007.

[34] C. Prawit, "An example of wireless error simulation," http://blog.chinaunix.net/u2/76263/showart_1768145.html.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 404568, 10 pages
http://dx.doi.org/10.1155/2013/404568

Research Article

A Type of Localization Method Using Mobile Beacons Based on Spiral-Like Moving Path for Wireless Sensor Networks

Chao Sha[1] and Ru-chuan Wang[1,2,3]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China*
[3] *Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education, Nanjing, Jiangsu 210003, China*

Correspondence should be addressed to Chao Sha; shac@njupt.edu.cn

Received 7 June 2013; Accepted 25 July 2013

Academic Editor: Hongli Xu

A type of energy optimization localization method with mobile beacon is proposed in this paper. Traverse point of the mobile beacon is marked with the help of the optimum deployment model and the clockwise spiral-like moving path with fixed step size is also established. According to the moving path and the localization time, energy consumption of the network could be estimated and we could also give the sleep scheduling strategy for the node to be localized. Simulation results show that this method could not only promote the accuracy and success rate of localization, but also reduce the energy consumption.

1. Introduction

In Wireless Sensor Networks, sensed data with location information is valuable [1–3]. Several schemes, broadly classified into two categories, have been proposed for dealing with the localization [4, 5]. First, the range-based schemes need either node-to-node distances or angles for estimating locations. The information can be obtained using time of arrival (TOA) [6], time difference of arrival (TDOA) [7], angle of arrival (AOA) [8], and received signal strength indicator (RSSI) technologies [1]. As we know, the range-based schemes typically have higher location accuracy but require additional hardware to measure distances or angles. Second, the range-free schemes do not need the distance or angle information for localization. Although these schemes cannot accomplish as high accuracy as the range-based ones, they provide an economic approach. The representative range-free localization schemes include centroid algorithm [1], DV_HOP algorithm [1], Amorphous localization method [4], APIT algorithm [5] as well as HOP-TERRAIN method [5].

However, the accuracy of current algorithms is mostly environmentally sensitive which leads to low reliability and low success rate about the location results [7, 9]. Meanwhile, locating with the help of fixed beacons will cause the following problems.

First, the network overhead will be increased since the unknown nodes should be directly adjacent to the beacons in order to acquire their location information which leads to high density of nodes.

Second, the communication overhead in localization is larger. The unknown nodes are often in listening mode during the localization process which increase the energy consumption.

Third, to enhance the accuracy of localization, it is necessary to deploy more beacons in the network. However, the more beacon nodes it uses, the greater the overhead layout of the entire network is.

2. Related Works

Therefore, more and more algorithms are proposed to acquire the position information of unknown nodes in wireless sensor networks with the help of mobile beacon, which becomes available with the rapid development of various related research area such as automation and aviation [10–13].

The mobile beacon could be equipped with powerful computational ability to find a result in a very short time which is close to real-time and it could move flexibly according to the motivation it depends on to place human cannot reach. Therefore, diverse algorithms are designed to drive mobile beacon to work on field of Wireless Sensor Networks, but as a primary prerequisite, localization is the first step.

A framework of mobile beacon based localization method is proposed in [11]. Three mobile beacons traverse the field as a group in the shape of isosceles right triangle which enables all the unknown nodes inside the triangle to be localized by receiving three beacon messages. However, it is difficult for these mobile beacons to work synchronously.

Tilak et al. [14] study the time interval for broadcasting of the mobile beacon and propose an adaptive and predictive protocols that control the frequency of localization based on sensor mobility behavior.

Bergamo and Mazzini [15] propose a scheme to perform localization, based on the estimation of the power received by only two beacons placed in known positions. By starting from the received powers, eventually averaged on a given window to counteract interference and fading, the distance between the sensor and the beacons is derived. However, the accuracy of this method is not high.

A type of node localization based on mean shift and joint particle filter is proposed in [16] which improves the accuracy of particle state estimation and reduces the necessary number of samples by using the current observations in sampling procedure to obtain a sample distribution. In addition, Juan et al. propose another mobile beacon localization algorithm based on the Gaussian Markov model [17]. This algorithm combines weighted centroid method and Extended Kalman Filter to ensure that sufficient localization information could be obtained for each unknown node. Nevertheless, the modeling methods of the two aforementioned algorithms are too complicated.

A rectangular trajectory based moving model for the mobile beacon is described in [18]. Although it reduces the energy consumption in communication, the computing cost of localization is higher because the step size of the beacon is not fixed.

As for the moving path, Li et al. propose a method to calculate the coordinates of the sending positions in rectangular ROI (region of interest) [19]. This method is also advanced based on virtual force to arrange the positions in arbitrary ROI. When mobile beacon moves according to the optimal path and emits RF signals at every position, the sensors in ROI could work out their position with multilateration. Yet, Li did not consider the energy consumption of the mobile node.

On the basis of the above research, a type of localization method using mobile beacon based on spiral-like moving path (SHP) for Wireless Sensor Networks is proposed in this paper to further reduce energy consumption as well as improve localization precision.

The next section of this paper provides a detailed realization process of SMP method, and the simulation results are shown in the fourth section. Finally, the conclusion is provided in the last section.

3. Method Description

3.1. Optimal Density of the Beacons in Static Network. The two main factors affecting the quality and cost about localization in wireless sensor networks are the number of the beacons as well as the distribution about them [20].

The density of the beacons is set to $\rho(B)$ and the value of it should neither too large nor too small.

A larger value of $\rho(B)$ could increase the overhead about computation as well as the cost of energy during the localization process. On the contrary, when $\rho(B)$ is too small, the successful rate and the accuracy for the localization algorithm could be unsatisfactory. For this purpose, we should firstly find out the optimal density of the beacons deployed in the network.

As we know, in an omnidirectional wireless sensor networks, when nodes are uniformly distributed and the distance between any two adjacent nodes is $\sqrt{3}r$, the network could be completely covered [21, 22]. r is defined as the communication range of the node, as shown in Figure 1.

On the basis of the model above, we put forward a type of triple coverage model as shown in Figure 2. The size of the network is defined as $L \times L$. Thus, most of the nodes in the network could communicate with at least three noncollinear beacon nodes in this case, which ensures the success rate of localization. $N(B)$ is defined as the number of the beacon nodes deployed in the static network and its value could be calculated by formula (1). k is an arbitrary positive integer constant:

$$N(B) = \begin{cases} \left\lfloor \frac{L}{r} \right\rfloor \times \left(\left\lfloor \frac{L}{r} \right\rfloor + 1 \right) + \left\lfloor \frac{L}{4r} \right\rfloor \times 2 + 1 \\ \qquad \left(\left\lfloor \frac{L}{r} \right\rfloor = 2k \right), \\ \left\lfloor \frac{L}{r} \right\rfloor \times \left(\left\lfloor \frac{L}{r} \right\rfloor + 2 \right) + \left\lfloor \frac{L}{4r} \right\rfloor \times 2 + 2 \\ \qquad \left(\left\lfloor \frac{L}{r} \right\rfloor = 2k + 1 \right). \end{cases} \tag{1}$$

Thus, the optimum density of the beacons deployed in the network could be calculated by formula (2) and the network size is defined as S:

$$\rho(B) = \frac{N(B)}{S} = \frac{N(B)}{L^2}. \tag{2}$$

3.2. Moving Track of the Beacon. In order to show the moving track of the mobile beacon node in detail, we expand the triple coverage model to a more general case as shown in Figure 3.

The gray rectangular area in Figure 3 is assumed as the sensing region whose length and width are defined as L and M, respectively, and L is no small than M in this case. Then, if the beacon nodes are deployed at all these gray points, unknown nodes at any position in the network will acquire their location in theory.

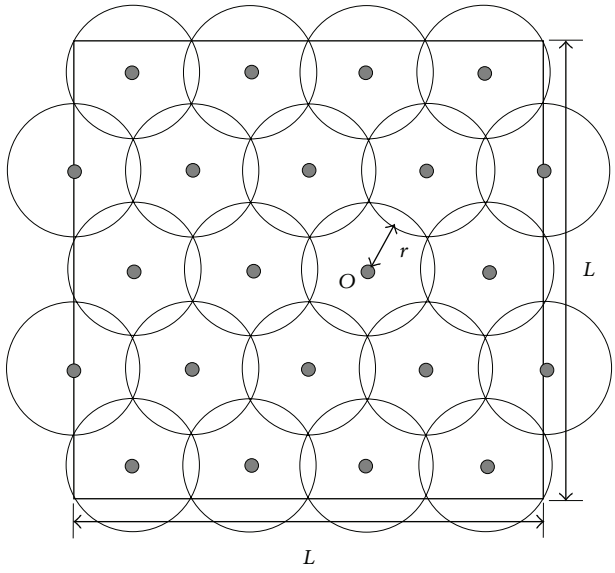


Figure 1: Full coverage model.

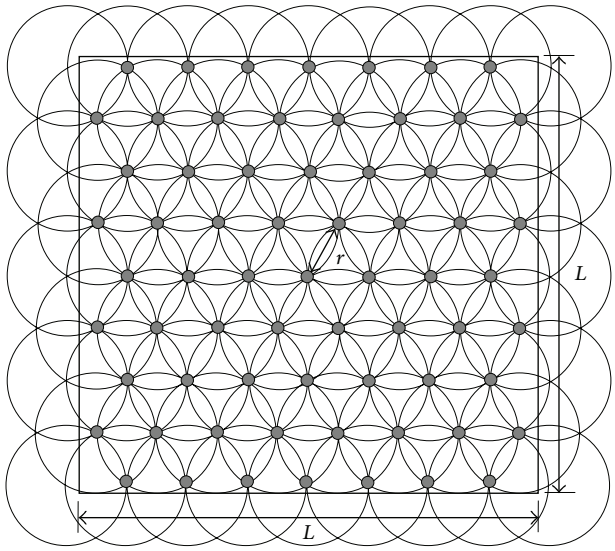


Figure 2: Triple Coverage Model.

Therefore, the minimum and maximum number of beacon nodes, $N_{\min}$ and $N_{\max}$, deployed in each line could be calculated as follows:

$$N_{\min} = \left\lceil \frac{L}{r} \right\rceil + 1,$$
$$N_{\max} = \left\lceil \frac{L}{r} \right\rceil + 2. \tag{3}$$

The number of rows about the beacon nodes needs to be deployed in the network is defined as H:

$$H = 2 \left\lceil \frac{M}{\sqrt{3}r} \right\rceil + 1. \tag{4}$$

Thus, we could get the total number of the beacon nodes, Num, with the help of

$$\text{Num} = \left(\left\lceil \frac{L}{r} \right\rceil + 1 \right) \times \left(\left\lceil \frac{M}{\sqrt{3}r} \right\rceil + 1 \right) + \left(\left\lceil \frac{L}{r} \right\rceil + 2 \right) \times \left(\left\lceil \frac{M}{\sqrt{3}r} \right\rceil \right). \tag{5}$$

As a result, if one beacon could move to these gray points one by one, each of unknown nodes anywhere could also calculate out their coordinates.

One of the feasible tracks for the beacon is moving from the first gray point in the upper left corner of the network to the center clockwise along the spiral-like path as shown in Figure 3. It is easy to know that the length of each moving step is r.

3.3. Network Coordinate System. In order to ensure that each of the unknown nodes could acquire the specific running track of the mobile beacon, a type of cartesian coordinates is established as Figure 4 shows and the origin of which is the network center.

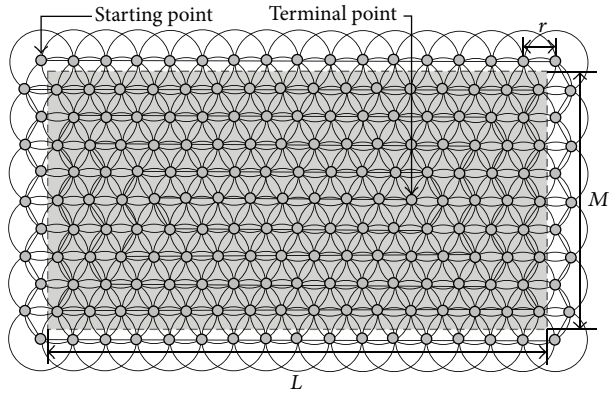


Figure 3: Clockwise spiral-like moving path.

In this paper, the gray points in Figure 3 is called as Broading Points (BPs) because when moving to these points, the beacon node should broadcast their current coordinates to the unknown nodes nearby. Thus, according to the different constraint relationships between L, M, and r, there exist four different cases as follows.

Case 1. The values of $\lceil L/r \rceil$ and $\lceil M/\sqrt{3}r \rceil$ are all odd numbers. In this case, there must be a BP at the network center whose coordinate is (0,0), as shown in Figure 4. It is easy to prove that the coordinates of the gray BPs could be calculated by formula (6) and (7) and the white BPs could acquire their position by formulas (8) and (9). The values of i, j, k, and l are all integers:

$$(x, y) = \left(i \times r, j \times \sqrt{3}r \right) \tag{6}$$

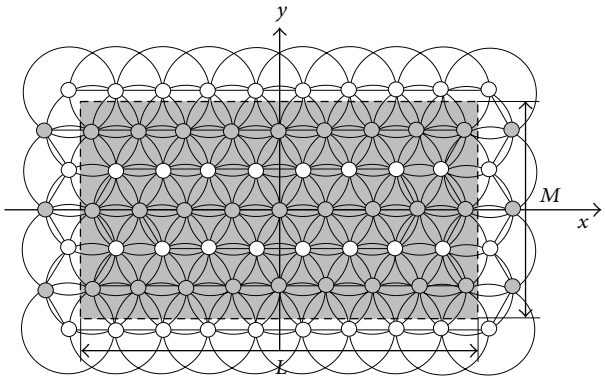


Figure 4: The values of $\lceil L/r \rceil$ and $\lceil M/\sqrt{3}r \rceil$ are all odd numbers.

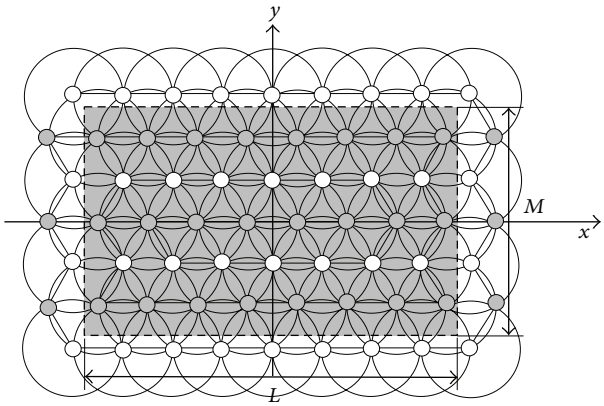


Figure 5: $\lceil L/r \rceil$ is an even number and $\lceil M/\sqrt{3}r \rceil$ is an odd number.

as well as

$$i \in \left[-\left(\frac{\lceil L/r \rceil + 1}{2} \right), \left(\frac{\lceil L/r \rceil + 1}{2} \right) \right],$$
$$j \in \left[-\left(\left\lfloor \frac{M}{\left(2\sqrt{3}r\right)} \right\rfloor \right), \left(\left\lfloor \frac{M}{\left(2\sqrt{3}r\right)} \right\rfloor \right) \right] \tag{7}$$

or

$$(x, y) = \left(\left(k + \frac{1}{2} \right) \times r, \left(\sqrt{3}l + \frac{\sqrt{3}}{2} \right) \times r \right) \tag{8}$$

as well as

$$k \in \left[-\left(\frac{\lceil L/r \rceil + 1}{2} \right), \left(\frac{\lceil L/r \rceil - 1}{2} \right) \right],$$
$$l \in \left[-\left(\frac{\lceil M/\left(\sqrt{3}r\right) + 1 \rceil}{2} \right), \left(\frac{\lceil M/\left(\sqrt{3}r\right) - 1 \rceil}{2} \right) \right]. \tag{9}$$

Case 2. $\lceil L/r \rceil$ is an even number while $\lceil M/\sqrt{3}r \rceil$ is an odd one. There is no BP at the network center in this case as shown in Figure 5. Similar to Case 1, the coordinates of the grey BPs in Figure 5 could be calculated by formulas (10) and (11) and the white BPs could acquire their position by formulas by (12) and (13) as follows:

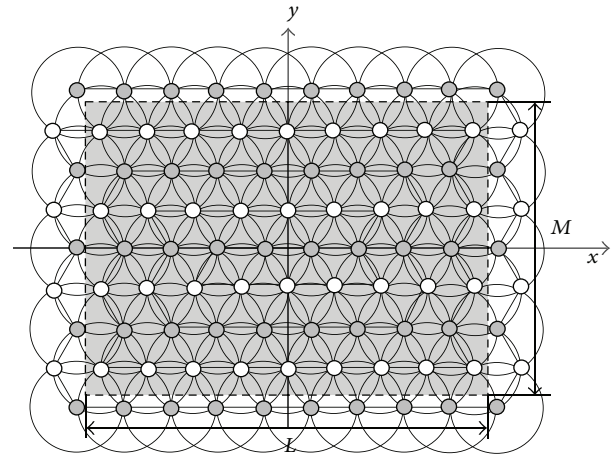


Figure 6: $\lceil L/r \rceil$ is an odd number and $\lceil M/\sqrt{3}r \rceil$ is an even number.

$$(x, y) = \left(\left(i + \frac{1}{2} \right) \times r, j \times \sqrt{3}r \right) \tag{10}$$

while

$$i \in \left[-\left(\frac{\lceil L/r \rceil + 1}{2} \right), \frac{\lceil L/r \rceil}{2} \right],$$
$$j \in \left[-\left(\left\lfloor \frac{M}{\left(2\sqrt{3}r\right)} \right\rfloor \right), \left(\left\lfloor \frac{M}{\left(2\sqrt{3}r\right)} \right\rfloor \right) \right] \tag{11}$$

or

$$(x, y) = \left(k \times r, \left(\sqrt{3}l + \frac{\sqrt{3}}{2} \right) \times r \right) \tag{12}$$

while

$$k \in \left[-\left(\frac{\lceil L/r \rceil}{2} \right), \left(\frac{\lceil L/r \rceil}{2} \right) \right],$$
$$l \in \left[-\left(\frac{\lceil M/\left(\sqrt{3}r\right) + 1 \rceil}{2} \right), \left(\frac{\lceil M/\left(\sqrt{3}r\right) - 1 \rceil}{2} \right) \right]. \tag{13}$$

Case 3. $\lceil L/r \rceil$ is an odd number while $\lceil M/\sqrt{3}r \rceil$ is an even one. There is also no BP at the network center as shown in Figure 6 and the coordinates of the grey and white BPs could then be calculated by, respectively,

$$(x, y) = \left(\left(i + \frac{1}{2} \right) \times r, j \times \sqrt{3}r \right) \tag{14}$$

as well as

$$i \in \left[-\left(\frac{\lceil L/r \rceil + 1}{2} \right), \frac{\lceil L/r \rceil - 1}{2} \right],$$
$$j \in \left[-\left(\left\lfloor \frac{M}{\left(2\sqrt{3}r\right)} \right\rfloor \right), \left(\left\lfloor \frac{M}{\left(2\sqrt{3}r\right)} \right\rfloor \right) \right] \tag{15}$$

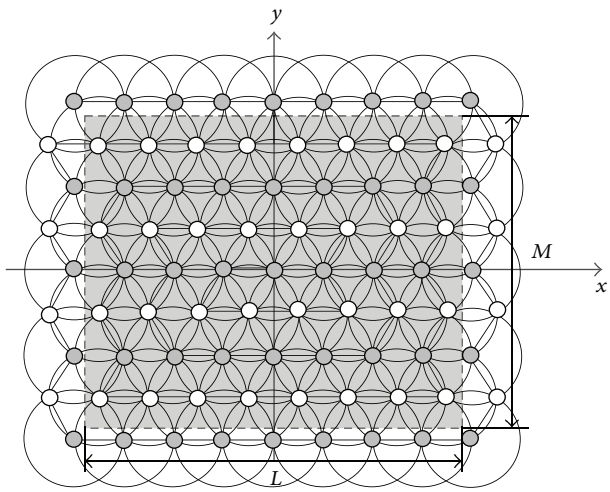


Figure 7: The values of $\lceil L/r \rceil$ and $\lceil M/\sqrt{3}r \rceil$ are all even numbers.

or

$$(x, y) = \left(k \times r, \left(\sqrt{3}l + \frac{\sqrt{3}}{2}\right) \times r\right) \tag{16}$$

as well as

$$k \in \left[-\left(\frac{\lceil L/r \rceil + 1}{2}\right), \left(\frac{\lceil L/r \rceil + 1}{2}\right)\right],$$
$$l \in \left[-\left(\frac{\lceil M/\left(\sqrt{3}r\right) \rceil}{2}\right), \left(\frac{\lceil M/\left(\sqrt{3}r\right) \rceil}{2}\right)\right]. \tag{17}$$

Case 4. The values of $\lceil L/r \rceil$ and $\lceil M/\sqrt{3}r \rceil$ are all even numbers. Similar to Case 1, there must be a BP at the network center in this case (Figure 7). So the grey BPs could be calculated by formulas (18) and (19) and the white BPs could acquire their position by formulas (20) and (21):

$$(x, y) = \left(i \times r, j \times \sqrt{3}r\right) \tag{18}$$

while

$$i \in \left[-\left(\frac{\lceil L/r \rceil}{2}\right), \left(\frac{\lceil L/r \rceil}{2}\right)\right],$$
$$j \in \left[-\frac{\lceil M/\left(\sqrt{3}r\right) \rceil}{2}, \frac{\lceil M/\left(\sqrt{3}r\right) \rceil}{2}\right] \tag{19}$$

or

$$(x, y) = \left(\left(k + \frac{1}{2}\right) \times r, \left(\sqrt{3}l + \frac{\sqrt{3}}{2}\right) \times r\right) \tag{20}$$

while

$$k \in \left[-\left(\frac{\lceil L/r \rceil}{2} + 1\right), \left(\frac{\lceil L/r \rceil}{2}\right)\right],$$
$$l \in \left[-\left(\frac{\lceil M/\left(\sqrt{3}r\right) \rceil}{2}\right), \left(\frac{\lceil M/\left(\sqrt{3}r\right) \rceil}{2}\right)\right]. \tag{21}$$

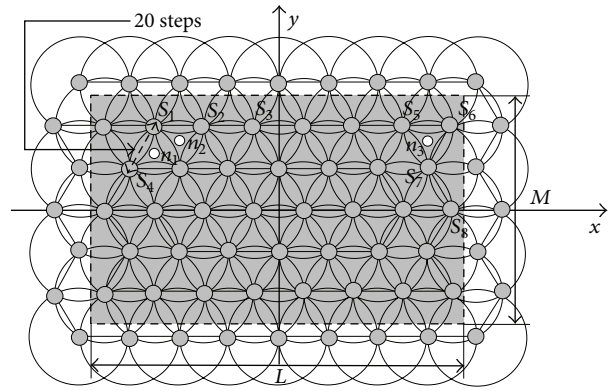


Figure 8: Localization process for unknown nodes.

3.4. Node Localization Based on Mobile Beacon

3.4.1. Moving and Broadcasting Strategy. The behavior of the beacon for localization could be described as follows.

Step 1. The beacon moves from the left corner to the network center clockwise along with the spiral-like path. Its step size and moving speed are r and v, respectively.

Step 2. The beacon stays at each BP for a period of time, defined as t. During this period, it broadcasts its coordinate continuously. The unknown nodes in the beacon's communicate range could receive the data packet when they are in listening mode.

Step 3. The localization algorithm terminates when the beacon moves to the end of the spiral-like path.

3.4.2. Localization Process and Sleeping Scheduling Strategy for Unknown Nodes. According to the track of the mobile beacon, it is easy to see that when one of the unknown nodes received the coordinate information about the mobile beacon for the first time, the following situations may occur.

Case 1. This unknown node could no longer receive any information from the beacon when it moves to the next BP and broadcasts its coordinate at that point. For example, the unknown node n_1 in Figure 8 could not communicate with the beacon when it moves from S_1 to S_2. Furthermore, the next time n_2 receives the coordinate information again is the moment when the beacon just arrives at S_4.

Case 2. The unknown node is still in the beacon's broadcasting range when it moves to the next BP. As shown in Figure 8, n_2 could receive the broadcasting information no matter the beacon is at S_1 or S_2. However, in this case, it could not receive the beacon's coordinate again when the beacon moves to S_3.

Case 3. After receiving the coordinate information for the first time, the unknown node could continuously communicate with the mobile beacon when the beacon moves to the next two BPs. As shown in Figure 8, n_3 could successively

receive the coordinate information of the beacon which stays in S_5, S_6, and S_7.

In addition, it is easy to know that, after communicating with the beacon for the first time, any one of the unknown nodes would at least receive the beacon's real-time coordinate information again when it rotates a circle according to the spiral-like moving path. The length of the time interval between these two information exchanges depends on the moving speed and step size of the beacon. Hence, the sleep scheduling algorithm for localization is described as follows.

Step 1. The unknown nodes in the network are all in listening mode at the beginning of the localization.

Step 2. When any one of the unknown nodes, named as n_i for simplicity, receives the data packets broadcasted by the mobile beacon for the first time, it will further analyse the beacon's coordinate info, $(X_m(j), Y_m(j))$, as follows.

If $|Y_m(j)/X_m(j)| \geq |M/L|$, we define an auxiliary parameter ΔP, as follows:

$$\Delta P = 2 \times \left\lceil \frac{M}{\left(\sqrt{3}r\right)} \right\rceil - \frac{\left(2 \times \left|Y_m\left(j\right)\right|\right)}{\left(\sqrt{3}r/2\right)}. \tag{22}$$

Furthermore, PTIE is defined as a consecutive period of time for information exchange between the beacon and n_i. Therefore, the length of PTIE is from the time when n_i just acquires the real-time coordinate of the beacon to the time when the beacon leaves form the last BP in n_i's communication range. Then the length of time interval between two PTIEs could be approximatively calculated by

$$\Delta T\left(S_i\right) = \left(\frac{r}{v} + t\right) \times \left\{2 \times \left[f\left(2\left\lceil \frac{M}{\left(\sqrt{3}r\right)} \right\rceil - \Delta P - 1\right) + f\left(\left\lceil \frac{L}{r} \right\rceil - \Delta P - 1\right)\right] - 2\right\}. \tag{23}$$

In which, the value of $f(x)$ could be defined as follows:

$$f\left(x\right) = \begin{cases} x & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases} \tag{24}$$

If $|Y_m(j)/X_m(j)| < |M/L|$, the length of time interval between two PTIEs, $\Delta T(S_i)$, could then be expressed as formula (26) in which the value of ΔQ could be calculated by

$$\Delta Q = \left\lceil \frac{L}{r} \right\rceil - 2 \times \left(\frac{\left|X_m\left(j\right)\right|}{r}\right) \tag{25}$$

$$\Delta T\left(S_i\right) = \left(\frac{r}{v} + t\right) \times \left\{2\left[f \times \left(2\left\lceil \frac{M}{\left(\sqrt{3}r\right)} \right\rceil - \Delta Q - 1\right) + f\left(2\left(\frac{\left|X_m\left(j\right)\right|}{r}\right) - 1\right)\right] - 2\right\}. \tag{26}$$

Step 3. In our model, it is easy to know that the unknown nodes could not receive sufficient coordinate information of the beacon for localization in one PTIE. As shown in Figure 8, the unknown node n_2 could only acquire two different coordinates when the beacon moves to S_1 and S_2 in the first PTIE. Therefore, to accomplish localization, any one of the unknown nodes needs to communicate with the beacon in two PTIEs. Consequently, if they could stay in sleeping mode during $\Delta T(S_i)$, the time interval between the two PTIEs, the energy consumption in localization might be reduced.

However, the energy consumption during node's mode switching could not be ignored. Thus, the following formula is proposed:

$$\Delta T\left(S_i\right) \times P_{\text{sleep}} + 2 \times E_{\text{switch}} < \Delta T\left(S_i\right) \times P_{\text{listen}}. \tag{27}$$

It would save energy when the unknown node be is the sleeping mode during $\Delta T(S_i)$ if and only if formula (27) holds up. P_{sleep} and P_{listen} are defined as the power consumption of the unknown node in sleeping and listening mode, respectively, and E_{switch} is the energy consumption in mode switching. Thus, the energy consumption reduced for an unknown node in its localization process could be calculated by

$$\Delta E = \Delta T\left(S_i\right) \times \left(P_{\text{listen}} - P_{\text{sleep}}\right) + 2 \times E_{\text{switch}}. \tag{28}$$

Step 4. After acquiring three or more coordinates in two PTIEs, the unknown node could calculate out its location with the help of the LQI based localization method mentioned in Section 3.4.3.

3.4.3. Localization Based on the Value of LQI. As we know, the free-space model, two-ray ground reflection model, and the shadowing model are the typical signal propagation models in Wireless Sensor Networks. Without losing of generality, we use the free-space propagation model [23] to calculate the distance between the beacon and unknown node in

$$L\left(\text{dB}\right) = PL\left(d_0\right) + 10 \times \eta \times \lg\left[\frac{d\left(S_i, S_j\right)}{d_0}\right] + X_\sigma, \tag{29}$$

$$\text{RSSI}\left(S_i, S_j\right) = P_T - L\left(\text{dB}\right).$$

S_i and S_j are the sender and receiver, respectively, and the distance between them is $d(S_i, S_j)$. RSSI (S_i, S_j) is the signal strength received by S_j and η denotes the path loss factor whose default value is 4. P_T and $PL\left(d_0\right)$ are regarded as the transmission power and the reference received power, respectively, and the values of them are −5 dBm and 55 dB [24]. X_σ is defined as a random variable and its mean and variance are 0 and 5.

Therefore,

$$\text{RSSI}\left(S_i, S_j\right) = P_T - PL\left(d_0\right) - 10 \times \eta \times \lg\left(\frac{d}{d_0}\right) + X_\sigma. \tag{30}$$

In most range-based localization algorithm, the distance between the sender and receiver could be calculated with

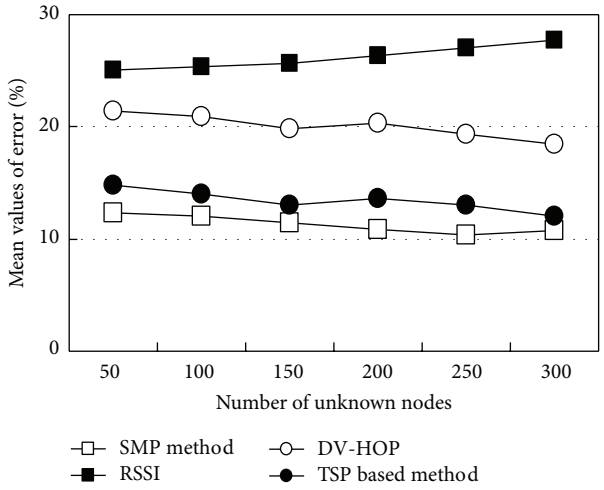


Figure 9: Mean values of error.

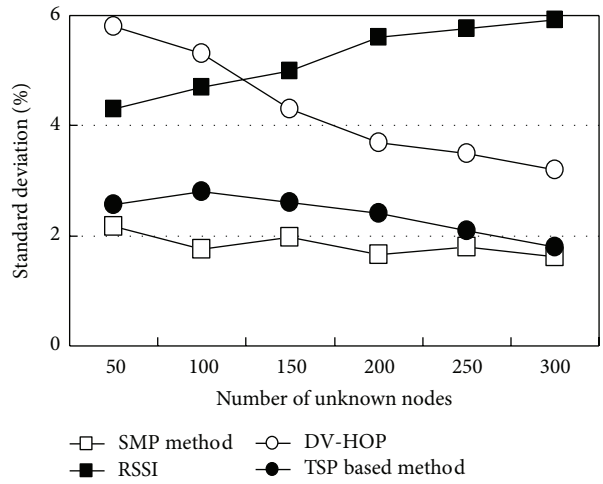


Figure 10: Standard deviation about localization.

Table 1: Values of parameters.

Parameter	Symbol	Value	Unit
Network size	$L \times M$	200×100	M^2
Communication radius of node	r	20	M
Number of beacons in static localization method	$N(B)$	80	
Length of time for the mobile beacon staying at each BP	t	10	S

the help of the Received Signal Strength Indication (RSSI). However, the value of RSSI could not be directly obtained from the data packet and we use the link quality indicator (LQI for short) [25] for distance computation. The value of LQI varies from 0 to 255 with high accuracy and it could be obtained from the MAC layer directly. The relationship between RSSI and LQI is shown in formula (31) [26] and the values of α and β are affected by the environment. Thus, the unknown node would finish its localization with the help of a series of LQIs that it acquires

$$\mathrm{LQI}\left(S_i, S_j\right) = \alpha + \beta \times \mathrm{RSSI}\left(S_i, S_j\right). \tag{31}$$

4. Simulation Results

To testify the effect of localization with the mobile beacon, the simulation is operated under Omnet++3.2 and Matlab7.0. We compare our SMP method to the static localization method by RSSI and maximum likelihood estimation as well as the DV-HOP method. Moreover, we also simulate the localization process about the virtual force based method with the help of a TSP based beacon moving model mentioned in [19].

4.1. Localization Error. Mean values of error about these four types of localization methods are shown in Figure 9 and values of the parameters during this experiment are set in Table 1. It can be seen from the figure that the SMP localization method has the best performance. The mean value of error about it is smaller than 12% and is relatively steady. This is because the mobile beacon moves along the spiral-like path expanded by the triple coverage model which ensures that all of the unknown nodes could acquire three or four beacon nodes' location-specific information nearby and could calculate out their coordinates accurately. Moreover, during the localization process, the mobile beacon broadcasts its coordinate if and only if it arrives at one of the BPs and the unknown nodes only need to be in listening state which avoids the interference of signals. While In TSP based moving model, beacon affected by the virtual repulsive force could hardly move to the network boundary which reduces the accuracy of localization [19].

The standard deviation of these localization algorithms is shown in Figure 10. In SMP method, most of the unknown nodes in the network could receive relatively accurate value of each LQI because the beacon could move along the spiral-like path and broadcast its coordinate at each BP. Therefore, the localization error of each unknown node is close to each other. Moreover, it is easy to know that the accuracy of localization in SMP has nothing to do with the number of the unknown nodes as well as their distribution, which keeps the standard deviation stable. However, in TSP based moving mode, the localization error of the unknown nodes near the boundary of the network is larger which increase the standard deviation.

4.2. Success Rate about Localization. Figures 11 and 12 show the success rate about localization of the four methods under different network sizes. The success rate is defined as the ratio about the number of nodes whose localization error is smaller than the threshold (set as 30%) and the total number of the unknown nodes in the network. As we see, the SMP localization method also has the best performance regardless of the network size as well as the number of the unknown nodes. This is because the BPs are uniformly distributed in the network in our model, so each of the unknown nodes could receive sufficient coordinates to enhance the success rate.

In particular, when the parameter t is larger, the beacon will stay at each BP for a long time which ensures that the

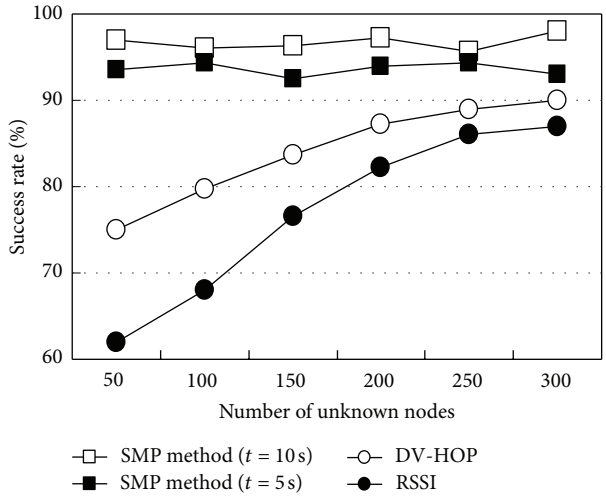


Figure 11: Success rate about localization (network size is 200 M × 100 M).

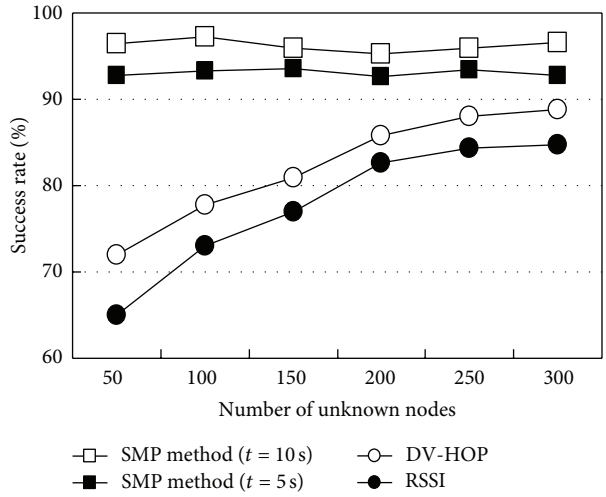


Figure 12: Success rate about localization (network size is 300 M × 200 M).

unknown nodes nearby could receive the accurate coordinate information of the beacon that will further increase the success rate about localization.

4.3. Percentage of Nodes in Sleeping Mode. Table 2 shows the percentage of unknown nodes in sleeping mode during the time interval between the two PTIEs. In this experiment, $v = 10$ m/s, $t = 5$ s, and $r = 20$ m. It is easy to see that this percentage is stable and high when the network size is fixed. Whether or not the unknown node could be in sleeping mode is related to the value of $\Delta T(S_i)$, which is subjected to the value of L, M, v, t, and r, according to formulas (27) and (28). While in our model, the above parameters are all fixed, so the values of the percentage of unknown nodes in sleeping mode keep stable.

On the other hand, with the expansion of the network, the value of $\Delta T(S_i)$ will increase. And because P_{sleep} is far less than

Table 2: Percentage of unknown nodes in sleeping mode.

Network size $L \times M = 200 \times 100$, Unit: M						
N_l	50	100	150	200	250	300
N_s	37	72	111	159	189	239
K	74%	72%	74%	79.5%	75.6%	79.7%
Network size $L \times M = 300 \times 200$, Unit: M						
N_l	50	100	150	200	250	300
N_s	41	81	133	169	220	258
K	82%	81%	88.7%	84.5%	88%	86%

(N_l: number of the unknown nodes; N_s: number of the sleeping nodes; K: percentage).

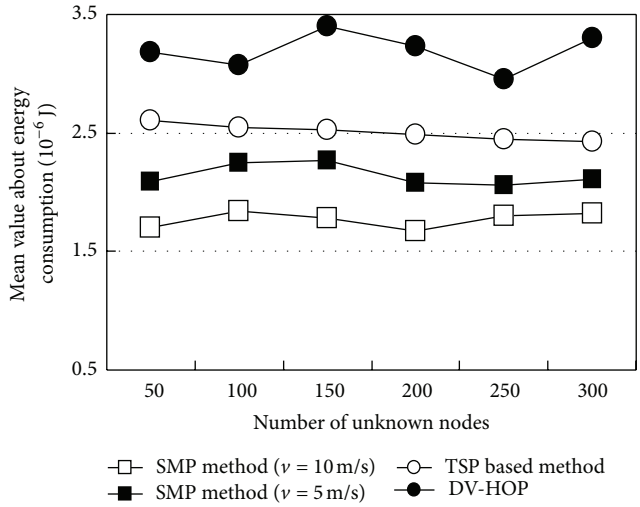


Figure 13: Energy consumption in localization (network size is 200 M × 100 M).

P_{listen}, there must be more nodes in sleeping mode. Thus, the percentage of unknown nodes in sleeping mode will increase when the network size is larger.

4.4. Energy Consumption in Localization. Figures 13 and 14 show the energy consumption of the network during the localization process. For the sake of simplicity, we ignore the mobile beacon's energy consumption in SMP method. As we see, in SMP, most of the unknown nodes could go into sleeping mode most of the time during the localization process and the unknown nodes do not need to send any message to the beacon which greatly reduces the energy consumption in localization.

Although the triple coverage model is also used in the TSP based beacon moving model, this localization method could only apply to the network with a circular shape and the redundancy of coverage is also high. Furthermore, the number of BPs of this algorithm is more than the SMP method which leads to high energy consumption.

From Figures 13 and 14, it is also known that the average energy consumption in SMP is unrelated to the number of the unknown nodes, because the distributed computing method

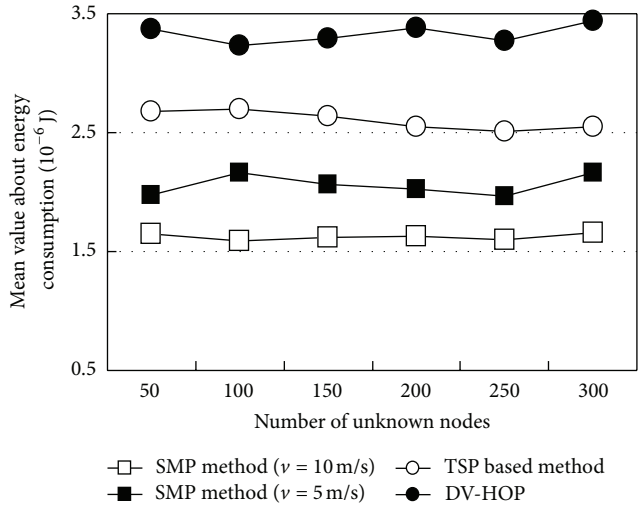


Figure 14: Energy consumption in localization (network size is 300 M × 200 M).

is used in SMP and nodes do not need to communicate with its neighbors during the localization process.

5. Conclusion

A type of energy efficient localization method based on mobile beacon is proposed in this paper. Each of the unknown nodes could not only acquire sufficient information of the beacon, but also go into sleeping mode to save energy. Simulation results show that the SMP method increases the accuracy of localization for Wireless Sensor Networks and reduces energy consumption of the unknown nodes.

Acknowledgments

The subject is sponsored by the National Natural Science Foundation of China (61202355), Research Fund for the Doctoral Program of Higher Education of China (20123223120006), China Postdoctoral Science Foundation (2013M531394), Natural Science Foundation of Jiangsu Province (BK2012436), Jiangsu Provincial Research Scheme of Natural Science for Higher Education Institutions (11KJB520014), Postdoctoral Foundation of Jiangsu Province (1202034C), and the Scientific Research Fund Project for Translation Talents of Nanjing University of Posts and Telecommunications (NY211018).

References

[1] W. Shanshan, Y. Jianping, C. Zhiping, and Z. Guomin, "A RSSI-based self-localization algorithm for wireless sensor networks," *Journal of Computer Research and Development*, vol. 45, no. 1, pp. 385–388, 2008.

[2] C. H. Ou and W. L. He, "Path planning algorithm for mobile anchor-based localization in wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 2, pp. 466–475, 2013.

[3] H. Chen, Q. Shi, R. Tan, H. V. Poor, and K. Sezaki, "Mobile element assisted cooperative localization for wireless sensor networks with obstacles," *IEEE Transactions on Wireless Communications*, vol. 9, no. 3, pp. 956–963, 2010.

[4] T. He, C. D. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 81–95, September 2003.

[5] Y. Yibing, *Localization Algorithms for Sensor Networks and Related Technologies*, Chong Qing University, Chongqing, China, 2006.

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

[7] Y. Haibing and Z. Peng, *Intelligent Wireless Sensor Networks*, vol. 1, Science Press, Beijing, China, 2006.

[8] D. Niculescu and B. Nath, "Ad Hoc positioning system (APS) using AOA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, vol. 3, pp. 1734–1743, April 2003.

[9] S. Zhang, J. Cao, C. Li-Jun, and D. Chen, "Accurate and energy-efficient range-free localization for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 6, pp. 897–910, 2010.

[10] E. Kim and K. Kim, "Distance estimation with weighted least squares for mobile beacon-based localization in wireless sensor networks," *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 559–562, 2010.

[11] D. Weijun, H. Xiaoli, W. Fubao, and L. Yanwen, "Research on distance measurement in wireless sensor networks," *Computer Science*, vol. 9, pp. 51–62, 2007.

[12] J. P. Sheu, W. K. Hu, and J. C. Lin, "Distributed localization scheme for mobile sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 516–526, 2010.

[13] 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.

[14] S. Tilak, V. Kolar, N. B. Abu-Ghazaleh, and K. D. Kang, "Dynamic localization control for mobile sensor networks," in *Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC '05)*, pp. 587–592, April 2005.

[15] P. Bergamo and G. Mazzini, "Localization in sensor networks with fading and mobility," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '02)*, pp. 750–754, September 2002.

[16] L. Haiyong, L. Jintao, Z. Fang, and L. Yiming, "Mobile node localization based on mean shift and joint particle filter in wireless sensor networks," *Chinese Journal of Sensors and Actuators*, vol. 22, no. 3, pp. 378–386, 2009.

[17] C. Juan, L. Chang-Geng, and N. Xin-Xian, "Node localization of wireless sensor networks based on mobile beacon," *Chinese Journal of Sensors and Actuators*, vol. 22, no. 1, pp. 121–125, 2009.

[18] D. Qifen, F. Yuanjing, and Y. Li, "Location algorithm based on the mobile beacon node in wireless sensor networks," *Chinese Journal of Sensors and Actuators*, vol. 21, no. 5, pp. 823–827, 2008.

[19] S. Li, C. Xu, Y. Yang, and Y. Pan, "Getting mobile beacon path for sensor localization," *Journal of Software*, vol. 19, no. 2, pp. 455–467, 2008.

[20] C. Weike, L. Wenfeng, S. Heng, and Y. Bin, "Weighted centroid localization algorithm based on RSSI for wireless sensor networks," *Journal of Wuhan University of Technology*, vol. 30, no. 2, pp. 265–268, 2006.

[21] H. H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *Wireless Ad Hoc and Sensor Networks*, vol. 1, no. 1-2, pp. 89–123, 2005.

[22] C. F. Huang and Y. C. Tseng, "The coverage problem in a wireless sensor network," in *Proceedings of the 2nd ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '03)*, K. M. Sivalingam and C. S. Raghavendra, Eds., pp. 115–121, ACM Press, San Diego, Calif, USA, September 2003.

[23] S. Y. Seidel and T. S. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 2, pp. 207–217, 1992.

[24] K. Yedavalli, B. Krishnamachari, S. Ravulat, and B. Srinivasan, "Ecolocation: a sequence based technique for RF localization in wireless sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 285–292, IEEE Computer Society, Los Angeles, Calif, USA, April 2005.

[25] K. Whitehouse, *The Design of Calamari. An Ad-Hoc Localization System for Sensor Networks*, University of California, Berkeley, Calif, USA, 2002.

[26] CHIPCON AS and SMART R F, "CC2420 Preliminary Datasheet (rev1. 3) [EB/OL]," 2008, http://www.ti.com/lit/ds/symlink/cc2420.pdf.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 216561, 8 pages
http://dx.doi.org/10.1155/2013/216561

Research Article

Fuzzy-Logic-Based Energy Optimized Routing for Wireless Sensor Networks

Haifeng Jiang,[1] Yanjing Sun,[2] Renke Sun,[1] and Hongli Xu[3]

[1] *School of Computer Science & Technology, China University of Mining and Technology, Xuzhou 221116, China*
[2] *School of Information & Electrical Engineering, China University of Mining and Technology, Xuzhou 221116, China*
[3] *Suzhou Institute for Advanced Study, University of Science and Technology of China, Suzhou 215123, China*

Correspondence should be addressed to Yanjing Sun; yanjingsun_cn@hotmail.com

Received 3 May 2013; Accepted 11 July 2013

Academic Editor: Shukui Zhang

Wireless sensor nodes are usually powered by batteries and deployed in unmanned outdoors or dangerous regions. So, constrained energy is a prominent feature for wireless sensor networks. Since the radio transceiver typically consumes more energies than any other hardware component on a sensor node, it is of great importance to design energy optimized routing algorithm to prolong network lifetime. In this work, based on analysis of energy consumption for data transceiver, single-hop forwarding scheme is proved to consume less energy than multihop forwarding scheme within the communication range of the source sensor or a current forwarder, using free space energy consumption model. We adopt the social welfare function to predict inequality of residual energy of neighbors after selecting different next hop nodes. Based on energy inequality, the method is designed to compute the degree of energy balance. Parameters such as degree of closeness of node to the shortest path, degree of closeness of node to Sink, and degree of energy balance are put into fuzzy logic system. Fuzzy-logic-based energy optimized routing algorithm is proposed to achieve multiparameter, fuzzy routing decision. Simulation results show that the algorithm effectively extends the network lifetime and has achieved energy efficiency and energy balance together, compared with similar algorithms.

1. Introduction

Wireless sensor networks (WSNs) have emerged as an attractive technology for their wide range of applications in civil and military areas. In contrast to traditional wireless networks, wireless sensor nodes are usually powered by batteries and deployed in unmanned outdoors or dangerous regions [1, 2]. So, energy-constraint is a prominent feature for wireless sensor networks. Since the radio transceiver typically consumes more energy than any other hardware components onboard a sensor node, designing energy optimized routing algorithm is of great importance to prolong network lifetime [3].

Due to limited power on each sensor node, the routing algorithm should seek for energy efficiency and find less energy-consuming paths to transmit data. Intuitively, the network lifetime should be extended. However, most energy-efficient routing algorithms tend to route data via sensors on energy-efficient paths and thereby drain their energy quickly. For the ultimate goal of wireless sensor networks is to maximize network lifetime, significant efforts have been done to improve energy-efficient routing for the perspective of energy balance. In [4], single-hop or multihop forwarding scheme is selected to transmit data to Sink, according to the ability of sensor nodes. The direct transmission mode can save energy of the nodes closer to the Sink since their relaying burden can be relieved in this mode. EBDG [5] takes full advantage of corona-based network partition, mixed routing, and data aggregation to balance energy consumption. In [6], multipath mechanism is used to achieve energy balance. In [7], a proactive multipath routing algorithm is provided to achieve spatial energy balance, but it is actually a load balancing mechanism because of the assumption that "energy burden" and "traffic load" can be assimilated. However, it is not an optimal solution because spreading traffic unaware of residual energy distribution is somewhat blindfold. To balance the energy consumption in WSNs, residual energy scheming based energy equilibrium routing

protocol (RESEE) is proposed in [8]. In RESEE, a fuzzy gradient classification based next hop strategy has been designed to balance the energy consumption as a whole. In [9], using the variance of residual energy of sensors to measure the energy balance, predicting based distributed energy-balancing routing algorithm has been proposed. In [10], maximizing network lifetime by energy-aware routing has been formulated as integer programming problem, achieved energy efficiency as well as energy balance. But this algorithm has the central control architecture, and Sink needs to collect information of nodes and broadcast data transmission matrix to determine routes periodically, leading to heavy overhead of communication.

To prolong the lifetime of wireless sensor networks, the routing algorithm must be designed to achieve both energy efficiency and energy balance together. It should not only reduce the energy consumption for data transmission to extent the lifetime of a single node but also balance the energy consumption for the whole network. However, it is hard to optimize energy efficiency and energy balance simultaneously, which is difficult to be accurately described by mathematical model. How to realize the optimal combination of energy efficiency and energy balance is the key issue to extend the network lifetime. Fuzzy logic, on the other hand, has potential for dealing with conflicting situations and imprecision in data using heuristic human reasoning without needing complex mathematical model [11]. It is very well suited for implementing routing and clustering heuristics and optimizations, like link or cluster head (CH) quality classification [12, 13]. Judicious cluster head election can reduce the energy consumption and extend the network lifetime. A fuzzy logic approach based on energy, concentration, and centrality is proposed for cluster head election in [14]. This mechanism has some demerits that are caused by the centralized election mechanism. The Sink has to collect the energy and distance information from all sensor nodes. In [15], based on the improvement of the mechanism in [14], CHEF algorithm is proposed, which is a localized cluster head election mechanism using fuzzy logic. In [16], an energy-aware distributed dynamic clustering protocol (ECPF) is proposed, in which fuzzy logic is employed to assess the fitness (cost) of a node to become a CH. Both node degree and node centrality are taken into account to compute fuzzy cost. Simulation results show that ECPF provides superior network lifetime and energy savings than CHEF. In [17], a fuzzy-logic-based clustering approach with an extension to the energy predication has been proposed to prolong the lifetime of WSNs. In addition to the residual energy, the expected residual energy has been introduced to act as a fuzzy descriptor during the online CH selection process.

Although there are some researches using fuzzy logic to optimize cluster head election, but in the field of using fuzzy logic approach for flat routing has not been studied enough. In [11], the gateway is responsible for setting up of routes for sensor nodes and for the maintenance of centralized routing table that indicates the next hop for each sensor node. Gateway uses fuzzy logic to determine the cost of link between any two sensor nodes by input fuzzy variables, such as transmission energy, remaining energy, and queue size. Once the costs of all possible links to the gateway are computed, the route will be determined using the shortest path algorithm. But this approach is centralized, which is not suitable for the widely distributed WSNs. In [18], angle of placement and number of packets forwarded to the neighboring node are used as fuzzy system input parameters, and the node with greater chance is selected as next hop. In this method, the number of packets forwarded to neighboring nodes takes the place of the residual energy of nodes. But there are many packets sent from other nodes, so this replacement is not accurate.

The rest of this paper is organized as follows. Section 2 introduces the system model and defines data generation patterns. In Section 3, three energy optimized parameters are defined. A detailed description of fuzzy-logic-based energy optimized routing is given in Section 4. The simulation model and the comparative performance evaluation of the proposed routing algorithm are presented in Section 5. Section 6 concludes this paper.

2. System Model and Problem Specification

There are n homogenous sensor nodes randomly and uniformly distributed over a target area, and a Sink node collects events or sensed data from the sensors in each round. The primary design objective of the routing algorithm is to maximize the network lifetime. We clarify the problem by detailing energy consumption model and data generation patterns.

2.1. Energy Consumption Model. The energy consumption of each sensor node consists of three components: sensing energy, communication energy, and data processing energy. Sensing and data processing require much less energy than communication, so we consider only communication energy consumption. We use the same energy consumption model as Heinzelman used it for wireless communication hardware [19]. If the node transmits an l-bit packet over distance d, the consumed energy is

$$E_{\mathrm{Tx}}(l, d) = lE_{\mathrm{elec}} + l\varepsilon_{\mathrm{amp}}d^{\alpha}, \tag{1}$$

where E_{elec} denotes the energy/bit consumed by the transmitter electronics. $\varepsilon_{\mathrm{amp}}$ denotes the energy dissipated in the transmission amplifier and α represents the path loss exponent. The value of α is 2 for free space channel model and 4 for multipath fading channel model.

When receiving an l-bit packet, the energy consumption is

$$E_{\mathrm{Rx}}(l) = lE_{\mathrm{elec}}. \tag{2}$$

2.2. Data Generation Patterns. Many previous studies assume that each sensor has to send data to Sink in each round. That is, all sensors have a uniform data generation rate. However, in many applications, this assumption becomes unrealistic. In the case of forest fire detection, events can occur rarely and randomly over the target area. Therefore, the consideration of diverse potential data generation patterns is more reasonable.

For our work, three data generation patterns are considered as follows.

(i) Uniform data generation: every sensor transmits a data packet to the Sink in each round.

(ii) Random data generation: every sensor reports a data packet to the Sink with probability p in each round.

(iii) Data generation from a local area: only sensors in a local area have data to be transmitted to the Sink in each round. The shape of the area can be a circle, a square, or any other.

3. Energy Optimized Parameters

3.1. Degree of Closeness of Node to the Shortest Path. According to the energy consumption model of sensor nodes, the energy consumption for data transmission is proportional to the square of the distance between the source node and the destination for the free space model. If all relay nodes are on the line from data source node to the Sink, the whole energy consumption for data transmission would be minimized. So, the degree of closeness of node to the shortest path (DCSP) should be used as one of energy optimized parameters. Consider

$$\mathrm{DCSP}(k) = \frac{d(i,\mathrm{Sink})}{d(i,k) + d(k,\mathrm{Sink})}, \tag{3}$$

where i denotes source node and k denotes its forwarding node, whose distance to Sink is less than i. Note that DCSP(k) attains its maximum (DCSP(k) = 1) when k lies on the line from i to Sink. The intuition behind the concept of DCSP is to make the data forwarding path not to deviate much from the shortest path between the current sender (i.e., the source of the data or any current forwarder) and the Sink.

3.2. Degree of Closeness of Node to Sink. In the process of data transmission, two data forwarding schemes: single-hop or multihop can be used within the communication range of the current sender. A forwarding scheme is said to be single-hop if each sensor in a data forwarding path can use at most one of its one-hop neighbors to forward a data packet toward its ultimate destination. A forwarding scheme is said to be multihop if the same data is forwarded through multiple neighbors of each sensor until the data reaches its destination. As can be seen from Figure 1, sensor i can send data to sensor j directly (single-hop) or through a relay node k (multihop). These two forwarding schemes consume different amounts of energy, which will be analyzed as follows.

Let us compute the energy consumption for sending data from sensor i to j using the previously mentioned two forwarding schemes for the free space model. Let

$$\begin{aligned} E_{\text{single-hop}} &= 2lE_{\text{elec}} + l\varepsilon_{\text{amp}} d^2(i,j), \\ E_{\text{multihop}} &= 4lE_{\text{elec}} + l\varepsilon_{\text{amp}} \left(d^2(i,k) + d^2(k,j)\right) \end{aligned} \tag{4}$$

be the energy consumption required to forward data from sensor i to j through single-hop and multihop forwarding schemes, respectively.

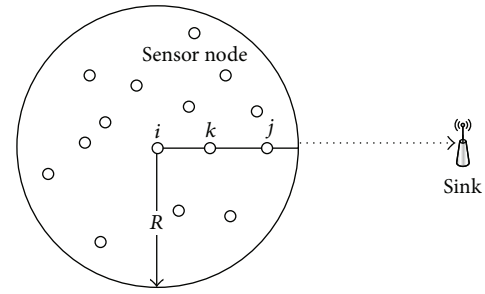


Figure 1: Data transmission in one-hop region.

Assume that sensor k lies on the line segment (i, j); that is, $d(i, j) = d(i, k) + d(k, j)$. If multihop short-range forwarding scheme is more energy efficient than single-hop, it implies $E_{\text{single-hop}} > E_{\text{multihop}}$. From (4), we can drive

$$d(i,k) \times d(k,j) > \frac{E_{\text{elec}}}{\varepsilon_{\text{amp}}}. \tag{5}$$

If sensor k is not on the line segment (i, j), we can drive

$$d^2(i,j) - d^2(i,k) - d^2(k,j) > \frac{2E_{\text{elec}}}{\varepsilon_{\text{amp}}}. \tag{6}$$

In this paper, we use the same parameter values for wireless communication hardware as Heinzelman used them in [19]. The numerical values of some parameters are as follows: R = 30 m, E_{elec} = 50 nJ/bit, and ε_{amp} = 10 pJ/bit/m^2. It is easy to check that (5) and (6) are not established. So, single-hop forwarding scheme is more energy efficient within the one-hop communication range of the source sensor or a current forwarder. In order to save energy, the neighbor node which is more close to Sink should be selected as next hop. The definition of degree of closeness of node to Sink (DCS) is

$$\mathrm{DCS}(k) = \frac{1/d(k,\mathrm{Sink})}{\sum_{j \in \mathrm{FN}(i)} 1/d(j,\mathrm{Sink})}, \tag{7}$$

where $d(k, \mathrm{Sink})$ represents distance from node k to Sink and FN(i) denotes the forwarding neighbor set of source node i.

3.3. Degree of Energy Balance. The existing energy-balancing routing algorithms are generally based on the residual energy of sensors to transform the selection of next hop, which is a kind of passive method of routing decision. When the data forwarding path is updated, the inequality of residual energy of sensors has already emerged. In our study, we adopt initiative routing adjustment strategy to predict the inequality of residual energy when selecting different forwarding neighbors as next hop and selecting the one with the highest degree of energy balance as next hop. In social sciences, there have been considerable efforts to define the so-called social welfare function to compare income welfare between space and time. In general, social welfare is a function of average and equality of an income population. In this paper, the energy unbalance

(EUB) of s set of sensors is computed using Atkinson welfare function according to [20]

$$\mathrm{EUB} = 1 - \left[\frac{1}{n}\sum_{i\in A}\left(\frac{E(i)}{\overline{E}}\right)^{1-\varepsilon}\right]^{1/(1-\varepsilon)} \quad (8)$$

EUB denotes energy unbalance of sensors in one-hop communication region A and n is the number of sensors in this region. $E(i)$ denotes the residual energy of sensor i and $\overline{E}$ the average residual energy. ε denotes the inequality aversion index, which takes values ranging from zero to infinity. The values of ε that are typically used include 1.5, 2.0, and 2.5.

To evaluate the alternative next hop, the sensor i calculates expected EUB of its local society consisting of its forwarding neighbors and itself according to the estimated residual energy of these sensors. After computing EUB for each alternative next hop, the forwarding neighbor is selected if the node gives the minimum EUB.

On the assumption that forwarding node k is selected as the next hop and the data is transmitted to it, the expected residual energy of sensor i is

$$E^{ik}(i) = E(i) - E_{\mathrm{Tx}}(l, d(i,k)). \quad (9)$$

The expected residual energy of sensor k after receiving and transmitting the same data (transmission distance approximates to R) is

$$E^{ik}(k) = E(k) - E_{\mathrm{Rx}}(l) - E_{\mathrm{Tx}}(l, R). \quad (10)$$

There is no change on residual energy of other neighbors not involved in data transmission, which is shown as

$$E^{ik}(j) = E(j). \quad (11)$$

Using the expected residual energy from (9), (10), and (11), sensor i can calculate the expected EUB for each decision k by (12), which is based on the Atkinson welfare function

$$\mathrm{EUB}^{ik} = 1 - \left[\frac{1}{n}\sum_{j\in N(i)+\{i\}}\left(\frac{E^{ik}(j)}{\overline{E^{ik}}}\right)^{1-\varepsilon}\right]^{1/(1-\varepsilon)}, \quad (12)$$

where

$$\overline{E^{ik}} = \frac{1}{n}\sum_{j\in N(i)+\{i\}} E^{ik}(j). \quad (13)$$

After sensor i has calculated EUB for each forwarding neighbor, the degree of energy balance (DEB) for selecting node k as next hop is calculated by

$$\mathrm{DEB}(k) = \frac{1/\mathrm{EUB}^{ik}}{\sum_{j\in \mathrm{FN}(i)} 1/\mathrm{EUB}^{ij}}. \quad (14)$$

4. Fuzzy-Logic-Based Routing

Fuzzy logic is used in this work as main implementation of perceptive reasoning. A fuzzy system basically consists of three parts: fuzzifier, fuzzy inference engine, and defuzzifier. The fuzzifier maps each crisp input value to the corresponding fuzzy sets and thus assigns it a truth value or degree of membership for each fuzzy set. The fuzzified values are processed by the inference engine, which consists of a rule base and various methods for inferring the rules. The rule base is simply a series of IF-THEN rules that relate the input fuzzy variables with the output fuzzy variables using linguistic variables, each of which is described by a fuzzy set. The defuzzifier performs defuzzification on the fuzzy solution space. That is, it finds a single crisp output value from the solution fuzzy space.

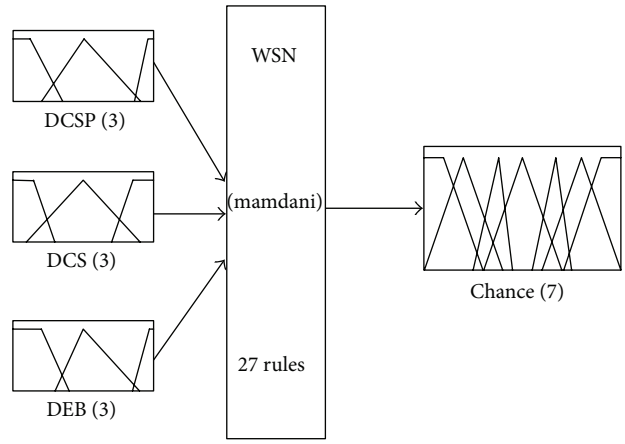


Figure 2: Model of fuzzy system.

The objective of our fuzzy-logic-based routing is to determine the energy optimized routing based on the parameters defined previously, such that the network lifetime is maximized. The fuzzy rule base has been tuned so as not only to minimize energy consumption but also to balance data traffic among sensor nodes effectively.

Figure 2 gives our fuzzy system model. Mamdani algorithm is used to realize fuzzy logic inference. The input fuzzy variables are degree of closeness of node to the shortest path (DCSP), degree of closeness of node to Sink (DCS), and degree of energy balance (DEB). The first two variables reflect the measure of energy efficiency for selecting one node as next hop, and the last variable shows the measure of energy balance for routing decision. The rule base consists of 27 (3^3) rules. There is a single output fuzzy variable, namely, chance, the defuzzified value of which determines the chance for one forwarding neighbor which has been selected as next hop.

Figure 3 displays details of the input and output fuzzy variables. The linguistic variables, used to represent DCSP and DCS, are divided into three levels: far, medium, and close, respectively, and there are three levels to represent DEB: poor, medium, and good, respectively. The output fuzzy variable to represent the node next hop election chance is divided into seven levels, which are very small, small, rather small, medium, rather large, large, and very large. The fuzzy rule base currently includes rules like the following: if DEB is good, DCSP is close, and DCS is close, the chance of the node to be selected as next hop is very large. The forwarding

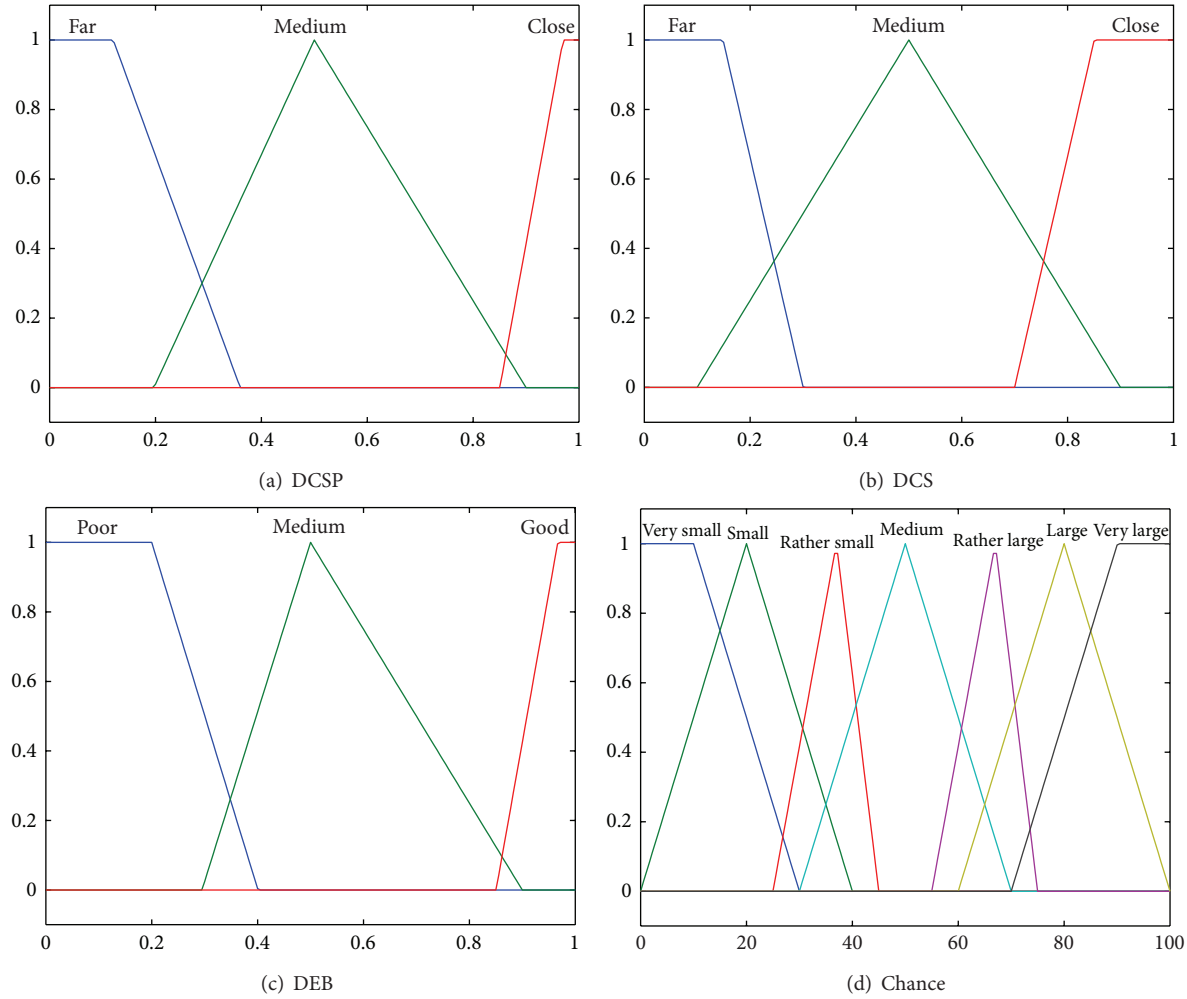


Figure 3: Fuzzy membership functions of input and output variables.

neighbors of the source sensor or a current forwarder are compared on the basis of chances, and the node with the maximum chance is then selected as the next forwarder. Mathematically, the crisp output domain value *chance*, from solution fuzzy region A, is given by

$$\text{Chance} = \frac{\sum_{i=1}^{27} W_i \mu_A(W_i)}{\sum_{i=1}^{27} \mu_A(W_i)}, \tag{15}$$

where W_i is the domain value corresponding to rule i and $\mu_A(W_i)$ is the predicate truth for that domain value.

5. Results and Discussion

In this section, we evaluate the performance of our proposed fuzzy-logic-based energy optimized routing (FLEOR) algorithm via MATLAB. We calculate the energy consumption for data transmission and reception. We define the network lifetime as the time when the residual energy of the first sensor node becomes zero, which is counted by round. We compare the performance of FLEOR algorithm with a predicting based distributed energy-balancing routing (PDEBR) [9], minimum transmission energy (MTE) routing [21], greedy perimeter stateless routing (GPSR) [22], and energy accounted minimum hop routing (EAMHR) [23] on the network lifetime, energy balance, and energy efficiency. In our simulations, sensor nodes are randomly and uniformly deployed over the square monitoring area. The Sink is placed at the outside of the monitoring area. Other simulation parameters are given in Table 1.

5.1. Network Lifetime. Figures 4, 5, and 6 give the network lifetime under different data generation patterns: uniform, random, and specific local area, respectively, when the number of sensors increases from 50 to 200. In our simulations, the data generation rate is set to be 0.25 for the random data generation pattern, which means that sensors generate data with probability of 0.25 in each round, while, for pattern of data generation from a local area, sensors located in a square area from (0, 0) to (50, 50) send data repeatedly.

As shown in Figure 4, FLEOR algorithm has extended the network lifetime under uniform data generation pattern, compared with PDEBR, EAMHR, GPSR, and MTE algorithms. GPSR and MTE algorithms make routing decisions

Table 1: Simulation parameters.

Parameter	Value
Network coverage/m^2	100×100
Number of sensors	50~200
Sink coordinates	(50, 110)
Initial energy/J	0.5
E_{elec}/(nJ·bit^{-1})	50
ε_{amp}/(pJ·bit^{-1}·m^{-2})	10
Data packet size/B	500
Control packet size/B	12
ε	2.5
Maximum transmission range/m	30

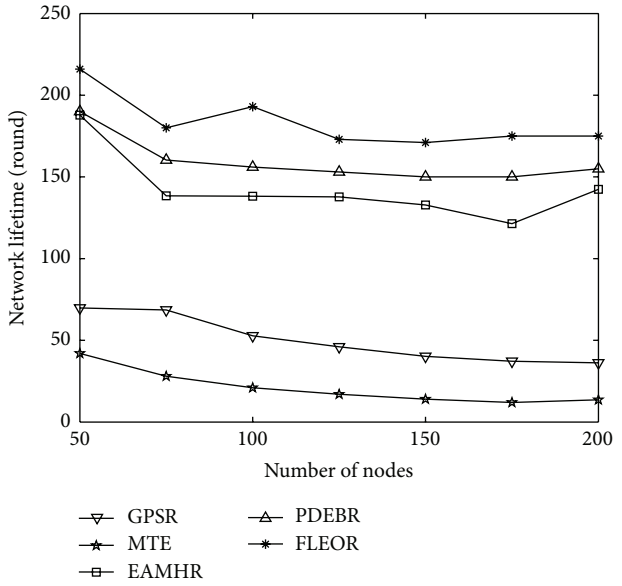


Figure 4: Network lifetime under uniform data generation pattern.

based on the location of neighbors and have no attempt on energy balance, resulting in short network lifetime. The node closest to Sink especially in MTE will relay the data of the whole network, resulting in quick energy consumption and the shortest network lifetime. EAMHR selects the node with the most residual energy as next hop from forwarding neighbors, which has achieved combination of energy efficiency and energy balance to a certain extent and prolonged network lifetime compared with GPSR and MTE. PDEBR predicts mean square deviation of residual energy of neighbors and selects the node with the minimum value as next hop from front neighbors. It has achieved the distributed local energy balance and has longer network lifetime compared with EAMHR, GPSR, and MTE. FLEOR combines energy efficiency and energy balance together through fuzzy logic. Compared with PDEBR and EAMHR, FLEOR has extended network lifetime further, which means that FLEOR can achieve a better combination of energy efficiency and energy balance.

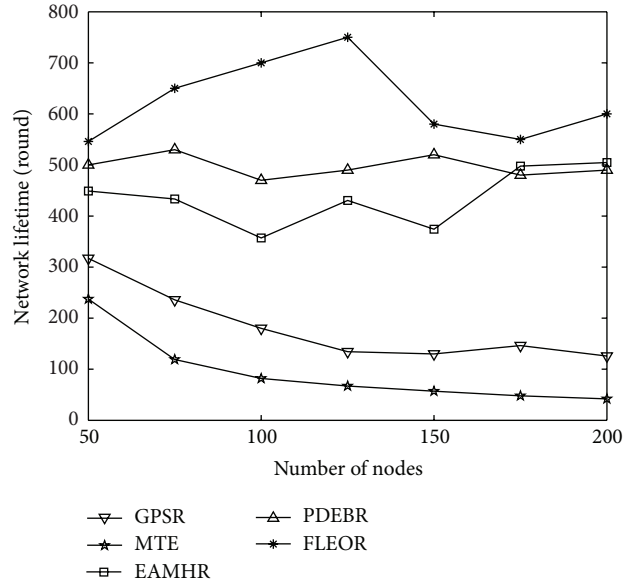


Figure 5: Network lifetime under random data generation pattern.

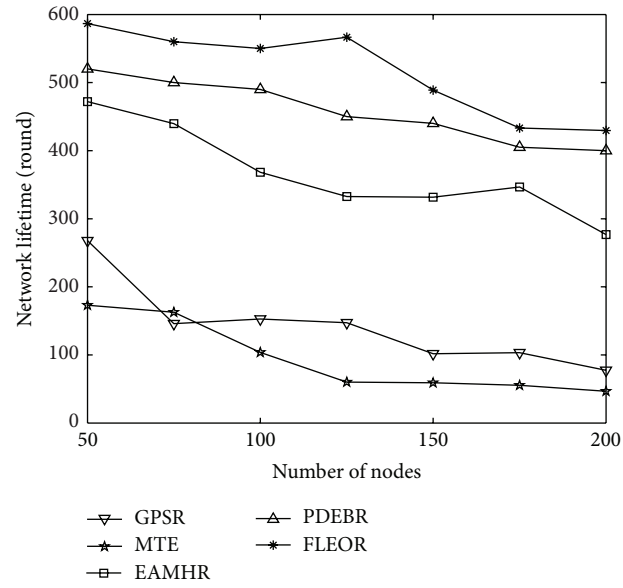


Figure 6: Network lifetime under data generation pattern from a local area.

Figures 5 and 6 show that FLEOR has significant advantages on network lifetime under random data generation pattern and local area data generation pattern, compared with other algorithms. With these results, we can say that FLEOR is adaptable to different data generation patterns and is more suitable for real network design requirements.

5.2. Energy Balance and Energy Efficiency. Figure 7 gives the average residual energy of nodes under uniform data generation pattern when the first node becomes incapacitated. In GPSR and MTE, there is no consideration on energy balance for routing decision. So, there are many nodes with more

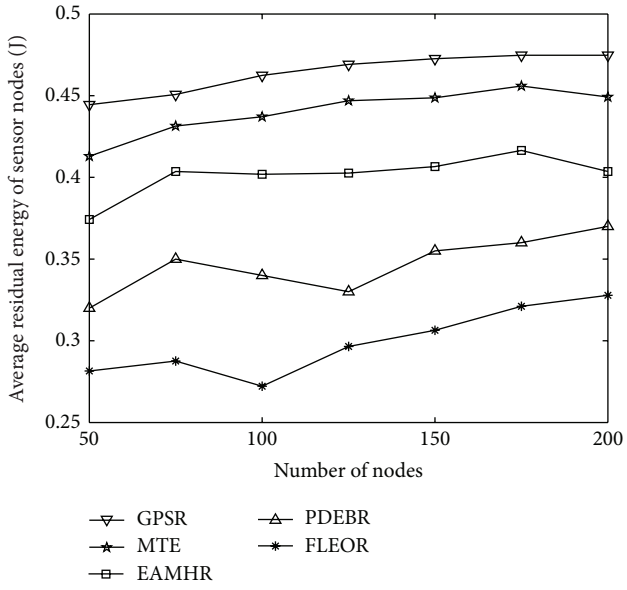


Figure 7: Average residual energy of sensor nodes.

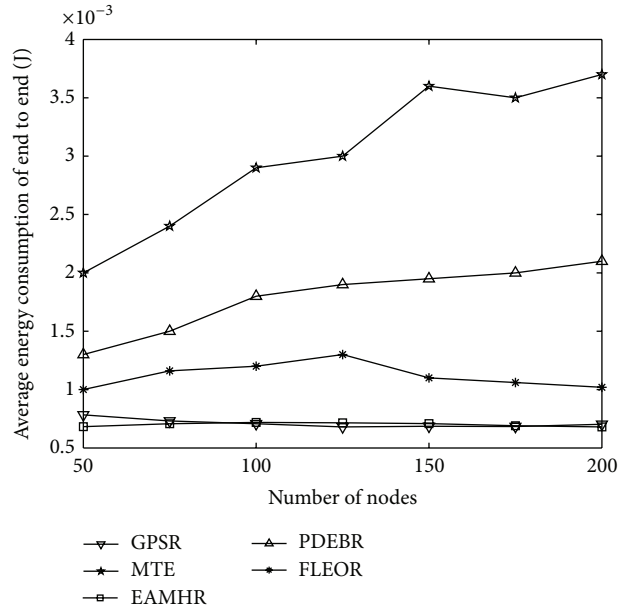


Figure 8: Average energy consumption of end to end.

residual energy when the first node becomes incapacitated, compared with FLEOR, PDEBR, and EAMHR.

Figure 8 gives the average energy consumption of end to end for different algorithms under uniform data generation pattern. From this figure, we can see that the average energy consumption of end to end in GPSR and EAMHR is close and maintained at a low level, which shows a good performance on energy efficiency. In PDEBR, the node with the minimum predicted mean square deviation of residual energy is selected as next hop, although it is near the sending node and far from the Sink. So, the average energy consumption of end to end in PDEBR is larger than GPSR, MTE, and FLEOR. In MTE, the multihop short-range forwarding scheme is used to transmit data, which has been proved to be less energy efficient within the communication range of the current forwarder. So, the average energy consumption of end to end in MTE is the most and shows the upward trend with the increase of network size. In FLEOR, the energy balance of nodes is considered preferentially when making routing decisions. As a result, the average energy consumption of end to end in FLEOR is higher than GPSR and EAMHR. At the same time, FLEOR has achieved energy efficiency, leading to less average energy consumption of end to end and restraint of its rising trend with the increase of the number of nodes, compared with MTE and PDEBR.

6. Conclusions

In this paper, we have designed three energy optimized parameters, such as the degree of closeness of node to the shortest path, degree of closeness of node to Sink, and degree of energy balance, and put these parameters into fuzzy logic system. The fuzzy-logic-based routing algorithm is proposed to realize energy optimized, multiparameter, and fuzzy routing decision. Simulation results show that the algorithm extends the network lifetime effectively compared with similar algorithms for different dada generation patterns and has a good performance in terms of energy balance and energy efficiency.

Our future work will focus on the applications for multimedia. While achieving optimized energy consumption of the whole network, the Qos, such as bandwidth, latency, and packet loss rate will be considered to meet the requirements of specific applications.

Acknowledgments

Financial support for this work is provided by the National Natural Science Foundation of China (no. 50904070, 51274204); the Fundamental Research Funds for the Central Universities (no. JGD101671) and the Fundamental Research Funds for the Central Universities (no. 2013RC11), Qing Lan Project, 333 talent project and six talent tops of jiangsu province are gratefully acknowledged.

References

[1] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 2, pp. 551–591, 2013.

[2] S. K. Zhang, Y. Sun, J. X. Fan, and H. Huang, "Cooperative data processing algorithm based on mobile agent in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 182561, 9 pages, 2012.

[3] F. Ren, J. Zhang, T. He, C. Lin, and S. K. D. Ren, "EBRP: Energy-balanced routing protocol for data gathering in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 12, pp. 2108–2125, 2011.

[4] C. Efthymiou, S. Nikoletseas, and J. Rolim, "Energy balanced data propagation in wireless sensor networks," *Wireless Networks*, vol. 12, no. 6, pp. 691–707, 2006.

[5] H. Zhang and H. Shen, "Balancing energy consumption to maximize network lifetime in data-gathering sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, pp. 1526–1539, 2009.

[6] S. Wu and K. S. Candan, "Power-aware single- and multipath geographic routing in sensor networks," *Ad Hoc Networks*, vol. 5, no. 7, pp. 974–997, 2007.

[7] S. J. Baek and G. de Veciana, "Spatial energy balancing through proactive multipath routing in wireless multihop networks," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 93–104, 2007.

[8] G.-Y. Li, Y. Cao, H. Feng, and W.-H. Wu, "Residual energy scheming based energy equilibrium routing protocol for wireless sensor network," *Journal of Central South University*, vol. 40, no. 6, pp. 1642–1648, 2009.

[9] X. W. Liu, F. Xue, and Y. Li, "Distributed energy balancing routing algorithm in wireless sensor networks," *Computer Science*, vol. 37, no. 1, pp. 122–125, 2010.

[10] Y.-H. Zhu, W.-D. Wu, J. Pan, and Y.-P. Tang, "An energy-efficient data gathering algorithm to prolong lifetime of wireless sensor networks," *Computer Communications*, vol. 33, no. 5, pp. 639–647, 2010.

[11] T. Haider and M. Yusuf, "A fuzzy approach to energy optimized routing for wireless sensor networks," *International Arab Journal of Information Technology*, vol. 6, no. 2, pp. 179–185, 2009.

[12] I. S. Alshawi, L. S. Yan, W. Pan, and B. Luo, "Lifetime enhancement in wireless sensor networks using fuzzy approach and a-star algorithm," *IEEE Sensor Journal*, vol. 12, no. 10, pp. 3010–3018, 2012.

[13] 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.

[14] I. Gupta, D. Riordan, and S. Sampalli, "Cluster-head election using fuzzy logic for wireless sensor networks," in *Proceedings of the 3rd Annual Communication Networks and Services Research Conference*, pp. 255–260, Canada, May 2005.

[15] J.-M. Kim, S.-H. Park, Y.-J. Han, and T.-M. Chung, "CHEF: Cluster head election mechanism using fuzzy logic in wireless sensor networks," in *Proceedings of the 10th International Conference on Advanced Communication Technology*, pp. 654–659, Republic of Korea, February 2008.

[16] H. Taheria, P. Neamatollahia, O. M. Younisb, and S. Naghibzadehc, "An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1469–1481, 2012.

[17] J. S. Lee and W. L. Cheng, "Fuzzy-logic-based clustering approach for wireless sensor networks using energy predication," *IEEE Sensors Journal*, vol. 12, no. 9, pp. 2891–2897, 2012.

[18] S. J. Dastgheib, H. Oulia, M. R. S. Ghassami, and S. J. Mirabedini, "A new method for flat routing in wireless sensor networks using fuzzy logic," in *Proceedings of the International Conference on Computer Science and Network Technology (ICCSNT '11)*, pp. 2112–2116, China, December 2011.

[19] W. R. Heinzelman, *Application-Specific Protocol Architectures for Wireless Networks*, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2000.

[20] A. B. Atkinson, "On the measurement of inequality," *Journal of Economic Theory*, vol. 2, no. 3, pp. 244–263, 1970.

[21] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Siences (HICSS '00)*, pp. 1–10, January 2000.

[22] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.

[23] K.-H. Han, Y.-B. Ko, and J.-H. Kim, "A novel gradient approach for efficient data dissemination in wireless sensor networks," in *Proceedings of the IEEE 60th Vehicular Technology Conference (VTC '04)*, pp. 2979–2983, September 2004.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 592129, 10 pages
http://dx.doi.org/10.1155/2013/592129

Research Article

Research on Vehicle Automatically Tracking Mechanism in VANET

Lin Wang,[1] Shukui Zhang,[1,2] Xiaoning Wang,[1] and Yang Zhang[3]

[1] *Institute of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[2] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjin 210093, China*
[3] *Institute of Electronics & Electrical Engineering, University of Glasgow, Glasgow G12 8QQ, UK*

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

Received 10 April 2013; Accepted 10 July 2013

Academic Editor: Hongli Xu

The intelligent vehicle is a complex system equipped with advanced technologies such as the artificial intelligence, automatic control, and computer, communication. It is a combination of multiple academic subjects and the latest technologies representing the developing tendency of future automobile technology and attracts more and more attention. In this paper, we made some useful explorations in the fields of intelligent vehicle control technology and obstacle avoidance, and a deep research is carried out about the distance of vehicle, vehicle tracking, vehicle lane changing and intersections obstacle avoidance and communication protocols, and some innovative ideas are proposed during the research. By using VANET, some predictable status of tracing and tracking of intelligent vehicle technology was researched, and the communication protocols between two vehicles, the safe spacing algorithm, the method of computing the actual distance at corners and straights, and the trajectory of lane change were designed. In addition, a series of vehicle tracing and tracking technologies, such as without knowing the road conditions, homeostatic mechanism, keeping a safe spacing to the target vehicle, and adjusting its own speed to track the target vehicle smoothly to the destination by comparing the actual distance and safe spacing between vehicles, were discussed here.

1. Introduction

The number of vehicles owned by people is rapidly growing with the development of economy and society. The safety problem in transportation is increasingly outstanding. It brings a serious threat to humans' life and property. As we all known, safe-driving is always one of the most important topics in vehicle engineering. In order to reduce traffic accidents, the intelligent vehicle emerges as the times require. It is a complex system equipped with advanced technologies such as the artificial intelligence, automatic control, computer, and communication. It is a combination of multiple academic subjects and latest technologies, representing the developing tendency of future automobile technology. Therefore, it attracts more and more attention. With the fast development in ad hoc wireless communications and vehicular technology, it is foreseeable that, in the near future, traffic information will be collected and disseminated in real time by mobile sensors instead of fixed sensors used in the current infrastructure-based traffic information systems. A distributed network of vehicles such as a vehicular ad hoc network (VANET) can easily turn into an infrastructure-less self-organizing traffic information system, where any vehicle can participate in collecting and reporting useful traffic information such as section travel time, flow rate, and density. Disseminating traffic information relies on broadcasting protocols [1]. In-network data aggregation is a useful technique to reduce redundant data and to improve communication efficiency. Traditional data aggregation schemes for wireless sensor networks usually rely on a fixed routing structure to ensure that data can be aggregated at certain sensor nodes [2].

Recent years have witnessed the growing popularity of sensor and sensor-network technologies, supporting important practical applications. One of the fundamental issues is how to accurately locate a user with few labeled data in

a wireless sensor network, where a major difficulty arises from the need to label large quantities of user location data, which in turn requires knowledge about the locations of signal trans mitters or access points [3]. Donghyun et al. considers the problem of computing the optimal trajectories of multiple mobile elements (e.g., robots, vehicles, etc.) to minimize data collection latency in WSNs [4].

A new category of intelligent sensor network applications emerges where motion is a fundamental characteristic of the system under consideration. In such applications, sensors are attached to vehicles or people that move around large geographic areas. For instance, in mission critical applications of WSNs, sinks can be associated to first responders. In such scenarios, reliable data dissemination of events is very important, as well as the efficiency in handling the mobility of both sinks and event sources. For this kind of applications, reliability means real-time data delivery with a high data delivery ratio. In their article, Erman et al. proposes a virtual infrastructure and a data dissemination protocol exploiting this infrastructure, which considers dynamic conditions of multiple sinks and sources [5]. WSNs have been increasingly available for critical applications such as security surveillance and environmental monitoring. An important performance measure of such applications is sensing coverage that characterizes how well a sensing field is monitored by a network [6]. Wireless has become one of the most pervasive core technology enablers for a diverse variety of computing and communications applications ranging from 3G/4G cellular devices, broadband access, indoor Wi-Fi networks, and vehicle-to-vehicle (V2V) systems to embedded sensor and RFID applications [7].

A large class of WSN applications involve a set of isolated urban areas covered by sensor nodes monitoring environmental parameters. Mobile sinks mounted upon urban vehicles with fixed trajectories (e.g., buses) provide the ideal infrastructure to effectively retrieve sensory data from such isolated WSN fields [8]. Jorge et al. [9] presents control and coordination algorithms for groups of vehicles. The focus is on autonomous vehicle networks performing distributed sensing tasks where each vehicle plays the role of a mobile tunable sensor. The author propose gradient descent algorithms for a class of utility functions which encode optimal coverage and sensing policies. Kwok and Martínez's work [10] includes incorporating nonholonomic vehicle dynamics into the convergence analysis in order to provide a more practical coverage scenario for implementation in a physical testbed. Since the convergence of the algorithms is only guaranteed to local optima, we are also working on extensions that help us find more optimal coverage configurations.

In this paper, we have made some positive and useful explorations in the fields of intelligent vehicle control technology and obstacle avoidance. On the basis of the current situation and the vital supporting techniques of these fields, a deep research is carried out about the distance of vehicle, vehicle tracking, vehicle lane changing, and intersection obstacle avoidance and communication protocols. Moreover, some innovative ideas are proposed during the research. Finally, a brief summary and an outlook are made about this paper.

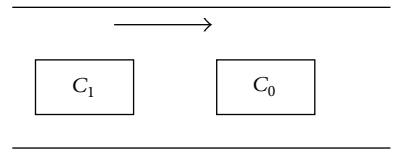


Figure 1: The longitudinal vehicle.

2. The Design of Vehicle Tracing Algorithm

Many related control strategies are needed to intervene with the vehicles when they are tracked. The algorithm of control is discussed as follows.

2.1. Vehicle-to-Vehicle Safe Spacing and Minimum Safe Spacing. To avoid collision, the spacing between two vehicles must be greater than the safe spacing. Safe spacing, the distance of two vehicles for driving safely, means vehicles running in the same direction must keep the distance between one and another for the safety of traffic. Minimum safe spacing or critical safe spacing indicates the minimum intervehicle distance for safety. The capacity of roads is proportional to the velocity and is inverse proportional to the spacing of vehicles which reveals the fact that increasing the safe spacing blindly will conversely lower the traffic capacity. To keep the capacity of roads in accordance with road safety, people pay much attention to the minimum safe spacing.

The vehicle, in the motion, travels in the longitudinal direction most of time; that is, there is no transverse velocity and acceleration. So it is totally useful to study the longitudinal inter-vehicle safe spacing. As shown in Figure 1, vehicle C_0 and vehicle C_1 are running on the same lane and vehicle C_0 is ahead of vehicle C_1. Computing the inter-vehicle safe spacing needs knowing the velocity and acceleration of both vehicles first. As it is difficult to capture both velocity and acceleration precisely, the problem demands simplification. Supposing that both of the vehicles are in condition of uniform and linear motion, the safe spacing during time 0 and t_m can be computed by

$$\mathrm{MSS} = \max\left(\left(v_1 - v_0\right)t, 0\right), \quad t \in \left[0, t_m\right]. \tag{1}$$

v_0 and v_1 are the longitudinal velocity of vehicle C_0 and vehicle C_1.

As both vehicles make uniform and linear motion, the relative velocity is always a constant, $v = v_0 - v_1 = \text{const}$, which deduces MSS as the following rules:

$$\mathrm{MSS} = \begin{cases} \left(v_1 - v_0\right)t_m & v_1 - v_0 \geq 0 \\ 0 & v_1 - v_0 < 0. \end{cases} \tag{2}$$

In this way, there are time and minimum safe spacing. As we set different values to these parameters, series of situations and solutions are acquired as follows.

(1) When relative velocity $v > 0$, inter-vehicle safe spacing will linearly increase by time. As time increases, the distance between two vehicles is getting shorter. To compensate the distance that vehicle C_1 travels

more than vehicle C_0 before they collide, the MSS must be increased, so MSS = $v * t_m$.

(2) When relative velocity $v < 0$, inter-vehicle distance is increasing by time, so MSS = 0.

(3) When relative velocity $v = 0$, the MSS can be set to any value approximating to 0 in theory.

The conclusions above are based on the analysis merely considering the longitudinal minimum safe spacing preserved in advance; however, the actual distance between vehicles can not be 0. Besides the factors as relative velocity, relative acceleration and time before collision during driving, the initial velocity deserves consideration as well. The safe spacing (SS) is given by

$$\text{SS} = f(v_1) + \text{MSS}. \tag{3}$$

According to (3), inter-vehicle safe spacing is a sum of critical safe spacing and a revised value. Researchers from home and abroad have put forward several models; one of them is a safe spacing model based on time interval between vehicles. The time interval (T) is defined as

$$T = \frac{R}{V}. \tag{4}$$

R means the relative distance between two vehicles and V means the velocity of the vehicle behind. The time interval contains the information of velocity and distance so it can demonstrate the danger level of collision. As regulations in China specify "Motor vehicles traveling on the highway must keep enough distance from the one ahead at the same lane. Under normal circumstances, when driving at 100 kilometers per hour, the safe distance is over 100 meters, when driving at 70 kilometers per hour, the safe distance is over 70 meters". So it can be inferred that the time interval is about 3 seconds on the highway. Equation (3) can be modified as follows:

$$\text{SS} = 3v_1 + \text{MSS}. \tag{5}$$

2.2. Queue and Dequeue of Vehicles. The capacity of roads is proportional to the velocity of vehicles. Obviously, vehicles with great disparity of velocity driving at the same lane will lower the traffic capacity. Therefore, classifying vehicles with different velocities into different queues and putting these queues on distinctive lanes can reduce the incidence of traffic accidents to some extent. As the road sign on the highway indicates, under normal circumstances, the maximum speed is 120 kilometers per hour and the minimum speed is 60 kilometers per hour. Small passenger cars traveling on the highway shall not exceed 120 kilometers per hour, other motor vehicles shall not exceed 100 kilometers per hour, and the motorcycle shall not exceed 80 kilometers per hour. According to the regulation, there are two plans of grouping; one is based on the type of vehicles and another is based on the velocity.

Classifying vehicles with types can avoid the situation of mixing up cars and trucks and make vehicles in the same class run at the right lane. This method secures road safety as it separates motor vehicles from nonmotor vehicles. However, it is unnecessary to subdivide the vehicular road owing to the uncertain quantity of vehicles in various kinds. On the contrary, if we do so, the road capacity will probably descend. So we place emphasis on the queue of vehicles by velocities. According to the discussion above, the velocity of motor vehicles ranges from 60 kilometers per hour to 120 kilometers per hour. Supposing that the road contains at least 3 lanes, we can divide vehicles into 3 kinds of queues by velocities.

Vehicle queue 1 (C_Queue[0]), the velocity is lower than 80 kilometers per hour (km/h);

Vehicle queue 2 (C_Queue[1]), the velocity is between 80 km/h and 100 km/h;

Vehicle queue 3 (C_Queue[2]), the velocity is higher than 100 km/h.

Vehicles in specific queues run at different lanes and vehicles in the same lane are put in descending order by velocities. There are some exceptional cases in the traffic requesting disposal, such as a certain vehicle that needs changing lane or overtaking; then it comes up with the problem of regrouping the queue. During the course of regrouping, some vehicles need to leave the old queue and join a new queue. When a certain vehicle is leaving the queue, its position must be detected. When it is the head of the queue, its next vehicle is obligated to be the leader; when it is in the middle of the queue, its next one must follow the front of the leaving one; when it is the tail, it can leave the queue without any trouble. We can define a variable of Boolean type to trace the vehicle whether it is in a certain queue or not.

2.3. Vehicle Tracking. Vehicle tracking involves the mutual interaction of multiple vehicles, so the vehicle tracking model is among the interaction model. Nowadays, the interaction between intelligent vehicles is in high frequency and the complexity varies from system to system. However, the vehicle tracking model is the one much simple and deep-going. Several conditions that can track vehicles are discussed as follows.

(1) The destination of the front vehicle is the same as the destination of the current vehicle.

(2) The velocity of the front vehicle approximates the velocity of the current vehicle.

(3) Traffic gets blocked or the lane is unchangeable and nonovertaking.

There are 3 kinds of road sections on the highway, which are straights, corners, and S-corners. As the S-corner consists of multiple corners, we simply regard the S-corner as corners. We merely take straights and corners into consideration and suppose that changing lanes is forbidden while tracking. All the vehicles are at the same lane when they are tracked, so we need to control the longitudinal direction of the vehicle rather than the transverse one to keep the velocity and safe spacing.

Most currently used vehicle tracking algorithms are safe spacing algorithms which adjust the velocity of the current vehicle to track the front vehicle by capturing images based

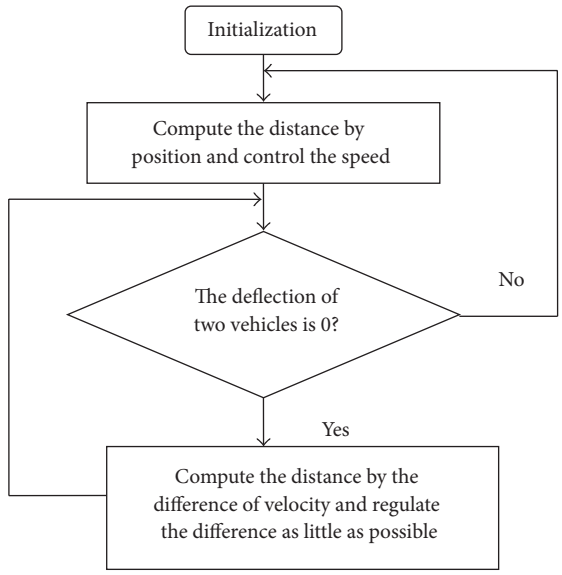


Figure 2: The flow diagram of the vehicle tracking algorithm.

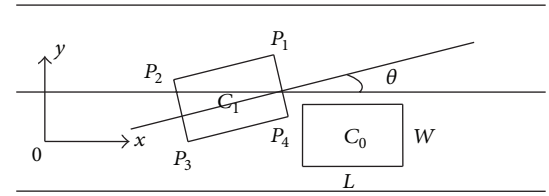


Figure 3: The sketch map of lane change.

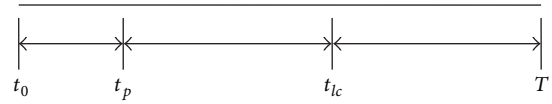


Figure 4: Time points of lane change.

on the visual perception technologies that record the relative position to the front vehicle. However, the complexity of the vehicle tracking algorithm on the basis of vehicle-to-vehicle and vehicle-to-roadside communications sharply diminishes compared with the visual tracking algorithm (Figure 2). After obtaining the positioning information of the front vehicle by communication, the current vehicle computes the vector difference which equals the distance between them to hold the safe spacing. It is relatively simple to apply vehicle tracking on the straight. Vehicles exchange information about positioning; velocities and so forth by communicating with each other and computing the distance L between them to keep the safe spacing. By comparing the distance with the safe spacing and regulating the velocity, vehicle tracking can be easily completed.

In the situation of corners, computing the distance of two vehicles by positioning is not suitable for use because the result computed now is not the actual length of the spacing but the linear distance between them. When vehicles are at the corner (the time that a deflection emerges), the spacing can be computed by knowing the velocities of both vehicles. In other words, the current vehicle captures the velocity of the front vehicle and deducts it with its own velocity, then multiplies the difference by the time during the corner. The distance L is given by

$$L' = L + \left(\left|v_2\right| - \left|v_1\right|\right)\Delta t, \tag{6}$$

where L is actual distance between two vehicles, v_1 is velocity of the front vehicle, v_2 is velocity of the front vehicle, and Δt is the time during the corner. When vehicles are at the corner, they regulate their own speed close to the velocities of other vehicles obtained by interaction to keep the safe spacing and tracking.

2.4. Lane Change of Vehicles. When a vehicle runs slower than its rear one all the time, overtaking occurs definitely with the case of lane change. Drivers need proficient skills and adequate experience when changing the lane; otherwise accidents will happen. At the very beginning of lane change, rear-end collisions are highly possible. When the distance between two vehicles is not enough, angle collisions take place likely. Figure 3 shows the sketch map of lane change. All the parameters are marked as shown in Figure 3, where C_0 and C_1 is Vehicle, L is length of the vehicle, W is width of the vehicle, θ is angle between the symmetry axis and x-axis, and $P_1 \sim P_4$ is 4 angles of the vehicle. Establishing coordinate system as Figure 3 shows, the longitudinal acceleration, longitudinal velocity, longitudinal position, transverse acceleration, transverse velocity, and transverse position of vehicle C_i are defined as a_{ix}, v_{ix}, x_i, a_{iy}, v_{iy}, y_i, $i \in \{0, 1\}$ and x_i, y_i means the left and front angle, such as P_1.

Lane change can be divided into several time periods. Time points of lane change are marked on the Figure 4, specifically described as shown in Figure 4, where t_0 is moment of starting lane change, t_p is moment of collision, t_{lc} is moment of finishing lane change, and T is a random time period after changing lane. Traffic in real life can be much complex, so we simplify the problem, supposing that vehicle C_0 does not have transverse velocity and acceleration; that is, $a_{oy} = 0$, $v_{oy} = 0$. Analyzing the relationship between the positions of two vehicles, the conditions of collision avoidance are given as follows:

$$x_0(t) > x_1(t) + L_0 + W_1 \sin(\theta(t)) \quad t \in \left[0, t_p\right]. \tag{7}$$

L_0 means the length of vehicle C_0, $\theta(t)$ means the angle between vehicle C_0 and x-axis at time t, and W_1 means the width of vehicle C_0

$$\theta(t) = tg^{-1}\frac{\partial y_1(t)}{\partial x_1(t)} = tg^{-1}\frac{\partial y_1(t)/\partial t}{\partial x_1(t)/\partial t} = tg^{-1}\frac{v_{y1}(t)}{v_{x1}(t)} \tag{8}$$

and setting

$$S(t) = x_0(t) - \left[x_1(t) + L_0 + W_1 \sin(\theta(t))\right]. \tag{9}$$

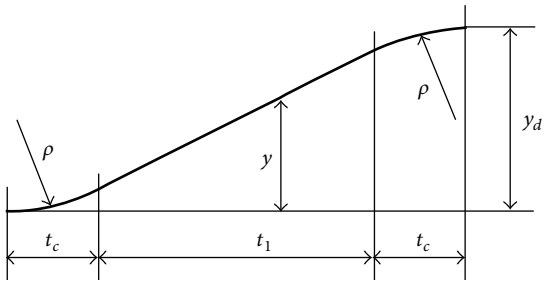


Figure 5: The initial segment and the final segment of the lane change trajectory.

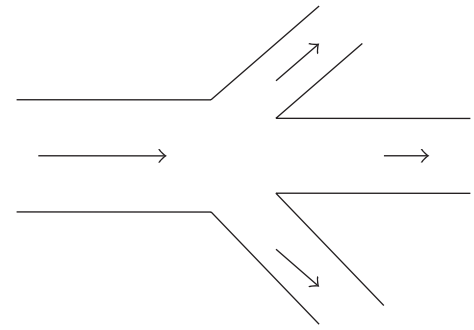


Figure 6: A branch of a road.

To avoid the collision, the result of (9) is greater than 0; that is,

$$S(y) = \left(S^*(0) + \int_0^t \int_0^\sigma \left(a_{0x}(t) - a_{1x}(t)\right) dt\, d\sigma + \left(v_{0x}(0) - v_{1x}(0)\right) t\right) > 0, \quad t \in \left[0, t_p\right], \tag{10}$$

$$S^*(0) = x_0(0) - L_0 - x_1(0). \tag{11}$$

In order to find the minimum initial distance to avoid collisions of vehicle C_0 and C_1, we set (11) to 0 and that means the two vehicles are end to end at the early time of lane change which can be

$$\text{MSS} = \min\left(\int_0^t \int_0^\sigma \left(a_{0x}(t) - a_{1x}(t)\right) dt\, d\sigma + \left(v_{0x}(0) - v_{1x}(0)\right) t\right) \quad t \in \left[0, t_p\right]. \tag{12}$$

Minimum safe spacing (MSS) between the two vehicles is determined by relative longitudinal acceleration and initial velocities of the two vehicles and the time duration before collision. Lane change can have a variety of trajectories. This paper introduces the one based on the trajectory of the arc. Assuming that the vehicle lane change is divided into three stages: the initial stage of the vehicle to generate a transverse acceleration, the intermediate stage of the vehicle running straight at a constant speed, and the final stage of the vehicle to generate a transverse deceleration (the value is the same as the one in the initial stage of acceleration), as shown in Figure 5, the initial segment and the final segment of the lane change trajectory constitutes by the arc of a circle when the middle part uses straight line. The curvature of the arc radius is ρ

$$\rho = \frac{v_x^2}{a_y}. \tag{13}$$

v_x^2 is the longitudinal velocity of the vehicle and a_y is the transverse acceleration of the vehicle.

Analyzing Figure 5 can get the following:

$$y_d = 2\rho\left(1 - \cos\left(\frac{v_x t_c}{\rho}\right)\right) + v_x t_l \sin\left(\frac{v_x t_c}{\rho}\right). \tag{14}$$

The time of lane change is the sum of the time passing the both of the arc and the straight line; that is, $t_{lc} = 2t_c + t_l$

$$t_{lc} = 2t_c + \frac{\left(y_d - 2\rho + 2\rho\cos\left(v_x t_c/\rho\right)\right)}{v_x \sin\left(v_x t_c/\rho\right)}. \tag{15}$$

Thus it can be calculated as follows:

$$\frac{dt_{lc}}{dt_c} = -\frac{\left[y_d - 2\rho + 2\rho\cos\left(v_x t_c/\rho\right)\right]\cos\left(v_x t_c/\rho\right)}{\rho\sin^2\left(v_x t_c/\rho\right)}. \tag{16}$$

Setting $dt_{lc}/dt_c = 0$, t_c can be given as

$$t_c = \frac{\rho}{v_x}\cos^{-1}\left(1 - \frac{y_d}{2\rho}\right). \tag{17}$$

The analysis shows that t_{lc} is the function of t_c. To calculate the minimum lane change time, we take the derivative of (15).

On analysis of the above equations, to make the lane change requires the least time, we can set $t_1 = 0$; that is, the time on the straight line is 0. The trajectory now consists of two sections of the same curvature arc, so

$$y_d(t) = \begin{cases} \rho\left(1 - \cos\left(\dfrac{v_x t}{\rho}\right)\right) & 0 \le t \le t_c \\ \rho\left[1 + \cos\left(\dfrac{v_x\left(t_{lc} - t\right)}{\rho}\right) - 2\cos\left(\dfrac{v_x t}{\rho}\right)\right] & t_c \le t \le t_{lc}. \end{cases} \tag{18}$$

The lane change time can be calculated by

$$t_{lc} = \frac{2\rho}{v_x}\cos^{-1}\left(1 - \frac{y_d}{2\rho}\right). \tag{19}$$

2.5. Selection of the Road. When driving on the highway, drivers always meet the situation of vehicle shunt as shown in Figure 6. Drivers need to know the information provided by the road signs to select the road and determine the next direction of the vehicle. In this case, drivers who are not familiar with the roads often spend much time considering the problem, which inevitably reduces the traffic capacity of the road. So we assume that the road signs and driving vehicles

can communicate with each other. When the distance between road signs and vehicles falls to S, they automatically establish communication. Vehicles automatically select the road according to the transmitted information to reach their destination. When there are N ($N > 1$) roads to reach their destinations, we can use the satellite sensors and high-precision digital map to choose the shortest one from the routes.

The greater the distance S when the road sign and the vehicle establish communication is, the longer time to choose the road will be. However, S cannot be infinitely large, so we limit S with a minimum distance to meet all the velocities of vehicles. Assuming that the maximum time from establishing to finishing communication is t_m. After learning the route, some vehicles need to change the lane to enter the right road.

Vehicles shall be under maximum speed of 120 kilometers per hour (33.3 meters per second). To let all the vehicles successfully choose the right road, the minimum distance S is $S_{\min}$

$$S_{\min} = 33.3t_m + v_x t_{lc}. \tag{20}$$

t_{lc} is the time of lane change and v_x is the longitudinal velocity of the vehicle. t_{lc} can be computed by the following:

$$t_{lc} = \frac{2\rho}{v_x}\cos^{-1}\left(1 - \frac{y_d}{2\rho}\right). \tag{21}$$

So (20) can be replaced by

$$S_{\min} = 33.3t_m + 2\rho\cos^{-1}\left(1 - \frac{y_d}{2\rho}\right). \tag{22}$$

Nowadays, there are a variety of communication tools such as WIFI, Zigbee, Bluetooth, and infrared. After acquiring the minimum communication distance, we can choose the most suitable tool for inter-vehicle communication by the comprehensive analysis of the performance and price of those tools.

2.6. The Traffic Junction's Obstacle Avoidance Algorithm. Model cars, under the two-dimensional traffic environment, mainly need to care about the problem of the traffic junction's obstacle avoidance. The intersection can be generally divided into "T" type intersection and Cross-type intersection. "T" type intersection can be seen as part of the Cross-type intersection, so we can solve the "T" type intersection corresponding to the Cross-type intersection.

As in Figure 7, "○" signs are the highly possible positions in which two vehicles traveling along two different directions may collide. In order to prevent collisions between vehicles, the velocity of the vehicle needs control so that the vehicle can pass the road crossing point in an order which can effectively solve the problem of the traffic junction's obstacle avoidance.

Establish a certain area as the intersection point management control region (region III and within the region shown in Figure 8) around the road crossing point (i.e., the "○" symbols). Form a queue of sequent vehicles driving into the region and control the vehicles at low speed. When the vehicle is in the most forward position in the queue, it can be permitted to travel through the crossing point at full speed. Arrange a security area in the region (region I in Figure 8) which allows only one vehicle at the same time. Outside the security area is a parking area (region II in Figure 8) in which vehicles driving through can immediately stop when there is a vehicle in region I.

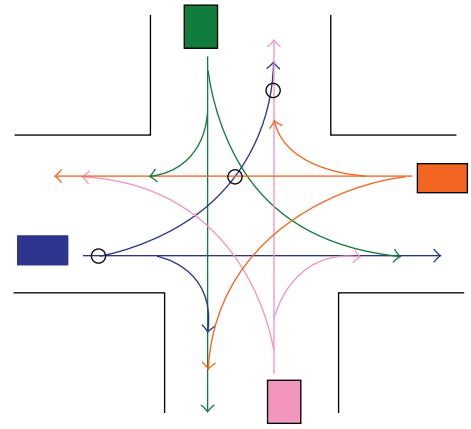


Figure 7: "T" type and Cross-type intersection.

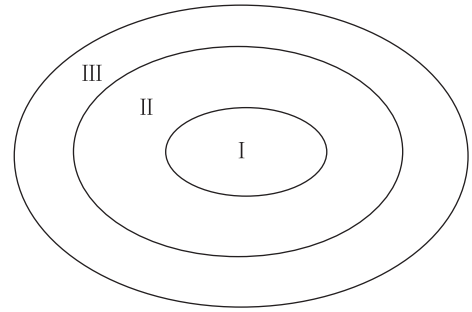


Figure 8: The disjunctive security regions.

When a vehicle enters the region I, the vehicle sends instructions to inform other vehicles that the intersection resource is occupied. Other vehicles receive this instruction and immediately reduce their speed to ensure they can stop outside region I. When the vehicle passes the crossing point, it needs to send instructions to inform other vehicles that there is no vehicle in region I and other vehicles take actions based on the positioning information. If the distance between the region and the certain vehicle is the shortest, then it starts to move, enters region I and sends instructions. This cycle is repeat. When the vehicle in motion does not receive any instructions, indicating that the road is clear, it can go through the road crossing point directly.

This intersection avoidance algorithm provides an effective method of traffic junction's management. It not only can let vehicles pass through the junction safely, but also change the traditional thought of controlling vehicles to run or wait by traffic lights.

3. The Communication Protocols between Two Vehicles

There must be a communication between two vehicles to complete the tracing and tracking. This section introduces the methods of the communication protocols, the provisions of transmission protocols, and feedback protocols. Finally, dynamic simulation of sending and receiving messages is carried out to verify the function of the protocols.

3.1. Pulse Width Modulator (PWM). At present, many communication tools can be put into use, but none of them meet the requirements of the intelligent vehicle as lack of communication protocols [11].

PWM is an abbreviation of the "Pulse Width Modulation." Its principle is using the digital output of the microprocessor to control the analog circuits by changing the pulse width to control the output voltage and changing the cycle of the pulse modulation to control the output frequency. It is frequently used in the field of measurement, communication, control, and power transform. PWM is different with some other technologies. It uses the high-resolution counters to modulate the duty cycle of the square wave and encodes the analog signals with numeric characters. Because at any moment, the DC power supply of the full amplitude only can be placed in one of the two cases on (ON) and off (OFF), the property of the PWM signal is digital, remaining unchanged. Current source or voltage is applied to the analog load in an on (ON) or off (OFF) repetitive pulse sequence. When it is on (ON), DC power supply is operating. When it is off (OFF), DC power supply is cut off.

PWM can be used to control the coding of the command word. Sending pulses of varying width represents different meanings. The receiver can use a timer to measure the pulse width. The timer is activated by the beginning of the pulse and deactivated when the pulse ends; thereby the elapsed time and the certain command can be determined. Therefore, we can use this function of PWM to specify the communication protocols to help the vehicle system establish effective communication and complete the tracing and tracking of the model car.

PWM generates a repeated alternately output signal between high level and low level, that is, the PWM wave, which specify the required clock period and duty cycle to control the duration of the high and low. Duty cycle refers the percentage of the time that the signal is at high level to the whole signal cycle and the time is determined by the pulse width. Therefore, we can set a fixed signal cycle to the PWM wave and specify the vehicle interactive communication protocol using different pulse width setting by different duty cycle.

Figure 9 describes three different PWM signals. (a) shows a 10% duty cycle PWM output, that is; 10% of the entire signal cycle is on and the remaining 90% of the time is off. (b) and (c) show, respectively, the duty cycle of the PWM output of 50% and 90%. These three PWM output encodes represent three different analog signal values at the strengths of 10%, 50%, and 90% to the full scale. For example, assuming that the power supply is 9 V and the duty cycle is 10%, the corresponding analog signal is 0.9 V.

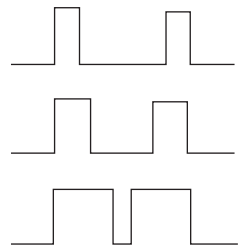


Figure 9: The three different PWM signals.

(1) Signals from the processor to the system are in digital form which means DA/AD converter is no longer needed and the impact of noise can be reduced to a minimum. Signal stored in digital form is particularly strong to resist the noise. Only when the noise is big enough to change logic 1 to logic 0 or change logic 0 to logic 1, the digital signal can be interfered.

(2) It can greatly extend the communication distance. Based on this advantage, the PWM sometimes can be used for communication between two vehicles.

3.2. The Definition of the Transmission Protocol. The communication between two vehicles must be controlled by messages. Messages shall not be disorganized but shall obey some kinds of rules. Since these rules have not been defined yet in the field of vehicle research, we need to specify them.

Considering that there are many protocols used for the communication between vehicles, we define the length of the message transmission protocol as 4 bytes, totally 32 bits, to ensure that each protocol can be defined and newly found protocols can be added as well.

Errors may occur during the process of message transmission, so receivers may get wrong messages as a result. In order to make the receiver distinguish which messages are wrong, the bit at the end of the last byte is used for error checking. The check method is "XOR"; that is, xor the first 31 bits. If the result is the same as the last one, it means no error occurs in the message transmission process and the received message is correct; otherwise error occurs during the transmission, the received message is wrong, and the sender is required to retransmit the message. The last bit of the message from the sender is the result of xor'ing its first 31 bits.

3.2.1. The Types of the Message Transmission. Many actions may occur when the vehicle is traveling on the road such as the ascent or descent, left turning or right turning. These actions are the message that we need to formulate. To make these message more specific, we classify them into the following types.

(1) Velocity Transmission. When a vehicle is tracking another one, the distance between them shall be monitored from time to time in order to prevent end-to-end collisions. To compute the safe spacing, the velocities of the two vehicles are needed.

Table 1: The turning message.

00000000	00000000	00010000	00000001	Turn left and the angle is less than or equal to 1°
00000000	00000000	00010000	00000010	Turn left and the angle is between 1° and 2°
00000000	00000000	00010000	00000100	Turn left and the angle is between 2° and 3°
⋮	⋮	⋮	⋮	⋮
00000000	00000000	00010000	00001000	Turn right and the angle is less than or equal to 1°
00000000	00000000	00010000	00001011	Turn right and the angle is between 1° and 2°
00000000	00000000	00010000	00001101	Turn right and the angle is between 2° and 3°
⋮	⋮	⋮	⋮	⋮

Some details such as checkouts and security were leaved out here.

One of the vehicles can get its own velocity by the sensor and get the other one's velocity by transmission. Therefore the transmission rate protocol is obviously necessary.

(2) Left or Right Turning. Vehicles may face the situations of turning a corner or turning round when they are traveling on the road. They need to turn left or right to continue moving forward on these occasions. In order to track the vehicle ahead successfully, the following one needs to be informed of the detailed messages including the degree of turning. So it is important to specify the left or right turning protocol.

(3) Acceleration and Deceleration. Vehicles will certainly be in the process of acceleration or deceleration, so the acceleration and deceleration protocol is needed to be established. When a driver starts a vehicle, he generally shifts into first gear and steps the throttle to accelerate. When the vehicle reaches a speed of 15 kilometers per hour, it is upshifted into second gear. If the driver needs acceleration, he simply inputs some gas by pressing the accelerator. When the vehicle reaches a speed of 30 kilometers per hour, it is upshifted into third gear. If the driver still needs acceleration, he speeds the vehicle up by the accelerator. When the vehicle reaches a speed of 40 kilometers per hour, it is upshifted into fourth gear. After that if acceleration is still needed, the drive keeps pressing the accelerator until the vehicle reaches the desired speed. Deceleration is implemented by downshifting and braking to be introduced next.

(4) Braking. Vehicles have to brake in case of emergency. Depending on the degree of emergency cases, the levels of brake are also different. In this paper, brake can be divided into four levels. Level 1 is sudden brake; the brake pedal is pushed to the floor with the fastest speed. Level 2 is intermediate brake; the brake pedal is pushed to two-thirds. Level 3 is light brake; the brake pedal can be pushed to one third. Level 4 is the lightest brake; use the first half of the brake pedal that is designed to be soft and push it slightly with the clutch.

(5) Overtaking. There are more than two vehicles in road traffic and there may be other vehicles inserted between them; of course they will also overtake other vehicles. When the distance between the two vehicles is large enough and the front vehicle is slower than the rear vehicle for a long time, the driver in the rear one can consider overtaking. The vehicle being tracked shall send messages to the following vehicle when it successfully finishes overtaking so as to indicate the following vehicle to overtake successfully as well.

(6) Front Road Status. Owing to the tracer which is following the front vehicle, the tracer will be blind to the road traffic if the front vehicle does not inform it. The road traffic ahead can be described as "Road Construction Ahead," "Traffic Jam Ahead," and "Rest Area Ahead."

(7) Emergency. Vehicle will encounter all sorts of unexpected situations, such as the front vehicle getting a flat tire, traffic accidents ahead, keeping intervehicle spacing and the front vehicle getting flameout, and so forth.

(8) Ascent or Descent. When the driving road is not flat, it always comes with ascents or descents. The front vehicle transmits messages to the tracer during ascents or descents and tells the tracer about the probable angle of the slope so that the tracer can track the front vehicle without any trouble.

(9) Road Condition. The vehicle being tracked needs to transmit messages to its tracer whether it is at the corner or the straight. This protocol is used to transmit messages like this.

(10) Other. In addition to the previous protocols, other protocols also exist; for example, the vehicle being tracked has arrived in destination, the target being tracked and the tracer switch roles, and so forth.

3.2.2. The Codes and Regulations of the Message Transmission

(1) Velocity Transmission. The current 2 bytes and the first 4 bits of the third byte are all 0 which represent sending the velocity of the vehicle; the unit of the velocity is km/h; the last 4 bits of the third byte and the first 7 bits of the fourth byte describe the velocity in binary representation, for example: 00000000000000000000000100000111 on behalf of the vehicle at a speed of 67 kilometers per hour.

(2) Left or Right Turning. The codes of left or right turning message are shown in Table 1.

Table 2: The codes of feedback protocol and its significations.

00000000	00000000	00000000	00000000	Transmit successfully
00000000	00000000	00000000	00000001	Transmit failed, please resend
00000000	00000000	00000000	00000010	The protocol does not belong to the certain type, please resend
00000000	00000000	00000000	00000011	Please switch the role
00000000	00000000	00000000	00000100	A tracer leaves
00000000	00000000	00000000	00000101	Request the velocity of the tracked vehicle
00000000	00000000	00000000	00000110	Request the road condition
⋮	⋮	⋮	⋮	⋮

3.2.3. The Codes and Regulations of the Error Message Transmission

(1) Errors Occur in the Process of Transmission. The vehicle being tracked transmits the correct message, but an error occurs in the process of transmission; that is, one bit or several bits of the message change from 0 to 1 or from 1 to 0, so that the tracer figures out that the message is incorrect after receiving and checking it. Then, the tracer shall respond to the message as "Transmission error, please resend!"

The vehicle being tracked transmits an error message itself, that is, the result of xor'ing the first 31 bits is not equal to the last bit. Then, the tracer shall respond with the message as "Transmission error, please resend!"

(2) The Message Does Not Belong to a Certain Type. As there are various kinds of messages, we classify these message into different types to make them easy to remember, such as left or right turning message and ascent or descent protocol. We design a page for each type, when transmitting the message that is uncorrelated with the certain page; the page cannot receive it. The tracer shall respond to the message as "The message does not belong to this type, please resend!"

(3) The Message Has Not Been Defined. As the length of the message is defined as 4 bytes, totally 32 bits, the code space still has much room to use. When the vehicle being tracked sends the code that is not corresponding with any actions, the tracer shall respond to the message as "The message has not been defined, please resend!"

3.2.4. The Definition of Feedback. The tracer replies to the vehicle being tracked after receiving the message to tell if the message is received correctly or if it is necessary to resend the message. The message sent back by the tracer is called feedback. The length of feedback is 4 bytes, 32 bits. The feedback protocol is more simpler than the transmission protocol; it is mainly divided into the following types.

(1) Transmit Successfully. If the tracer receives messages from the vehicle being tracked correctly, the feedback is "Transmit successfully!"

(2) Transmit Failed, Please Resend. The vehicle being tracked transmits an error message itself; that is, the result of xor'ing the first 31 bits is not equal to the last bit. The feedback is "Transmission failed, please resend!"

(3) The Message Does Not Belong to a Certain Type, Please Resend. When the tracer receives the message that is not corresponding to the certain type, the feedback is "The message does not belong to the certain type, please resend!"

(4) Please Switch the Role. When the vehicle being tracked transmits the message of switching the role, the tracer replies with the feedback "Please switch the role of tracer!" and informs the vehicle to slow down.

(5) A Tracer Leaves. When a tracer is about to leave the queue, the tracer shall send the message "A tracer leaves!" to let the vehicle being tracked know the situation.

(6) Request the Velocity of the Tracked Vehicle. When it comes to the case of computing the safe spacing between two vehicles at a certain time, the velocity of the tracked vehicle is needed and the tracer can send the message "Request the velocity of the tracked vehicle!" to get the information.

(7) Request the Road Condition. The vehicle being tracked needs to transmit messages to its tracer whether it is at the corner or the straight so as to compute the safe spacing between them. The tracer needs to send the message; that is, "Request the road condition!"

3.2.5. The Codes and Regulations of the Feedback. The codes of feedback message are shown in Table 2.

The security and validation of the protocol can refer to the relevant sections of the existing mobile network protocols and are no longer discussed in this paper [12].

4. Conclusion

This paper mainly talks about the vehicle tracing and tracking mechanism automatically. By learning the vital supporting techniques of intelligent vehicles, we discuss how to smooth the situations that the intelligent vehicle meets and design algorithms to solve part of the problems on the combination of these techniques and the traffic conditions. This paper concentrates not only on a single vehicle, but also on the mutual interaction of multiple vehicles concerning vehicle tracking

and intersection obstacle avoidance. As it refers to intersection obstacle avoidance, the tradition of controlling vehicles to run or wait by traffic lights at crossroads has been altered by a new thought that vehicles contact each other and exchange the messages of the "occupied" or "released" state of the road resource at junctions. Furthermore, the research on vehicle communication protocols is a huge breakthrough. Although applying wireless communication technologies to intelligent vehicles has been mentioned by domestic researchers many times, none of them make a detailed definition of the vehicle communication protocols.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant no. 61070169 Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, Specialized Research Foundation for the Doctoral Program of Higher Education of China no. 20103201110018, and Application Foundation Research of Suzhou of China no. SYG201118.

References

[1] S. Panichpapiboon and W. Pattara-atikom, "A Review of information dissemination protocols for vehicular Ad Hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 3, pp. 784–798, 2011.

[2] B. Yu, C.-Z. Xu, and M. Guo, "Adaptive forwarding delay control for VANET data aggregation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 11–18, 2012.

[3] J. J. Pan, S. J. Pan, J. Yin, L. M. Ni, and Q. Yang, "Tracking mobile users in wireless networks via semi-supervised colocalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 587–600, 2012.

[4] K. Donghyun, B. H. Abay, R. N. Uma, W. Wu, W. Wang, and A. O. Tokuta, "Minimizing data collection latency in wireless sensor network with multiple mobile elements," in *Proceedings of the IEEE Infocomm*, pp. 504–512, March 2012.

[5] A. T. Erman, A. Dilo, and P. Havinga, "A virtual infrastructure based on honeycomb tessellation for data dissemination in multi-sink mobile wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 17, pp. 1–72, 2012.

[6] R. Tan, G. Xing, B. Liu, J. Wang, and X. Jia, "Exploiting data fusion to improve the coverage of wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 2, pp. 450–462, 2012.

[7] D. Raychaudhuri and N. B. Mandayam, "Frontiers of wireless and mobile communications," *Proceedings of the IEEE*, vol. 100, no. 4, pp. 824–840, 2012.

[8] C. Konstantopoulos, G. Pantziou, D. Gavalas, A. Mpitziopoulos, and B. Mamalis, "A rendezvous-based approach enabling energy-efficient sensory data collection with mobile Sinks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 809–817, 2012.

[9] C. Jorge, M. Sonia, K. Timur, and B. Francesco, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[10] A. Kwok and S. Martínez, "Deployment algorithms for a power-constrained mobile sensor network," in *IEEE International Conference on Robotics and Automation (ICRA '08)*, pp. 140–145, Pasadena, Calif, USA, 2008.

[11] J. Jia, J. Chen, X. Wang, and L. Zhao, "Energy-balanced density control to avoid energy hole for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 812013, 10 pages, 2012.

[12] H. Shen, G. W. Bai, L. Zhao, and Z. M. Tang, "An adaptive opportunistic network coding mechanism in wireless multimedia sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 565604, 13 pages, 2012.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 403165, 15 pages
http://dx.doi.org/10.1155/2013/403165

Research Article

EasiND: Neighbor Discovery in Duty-Cycled Asynchronous Multichannel Mobile WSNs

Tingpei Huang,[1,2] Haiming Chen,[1] Li Cui,[1] and Yuqing Zhang[2]

[1] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*
[2] *University of Chinese Academy of Sciences, Beijing 100049, China*

Correspondence should be addressed to Li Cui; lcui@ict.ac.cn

Received 7 June 2013; Accepted 1 July 2013

Academic Editor: Hongli Xu

Neighbor discovery is one of the first steps to establish communication links between sensor nodes; thus it becomes a fundamental building block for wireless sensor networks (WSNs). Traditional neighbor discovery protocols mainly focus on static wireless networks or networks where all nodes operate on the same frequency. However, the proliferation of mobile devices and multichannel communications post new challenges to neighbor discovery problem. In this paper, we present a neighbor discovery protocol named EasiND for asynchronous duty-cycled multichannel mobile WSNs. First, we propose a neighbor discovery system based on quorum system, which can bound the discovery latency in multichannel scenarios with low power consumptions. Second, we design an optimal asynchronous neighbor discovery system for multichannel mobile WSNs based on cyclic difference set. It is optimal in the sense that it minimizes the power consumption with bounded discovery latency under desired duty cycles. Finally, we validate the performance of EasiND through both theoretical analysis and test-bed evaluations. EasiND provides a 33.3% reduction in power-latency product in theory compared to U-Connect. Meanwhile, test-bed evaluation results show that EasiND decreases average discovery latency by up to 86% compared to U-Connect and achieves at least 93.5% average fraction of discoveries in a predefined time limitation under various network conditions.

1. Introduction

Neighbor discovery, namely, the ability for each node to find the neighboring nodes in the physical proximity, is one of the first steps and a fundamental building block in configuring and managing wireless networks, since a node first has to find at least one potential target node within its communication range before initializing any data communications. For example, the information obtained from neighbor discovery, namely, the set of nodes that a node can directly communicate with, is needed to support basic functionalities such as medium access control and routing. Furthermore, this information is also needed by topology control and clustering protocols to enhance the performance and efficiency of the networks. Due to its critical importance, neighbor discovery has received a lot of attention, and a number of approaches have been proposed in the literature [1–9]. Most existing neighbor discovery schemes are designed for static wireless networks [2, 4, 10–13] or networks where all nodes operate on the same frequency (single-channel communication networks) [1, 3, 5]. Therefore, those neighbor discovery methods cannot be directly applied to multichannel mobile WSNs.

The rapid proliferation of millions of mobile sensor nodes and smart phones has resulted in a wide variety of mobile applications, such as mobile social networks and mobile sensing applications [14–17]. Meanwhile, the multichannel communications which allow radios tune their operating frequency over different channels provide new opportunities for improving the performance of wireless networks; thus they have been widely adopted by current WSN systems [18–20]. The mobility of sensor nodes and multichannel communication make neighbor discovery in such multichannel and mobile WSNs more challenging, especially to implement neighbor discovery schemes with high energy efficiency and low discovery latency without requiring clock synchronization between sensor nodes. So far, only several related methods have been proposed for multichannel WSNs [21–23].

These neighbor discovery schemes designed for personal area networks focus on reducing discovery latency but put less effort on energy conservation. However, discovery latency, energy consumption, and fraction of discoveries are all critical performance metrics for neighbor discovery scheme in asynchronous duty-cycled multichannel mobile WSNs. On the one hand, most mobile devices or sensor nodes are battery powered and require low energy consumption to prolong the network life as long as possible. On the other hand, in time-sensitive and data-intensive mobile applications, in order to enable full potential contact opportunities, it is very important for sensor nodes to discover as many as possible neighbors in a short time period.

To cope with the previous problems, we design a novel neighbor discovery scheme, named EasiND, for asynchronous duty-cycled multichannel mobile WSNs. The key goal for EasiND is to reduce power consumption, while still enabling effective neighbor discovery, where effectiveness can be determined by both the ability to discover more neighbors in a reasonable time constraint and the latency to discovery these neighbors.

The main contributions of this paper are summarized as follows.

(i) We first formulate the neighbor discovery problem in multichannel WSNs by connecting it with quorum system. Also, we introduce an algorithm using the intersection property of quorum system to construct a neighbor discovery system, which enables successful discoveries over multiple channels between any two neighbor discovery sequence.

(ii) Without requiring global clock synchronization, we propose an algorithm to construct asynchronous neighbor discovery system for duty-cycled multichannel mobile WSNs by utilizing the rotation closure property of cyclic quorum systems. Our algorithm can bound the worse-case discovery latency and achieve minimum discovery latency in theory at desired duty cycle. Meanwhile, our scheme yields better performance in trade-off between power consumption and discovery latency.

(iii) We evaluate our method through both theoretical analysis and test-bed experiments. Evaluation results show that our method consistently outperforms U-Connect in terms of power-latency product and fraction of discoveries under various network conditions.

The rest of the paper is organized as follows. We discuss related work on neighbor discovery in Section 2. In Section 3, we review some definitions and concepts about cyclic quorum system. Section 4 describes design details of EasiND. In Section 5, we present a detailed evaluation of EasiND. Finally, Section 6 concludes the paper.

2. Related Work

Neighbor discovery protocols for duty-cycled asynchronous wireless networks mostly operate on a time-slot basis, where they divide the time into slots and all nodes use the same size of the time slot. Based on the protocol used, nodes wake up during some specific active time slots to run the neighbor discovery process, and keep asleep during the remaining time slots. Once they are active in overlapping time slots, a successful discovery would happen between two neighbors. To be energy efficient, a neighbor discovery scheme must use as few active time slots as possible to discover as many neighbors as possible within a reasonable time limitation.

Current neighbor discovery methods fall broadly into the following two categories.

2.1. Single-Channel Based Methods. The neighbor discovery protocols where all nodes operate on the same operation frequency belong to this category. These neighbor discovery protocols can be further divided into two classes, probabilistic and deterministic.

The schemes proposed in [2, 4, 10, 24] are probabilistic, where each node chooses operating in transmit, listening or sleep states with certain predefined probability. In birthday protocol [10], nodes listen, transmit, or sleep in a probabilistic round-robin fashion, which provides a concrete way to trade off between discovery energy efficiency, success in discovering neighbors and discovery latency. The methods in [2, 24] are based on additional feedback mechanism. If a node cannot receive a beacon due to collisions, it transmits a feedback message. If a node does not receive any feedback message after transmitting a beacon, it goes into a passive state on the assumption that the beacon was successfully received. The paper [4] studies neighbor discovery in multipacket reception networks where packets from multiple transmitters can be received successfully at a receiver.

Deterministic neighbor discovery protocols let nodes wake up at specific time slots according to a deterministically designed schedule. The approaches presented in [1, 3, 11–13] belong to this category. The methods presented in [11, 13] are quorum based. In [11], the time is divided into a sequence of slots which are grouped into a $m \times m$ row major grid matrix within contiguous slots, where m is a global parameter which depends on the required duty cycle. Each node randomly picks a row and a column, in which the node keeps awake and the discovery between these two nodes will occur at two intersections. The paper [12] formulates wakeup schedule as a block design problem in combinatorics and gives an optimal solution which achieves minimum idle state energy consumption with bounded neighbor discovery latency. The Disco proposed in [1] is based on Chinese Remainder Theorem, where each node selects two prime numbers independently and begins to transmit and receive whenever the local counter of node is divisible by either of two primes. The sum of two primes' reciprocal is equal to node's duty cycle. To improve Disco's performance, U-Connect proposed in [3] presents an activation pattern using one prime p. Instead of just waking up only one time slot every p time slots, U-Connect also wakes up $(p + 1)/2$ time slots every p^2 time slots. U-Connect achieves lower discovery latency when compared to Disco. The paper [9] presents a group-based discovery protocol as a performance add-on to existing pairwise mobile discovery designs. It designs a schedule reference mechanism among nodes to accelerate

the discovery process. The paper [7] proposes an on-demand generic discovery accelerating middleware for many existing neighbor discovery protocols.

2.2. Multichannel Based Methods. In the literature, a lot of research work has been done for single-channel wireless networks. However, there are little work on neighbor discovery problem for multichannel wireless networks. Only several recent approaches are proposed for personal area networks or wireless mesh networks [21–23]. The SWEEP strategies in [22] are designed for IEEE 802.15.4-based networks operating in the beacon-enabled mode. SWEEP determines the listening schedule of discovering node to detect a foreign personal area network. Such a schedule decides when to listen on which channel and for how long. The paper [23] improves the performance of SWEEP by aggressively changing the channel after short, specifically selected time periods of observation with reusage of observations made on a given channel. The approach proposed in [21] is designed for multichannel wireless mesh networks. The focus of all the previous schemes is to minimize the discovery latency and they are inefficient in energy consumption.

In contrast to existing work, we study the neighbor discovery problem in asynchronous duty-cycled multichannel mobile WSNs. Our paper is motivated by the QCH introduced in [25], which establishes control channel for multichannel and dynamic spectrum access networks. Because QCH does not consider energy consumption problem, it cannot be directly applied to duty-cycled multichannel mobile WSNs. We extend QCH and propose a novel neighbor discovery protocol named EasiND for duty-cycled multichannel mobile WSNs. The main goal of EasiND is to reduce energy consumption, while still enabling effective neighbor discovery for multichannel communications, where effectiveness can be determined by both the ability to discover more neighbors in a reasonable time constraint and the latency to discovery these neighbors.

3. Preliminaries for Quorum System

In this section, we introduce some background knowledge related to the cyclic quorum system that will be used throughout this paper.

Definition 1. Given a cycle length n, let $U = \{0, 1, \ldots, n-1\}$ be a finite universal set of n elements. A quorum system G under U is a collection of nonempty subsets of U, which satisfies the following intersection property:

$$\forall g, h \in G: \quad g \cap h \neq \emptyset. \tag{1}$$

Each $g \in G$ is a subset of U and is called a quorum.

Definition 2. A set $D = \{a_1, a_2, \ldots, a_k\}$ modulo n, $a_i \in \{0, 1, \ldots, n-1\}$, is called a cyclic (n, k, λ)-difference set if for every $d \not\equiv 0 \pmod{n}$ there are exactly λ ordered pairs (a_i, a_j), $a_i, a_j \in D$ such that $a_i - a_j \equiv d \pmod{n}$.

Definition 3. A set $D = \{a_1, a_2, \ldots, a_k\}$ modulo n, $a_i \in \{0, 1, \ldots, n-1\}$, is called a relaxed cyclic (n, k)-difference set if for every $d \not\equiv 0 \pmod{n}$ there exists at least one ordered pair (a_i, a_j), $a_i, a_j \in D$ such that $a_i - a_j \equiv d \pmod{n}$.

Theorem 4. *A group of sets $g_i = \{a_1+i, a_2+i, \ldots, a_k+i\}$ modulo n, $i \in \{0, 1, \ldots, n-1\}$, is a group of cyclic quorum sets if and only if $D = \{a_1, a_2, \ldots, a_k\}$ is a relaxed cyclic (n, k)-difference set.*

Theorem 4 has been proved in [26], which demonstrates that we can construct cyclic quorum system using cyclic difference sets. For example, $D = \{0, 1, 2\}$ modulo 4 is a relaxed cyclic (4,3)-difference set. The sets $G = \{\{0, 1, 2\}, \{1, 2, 3\}, \{2, 3, 0\}, \{3, 0, 1\}\}$ are a cyclic quorum system constructed by relaxed cyclic (4,3)-difference set $D = \{0, 1, 2\} \pmod 4$. The paper [13] has proved that any quorum g in a cyclic quorum system under $U = \{0, 1, \ldots, n-1\}$ must have a cardinality $|g| \geq \sqrt{n}$. That is to say, given n, the minimum size k of the quorums in cyclic quorum system is the $\sqrt{n}$. In fact, we can achieve this theoretical lower bound by constructing the Singer difference sets [27]. Next, we give the Singer difference sets theorem which is introduced and proved in [28].

Theorem 5. *Let q be a prime power; then there exists a $(n, k, 1)$-difference set under Z_n. Such a difference set is called a Singer difference set. Here, $n = q^2 + q + 1$, $k = q + 1$, and Z_n denotes the set of nonnegative integers less than n.*

Because $n = q^2 + q + 1$ and $k = q + 1$, so k approximates the lower bound $\sqrt{n}$.

Definition 6. Given an integer $i \geq 0$ and a quorum g in a quorum system G under U, we define $g + i = \{(x + i) \bmod n : x \in g\}$.

Definition 7. A quorum system G under U is said to have the rotation closure property if $\forall g, h \in G$, $i \in \{0, \ldots, n-1\}$, $(g + i) \cap h \neq \emptyset$.

For instance, the quorum system $G = \{\{0, 1, 2\}, \{1, 2, 3\}, \{2, 3, 0\}, \{3, 0, 1\}\}$ under $\{0, 1, 2, 4\}$ has the rotation closure property.

The following theorem has been proved in [13].

Theorem 8. *The cyclic quorum system satisfies the rotation closure property.*

4. Design of EasiND

In this section, we present the design of EasiND neighbor discovery protocol in detail. EasiND allows some nodes with no prior clock synchronization information to discover other networks or devices which would dynamically adapt their operating frequency over different channels. The discovery is completed in bounded time when nodes are within the transmission range of each other in ideal communication environment. For the ease of presentation, we first provide the problem formulation and introduce the synchronous neighbor discovery problem in duty-cycled multichannel

mobile WSNs and then describe the asynchronous neighbor discovery system.

4.1. Problem Statement. We consider the following network scenarios. There are one or more mobile nodes which move around some existed networks or devices with unpredictable mobility patterns to collect data from or want to join in the existed networks. The existed networks adapt their operating frequency over different channels according to current communication conditions, and they periodically transmit beacons to let mobile nodes to discover them. Therefore, our neighbor discovery protocol EasiND consists of two key components, beacon scheduling sequence (BSS) and channel scanning sequence (CSS). Suppose there are N channels in our duty-cycled multichannel networks, labeled as $0, 1, \ldots, N-1$. For example, in IEEE 802.15.4-based 2.4 GHz WSN, there are 16 channels numbered from 11 to 26 that can be used in our implementation of EasiND. We assume that time is divided into neighbor discovery periods (NDPs), where each NDP is composed of T time slots. For the sake of expression, we assume that each time slot is of unit duration so that each NDP is also T. We assume that all nodes use the same duty cycle; namely, we consider symmetric duty-cycled systems and leave the neighbor discovery in asymmetric duty-cycled multichannel WSNs to future study. Moreover, we assume that all nodes in network are synchronized with each other for the time being. We will present the design of asynchronous neighbor discovery system which does not require global clock synchronization in Section 4.3.

Next, for the sake of clarification, we present the design of BSS and CSS, respectively. The uniform formulation of BSS and CSS is reasonable, as we will see.

A BSS determines the intervals of time at which each network or device transmits beacons on its current operating channel. We represent a BSS s of period T as a set of triples:

$$\begin{aligned} s = \{&(0, x(0), c), (1, x(1), c), \ldots, (i, x(i), c), \ldots, \\ &(T-1, x(T-1), c)\}, \end{aligned} \tag{2}$$

where $c \in \{0, 1, \ldots, N-1\}$ and $x(i)$ is a binary value function which is defined by the following formulation:

$$x(i) = \begin{cases} 0, & \text{if node goes to sleep in the } i\text{th time slot,} \\ 1, & \text{if node becomes awake in the } i\text{th time slot.} \end{cases} \tag{3}$$

The triple $(i, x(i), c)$ indicates that network transmits beacons in the ith time slot of a BSS period T on channel c if $x(i)$ is equal to 1.

A CSS determines the schedule with which each mobile node scans all available channels. We denote a CSS r of period T as a set of triples:

$$\begin{aligned} r = \{&(0, x(0), c_0), (1, x(1), c_1), \ldots, (i, x(i), c_i), \ldots, \\ &(T-1, x(T-1), c_{T-1})\}, \end{aligned} \tag{4}$$

where $c_i \in \{0, 1, \ldots, N-1\}$ represents the channel index of sequence r in the ith time slot of a CSS period and i is the slot index. The triple $(i, x(i), c_i)$ indicates that mobile node becomes awake and goes to operate on channel c_i at the ith time slot when $x(i)$ equals 1.

Given two sequences s and r, if $(x, y, z) \in s \cap r$ and $y = 1$, the triple (x, y, z) is called an overlap between s and r. In this case, the xth time slot is called rendezvous time slot and channel z is called rendezvous channel between s and r. If network selects s and mobile node selects r, respectively, as their neighbor discovery scheduling sequence, then successful discovery occurs in the rendezvous time slots on rendezvous channels.

Let $I(s, r)$ denotes a binary value function that indicates whether the triple (x, y, z) is an overlap between s and r, for example,

$$I_{x,z}(s, r) = \begin{cases} 1, & \text{if } \exists y = 1, \text{ s.t. } (x, y, z) \in s \cap r, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Let $O(s, r)$ denotes the number of overlaps between s and r. Then, we have

$$O(s, r) = \sum_{x=0}^{T-1} \sum_{z=0}^{N-1} I_{x,z}(s, r). \tag{6}$$

Let $P_x(s)$ denotes a binary value function which indicates whether node keeps awake in the time slot x in the sequence of s of period T. Specifically, $P_x(s)$ is defined as follows:

$$P_x(s) = \begin{cases} 1, & \text{if } y = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Let $D_{\mathrm{BSS}}(s)$ and $D_{\mathrm{CSS}}(r)$ denote the number of time slots on which nodes keep awake in the sequences s and r of period of T, respectively. Then, we have

$$\begin{aligned} D_{\mathrm{BSS}}(s) &= \sum_{x=0}^{T-1} P_x(s), \\ D_{\mathrm{CSS}}(r) &= \sum_{x=0}^{T-1} P_x(r). \end{aligned} \tag{8}$$

Based on our assumption of symmetric duty-cycled system, we have $D_{\mathrm{BSS}}(s) = D_{\mathrm{CSS}}(r)$.

Thereby, we have the following neighbor discovering system design problem.

Problem 9. Give T and C, where C denotes the duty-cycle used by nodes in our system, the neighbor discovery system in duty-cycled multichannel mobile WSNs is to devise a set of BSS of period T denoted as S, and a set of CSS of period T denoted as R, which satisfy the following three properties:

(1) $\forall s \in S$ and $\forall r \in R$, $|s| = T$ and $|r| = T$;

(2) $k \geq 1$, where $k = \min_{\forall s \in S, \forall r \in R}\{O(s, r)\}$;

(3) $D_{\mathrm{BSS}}(s) = D_{\mathrm{CSS}}(r) \leq C \times T$.

From (2) and (4), we know that the only difference between the constructions of s and r is that the BSS s uses

Input:
m, N, a channel set $C = \{0, 1, \ldots, N-1\}$, $U = \{0, 1, \ldots, m-1\}$, a quorum system G under U and a binary variable b, $b = 0$ demonstrates that we need to construct BSS s, otherwise, to construct CSS r;
Output:
Q;
(1) $Q = \emptyset$;
(2) **for** $j = 0$ to $(|G| - 1)$ **do**
(3) **for** $f = 0$ to $(N - 1)$ **do**
(4) **for** $i = 0$ to $(m - 1)$ **do**
(5) **if** $i \in g_j$ **then**
(6) **if** $b = 0$ **then**
(7) $x(i + f \cdot N) = 1$;
(8) $c = c_{\text{bss}}$;
(9) **else**
(10) $x(i + f \cdot N) = 1$;
(11) $c = c_{\text{css}} = f$;
(12) **end if**
(13) **end if**
(14) **if** $z \notin g_j$ **then**
(15) $x(i + f \cdot N) = 0$;
(16) $c = c_r$, randomly selected from the set $\{0, 1, \ldots, N-1\}$;
(17) **end if**
(18) **end for**
(19) **end for**
(20) $S = S \cup s$, $R = R \cup r$, $Q = S \cup R$;
(21) **end for**

Algorithm 1: Quorum-based neighbor discovery system construction algorithm.

the same channel during the whole period of T; however, the CSS r should visit all the channels available in the system. Therefore, we can design a uniform set of neighbor discovery sequences of period of T for BSS and CSS using (4), denoted as Q. So, we have $S \subset Q$ and $R \subset Q$. The set Q is called a neighbor discovery system of period of T. The k and $D_{\text{BSS}}(s)/T = D_{\text{CSS}}(r)/T$ are the overlapping degree and duty cycle of the neighbor discovery system Q, respectively.

It is already clear that a neighbor discovery system Q is a quorum system under the universal set $U = \{(x, y, z) \mid x \in [0, T-1], y \in \{0, 1\}, z \in \{0, 1, \ldots, N-1\}\}$, because it satisfies the intersection property: any two sequences in Q have at least one overlap. Each neighbor discovery sequence in Q is a quorum.

4.2. Construction of Quorum-Based Neighbor Discovery System in Duty-Cycled Multichannel Mobile WSNs. In this section, we present an algorithm which uses a quorum system to construct a neighbor discovery system Q for duty-cycled multichannel mobile WSNs; namely, we need to construct a set of BSS S and CSS R of period T, such that they satisfy three properties defined in Problem 9. We refer to such algorithm as Algorithm 1.

4.2.1. Construction of BSS. First, we introduce the algorithm to construct BSS set S, which is easily to be modified to construct CSS set R as we will see. Without loss of generality, we assume that each BSS s is composed of N segments, where each segment is composed of m time slots. Therefore, the period of each BSS is $T = m \times N$. Specifically, suppose $m = 7$ and $N = 3$; then, we have the following construction process.

(1) First, we construct a universal set $U = \{0, 1, 2, 3, 4, 5, 6\}$ and a quorum system G under U. Let $G = \{g_0, g_1, g_2, g_3\}$; then we can construct the following quorums:

$$\begin{aligned} g_0 &= \{0, 1, 2\}, \\ g_1 &= \{0, 2, 3\}, \\ g_2 &= \{1, 2, 3\}, \\ g_3 &= \{1, 2, 4\}. \end{aligned} \tag{9}$$

(2) Using the quorum $g_0 \in G$, we construct a BSS s_0 using the following procedures.

(i) We make the beacon transmission schedule for the first segment of m time slots using the following two equations:

$$\begin{aligned} x(i) &= \begin{cases} 1, & \text{if } i \in g_0, \\ 0, & \text{if } i \notin g_0, \end{cases} \\ c &= \begin{cases} c_{\text{bss}}, & \text{if } i \in g_0, \\ c_r, & \text{if } i \notin g_0, \end{cases} \end{aligned} \tag{10}$$

Time slots	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s_0	1, 1	1, 1	1, 1	0, c	0, c	0, c	0, c	1, 1	1, 1	1, 1	0, c	0, c	0, c	0, c	1, 1	1, 1	1, 1	0, c	0, c	0, c	0, c
s_1	1, 1	0, c	1, 1	1, 1	0, c	0, c	0, c	1, 1	0, c	1, 1	1, 1	0, c	0, c	0, c	1, 1	0, c	1, 1	1, 1	0, c	0, c	0, c
s_2	0, c	1, 1	1, 1	1, 1	0, c	0, c	0, c	0, c	1, 1	1, 1	1, 1	0, c	0, c	0, c	0, c	1, 1	1, 1	1, 1	0, c	0, c	0, c
s_3	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c

Figure 1: An illustration of BSS set S with $m = 3$, $N = 3$, $c_{\text{bss}} = 1$. The pair in the grey box indicates that the network becomes awake on channel c_{bss}, and the pair in the white box indicates that the network goes into sleep to save energy. The variable c in white box is randomly selected from the set $\{0, 1, \ldots, N-1\}$.

where c_{bss} is the channel which network operates on and c_r is a randomly selected channel from $\{0, 1, \ldots, N-1\}$. $x(i)$ indicates in which time slots the network should keep awake. For the sake of uniformity, we schedule the network to work on channel c_r when the node turns off its radio. In the practical implementation of EasiND, we make no difference between these two cases because it makes no sense to schedule an operating channel for a node when its radio is turned off.

(ii) Repeat the previous procedure to make beacon transmission schedule for each of other two segments. It is should be noted that in the remaining segment, the time slot index i should be the modulo over m for constructing the previous two equations $x(i)$ and c.

(3) Repeat step (2) for each of the other quorums in G, for example, $g_1 = \{0, 2, 3\}$, $g_2 = \{1, 2, 3\}$, and $g_3 = \{1, 2, 4\}$, to construct other three BSSs s_1 to s_3. The four BSSs, namely, s_0, s_1, s_2, s_3, are the elements of the set S, where each sequence has a period of T.

The beacon transmission schedules in BSS set S are illustrated in Figure 1. One quorum in G is used to construct a beacon transmission schedule in S. Thus, we have $|G| = |S|$. Note that $\forall s_0, s_1 \in S$, there are two corresponding quorums $g_0, g_1 \in G$ used to construct s_0 and s_1, respectively. Every beacon transmission schedule in S has the same period of T, where $T = m \times N$. Because of the intersection property of G, g_0 and g_1 overlap at least N times on a specific channel c_{bss} in period of T.

4.2.2. Construction of CSS. We employ the same algorithm to construct CSS set R, except that in step (2) we use the following different equation to design channel scanning scheme. Here, c_{css} is setted to different value in three segments, which indicates the channel that a node should scan. As shown in Figure 2, which illustrates the channel scanning schedules in R, the c_{css} is set to 0 in the first seven time slots (first segment), and 1 in the second seven time slots (second segment), and 2 in the third seven time slots (third segment). Therefore, the node should have scanned each channel at least once in the period of T time slots. That is why we divide the period of T into N segments:

$$c(i) = \begin{cases} c_{\text{css}}, & \text{if } i \in g_0, \\ c_r, & \text{if } i \notin g_0. \end{cases} \tag{11}$$

Note that $\forall s \in S$, $r \in R$; we have $s \cap r \neq \emptyset$ because we use the same procedures and quorum system G to construct S and R. Therefore, the Q which includes S and R is a neighbor discovery system that satisfies the requirements of Problem 9. We refer to the neighbor discovery system constructed using Algorithm 1 as a quorum-based neighbor discovery system.

4.3. Quorum-Based Asynchronous Neighbor Discovery System in Duty-Cycled Multichannel Mobile WSNs. In this section, we present an asynchronous neighbor discovery system which does not require global clock synchronization based on the rotation closure property of cyclic quorum system. The objective is to design an asynchronous BSS set S and CSS set R, so that $\forall s \in S$ and $\forall r \in R$ overlap by at least half of a time slot for every NDP of period T; even the time slot boundaries are misaligned by an arbitrary amount.

We extend the concept of the rotation closure property in Definition 7 to enable its application to our asynchronous neighbor discovery system. We will demonstrate that a neighbor discovery system with the rotation closure property is an asynchronous neighbor discovery system which requires no global clock synchronization.

Definition 10. Given an integer $i \geq 0$, a BSS sequence s and a CSS sequence r in a neighbor discovery system Q of period T, we define

$$\begin{aligned} s + i = \; & \{(j, x(j), c(j)) \mid x(j) = x_s((j+i) \bmod T), \\ & c(j) = c_s((j+i) \bmod T), \\ & j \in [0, T-1]\}, \\ r + i = \; & \{(j, x(j), c(j)) \mid x(j) = x_r((j+i) \bmod T), \\ & c(j) = c_r((j+i) \bmod T), \\ & j \in [0, T-1]\}, \end{aligned} \tag{12}$$

where $x_s((j+i) \bmod T)$ and $c_s((j+i) \bmod T)$ are the $(j+i)$th element in s, and $x_r((j+i) \bmod T)$ and $c_r((j+i) \bmod T)$ are

Time slots	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
r_0	1, 0	1, 0	1, 0	0, c	0, c	0, c	0, c	1, 1	1, 1	1, 1	0, c	0, c	0, c	0, c	1, 2	1, 2	1, 2	0, c	0, c	0, c	0, c
r_1	1, 0	0, c	1, 0	1, 0	0, c	0, c	0, c	1, 1	0, c	1, 1	1, 1	0, c	0, c	0, c	1, 2	0, c	1, 2	1, 2	0, c	0, c	0, c
r_2	0, c	1, 0	1, 0	1, 0	0, c	0, c	0, c	0, c	1, 1	1, 1	1, 1	0, c	0, c	0, c	0, c	1, 2	1, 2	1, 2	0, c	0, c	0, c
r_3	0, c	1, 0	1, 0	0, c	1, 0	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 2	1, 2	0, c	1, 2	0, c	0, c

Figure 2: An illustration of CSS set R with $m = 3$, $N = 3$. The pair in the grey box indicates that the network becomes awake on channel c_{css}, which is 0 in the first seven time slots (first segment), 1 in the second seven time slots (second segment), and 3 in the last seven time slots (third segment). The specific allocation procedure of c_{css} refers to lines 8 and 11 in Algorithm 1. The pair in the white box indicates that the network goes into sleep to save energy. The variable c in white box is randomly selected from the set $\{0, 1, \ldots, N-1\}$.

the $(j+i)$th element in r. For example, given $T = 8$, $m = 4$, $N = 2$, and

$$r = \{(0,1,0),(1,1,0),(2,1,0),(3,0,c),(4,1,1),(5,1,1),(6,1,1),(7,0,c)\}, \tag{13}$$

we have

$$r+2 = \{(0,1,0),(1,0,c),(2,1,1),(3,1,1),(4,1,1),(5,0,c),(6,1,0),(7,1,0)\}. \tag{14}$$

Definition 11. A neighbor discovery system Q of period T is said to have the rotation closure property if $\forall s, r \in Q$, $i \in [0, T-1]$, $s \cap (s+i) \neq \emptyset$, $r \cap (r+i) \neq \emptyset$, $r \cap (s+i) \neq \emptyset$ and $s \cap (r+i) \neq \emptyset$.

Theorem 12. *If a neighbor discovery system Q of period of T satisfies the rotation closure property, any pair of s ($s \in S$) and r ($r \in R$) in Q must overlap at least once for every T consecutive time slots even when the time slot boundaries are misaligned by an arbitrary amount.*

Proof. Suppose that a neighbor discovery system Q satisfies the rotation closure property and two nodes, U and V, respectively; select sequences s and r from S and R. For the sake of easy description, we assume that the length of time slot is 1. We consider the following two cases.

(1) The time slot boundaries are aligned. Without loss of generality, we suppose that node U's clock is i slots ahead of node V's clock. Relative to node V's clock, node U's sequence s is equivalent to $r+i$. Because of the rotation closure property of Q, we have $s \cap (r+i) \neq \emptyset$. Therefore, two sequences s and r must overlap at least one time slot in the period of T. It is obvious that we can obtain the same results with the assumption that node V's clock is i slots ahead of node U's clock.

(2) The time slot boundaries are misaligned. Suppose that node U's clock is ahead of node V's clock by an arbitrary amount of time, such as $i+\delta$, where $i \in [0, m-1]$, $0 < \delta < 1$. If we advance node V's clock by δ, and the sequence r turns into r', then, the boundaries of s and r' are aligned and must overlap at least one time slot in the period of T. Therefore, two sequences s and r must overlap at least δ time slot in every period of T.

□

Theorem 12 states that any neighbor discovery system Q with closure rotation property can guarantee successful discoveries between two nodes, which select BSS sequence from S and CSS sequence from R, respectively, although they are not synchronized with each other. We call such quorum-based neighbor discovery system which satisfies the rotation closure property an quorum-based asynchronous neighbor discovery system. Next, we introduce an algorithm to construct a quorum-based asynchronous neighbor discovery system.

4.3.1. Construction of Quorum-Based Asynchronous Neighbor Discovery System. Here, we extend the Algorithm 1 based on cyclic quorum system introduced in [26] to achieve this objective. Specifically, in step (1) of Algorithm 1, we first construct a relaxed cyclic (m,k)-different set $D = \{a_1, a_2, \ldots, a_k\}$, then we construct a cyclic quorum system G which consists of group of cyclic quorum sets $g_i = \{a_1 + i, a_2 + i, \ldots, a_k + i\} \bmod m$, where $i \in \{0, 1, \ldots, m-1\}$. For example, we construct a relaxed cyclic (7,3)-different set $D = \{1, 2, 4\}$ and a cyclic quorum system G under D. Let $G = \{g_0, g_1, g_2, g_3, g_4, g_5, g_6\}$, then we have

$$\begin{aligned} g_0 &= \{1,2,4\}, \\ g_1 &= \{2,3,5\}, \\ g_2 &= \{3,4,6\}, \\ g_3 &= \{4,5,0\}, \\ g_4 &= \{5,6,1\}, \\ g_5 &= \{6,0,2\}, \\ g_6 &= \{0,1,3\}. \end{aligned} \tag{15}$$

After building the cyclic quorum system G, we construct BSS set S and CSS set R using the same procedures of step (2) and step (3) in Algorithm 1. An asynchronous BSS and CSS based on the previous relaxed cyclic (7,3)-difference set $D = \{1, 2, 4\}$ and cyclic quorum system G are illustrated in

Input:
m, N, k, a channel set $C = \{0, 1, \ldots, N-1\}$, and a binary variable b, $b = 0$ demonstrates that we need to construct BSS s, otherwise, to construct CSS r;
Output:
Q;
(1) $Q = \emptyset$;
(2) construct a relaxed cyclic (m, k)-difference set $D = \{a_1, a_2, \ldots, a_k\}$;
(3) construct a cyclic quorum system $G = \{g_0, g_1, \ldots, g_m\}$ using the relaxed cyclic (m, k)-difference set D;
(4) **for** $j = 0$ to $(|G| - 1)$ **do**
(5) **for** $f = 0$ to $(N - 1)$ **do**
(6) **for** $i = 0$ to $(m - 1)$ **do**
(7) **if** $i \in g_j$ **then**
(8) **if** $b = 0$ **then**
(9) $x(i + f \cdot N) = 1$;
(10) $c = c_{\text{bss}}$;
(11) **else**
(12) $x(i + f \cdot N) = 1$;
(13) $c = c_{\text{css}} = f$;
(14) **end if**
(15) **end if**
(16) **if** $z \notin g_j$ **then**
(17) $x(i + f \cdot N) = 0$;
(18) $c = c_r$, randomly selected from the set $\{0, 1, \ldots, N-1\}$;
(19) **end if**
(20) **end for**
(21) **end for**
(22) $S = S \cup s$, $R = R \cup r$, $Q = S \cup R$;
(23) **end for**

Algorithm 2: Quorum-based asynchronous neighbor discovery system construction algorithm.

Figures 3 and 4. We refer to this construction algorithm for quorum-based asynchronous neighbor discovery system as Algorithm 2.

Because of the rotation closure property of the cyclic quorum system G, it is obvious to conclude that the neighbor discovery system Q of period T constructed by Algorithm 2 satisfies the rotation closure property.

5. Performance Evaluation

In this section, we evaluate the performance of EasiND with earlier method U-Connect proposed in [3].

5.1. Evaluation Metrics. We consider two metrics, namely, power-latency product (PLP) and fraction of discoveries (FD) to evaluate neighbor discovery system.

5.1.1. Power-Latency Product. In duty-cycled asynchronous multichannel mobile WSNs, energy efficiency and discovery latency are two key performance metrics to evaluate the neighbor discovery system. On the one hand, nodes adopt low duty cycle to reduce energy consumption. On the other hand, nodes need to complete quick successful neighbor discovery for further exchanging control information or data communication. Particularly, in mobile sensor networks, nodes require lower neighbor discovery latency to quickly establish network connection for time-sensitive data communication. It is obvious that there is a trade-off between achieving high energy efficiency and reducing the neighbor discovery latency. Therefore, we use the metric power-latency product (PLP) introduced in [3] to trade off between energy efficiency and discovery latency. It is defined as the product of the average power consumption with the worst-case neighbor discovery latency in an ideal communication channel.

5.1.2. Fraction of Discoveries. Another important metric, fraction of discoveries (FD), is often used to evaluate the performance of neighbor discovery protocol. In our paper, it is defined as the number of neighbors discovered in a fixed time limits. It is very important to discover more neighbors in a short time for data-intensive applications in mobile WSNs. For example, in order to increase the opportunities of delivering more data, network needs to discover more mobile sinks when they come into its communication range as quickly as possible.

Time slots	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
s_0	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c
s_1	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c
s_2	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1
s_3	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c
s_4	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1
s_5	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1
s_6	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c

FIGURE 3: An illustration of beacon scheduling system BSS S with $m = 7$, $N = 3$, $c_{bss} = 1$ and a relaxed cyclic difference $D = \{1, 2, 4\}$. The pair in the grey box indicates that the network becomes awake on channel c_{bss}, and the pair in the white box indicates that the network goes into sleep to save energy. The variable c in white box is randomly selected from the set $\{0, 1, \ldots, N - 1\}$.

Time slots	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
r_0	0, c	1, 0	1, 0	0, c	1, 0	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 2	1, 2	0, c	1, 2	0, c	0, c
r_1	0, c	0, c	1, 0	1, 0	0, c	1, 0	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 2	1, 2	0, c	1, 2	0, c
r_2	0, c	0, c	0, c	1, 0	1, 0	0, c	1, 0	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 2	1, 2	0, c	1, 2
r_3	1, 0	0, c	0, c	0, c	1, 0	1, 0	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 2	0, c	0, c	0, c	1, 2	1, 2	0, c
r_4	0, c	1, 0	0, c	0, c	0, c	1, 0	1, 0	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 2	0, c	0, c	0, c	1, 2	1, 2
r_5	1, 0	0, c	1, 0	0, c	0, c	0, c	1, 0	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 1	1, 2	0, c	1, 2	0, c	0, c	0, c	1, 2
r_6	1, 0	1, 0	0, c	1, 0	0, c	0, c	0, c	1, 1	1, 1	0, c	1, 1	0, c	0, c	0, c	1, 2	1, 2	0, c	1, 2	0, c	0, c	0, c

FIGURE 4: An illustration of channel scanning system CSS R with $m = 7$, $N = 3$ and a relaxed cyclic difference $D = \{1, 2, 4\}$. The pair in the grey box indicates that the network becomes awake on channel c_{css}, which is 0 in the first seven time slots (first segment), 1 in the second seven time slots (second segment), and 2 in the last seven time slots (third segment). The pair in the white box indicates that the network goes into sleep to save energy. The variable c in white box is randomly selected from the set $\{0, 1, \ldots, N - 1\}$.

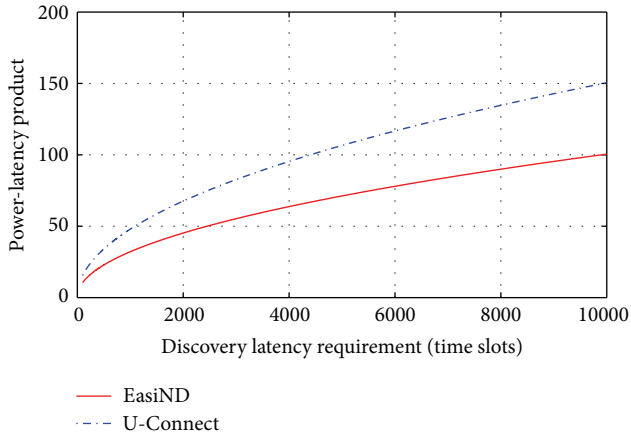


FIGURE 5: The power-latency product for EasiND and U-Connect at various discovery latency requirements.

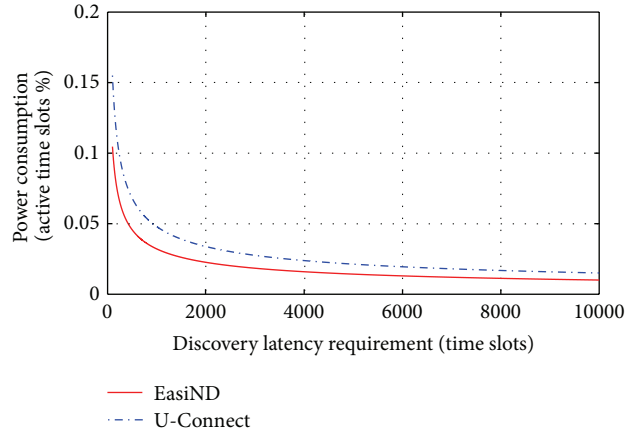


FIGURE 6: The average power consumption for EasiND and U-Connect at various discovery latency requirements.

5.2. Theoretical Analysis for PLP. As presented in the previous section, we adopt slotted neighbor discovery schedules. There are many advantages to employ slotted discovery mechanism as discussed in [3], such as easy implementation and overcoming clock drift problem. Therefore, the average power consumption can be defined as the ratio between active and dormant time slots. According to Theorem 5, we can construct an optimal cyclic quorum system G based on Singer difference set. Therefore, the average power consumption of EasiND constructed by G in period of T is

$$P_e = \frac{k}{k^2 - k + 1}, \tag{16}$$

(a) (b)

Figure 7: Ez240 wireless sensor node.

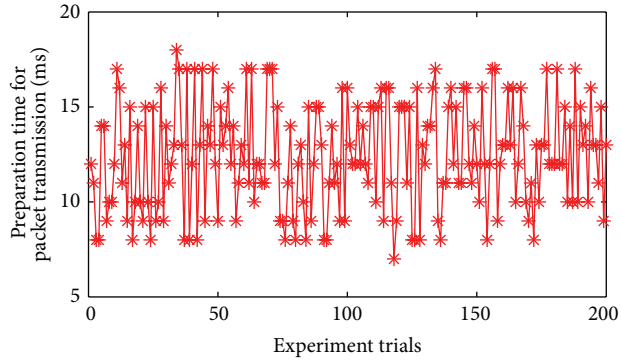
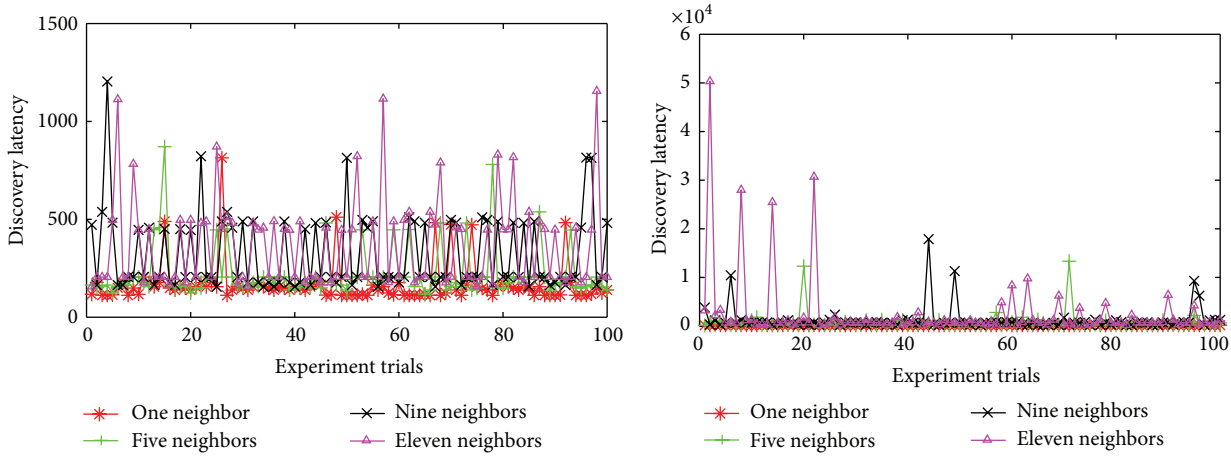
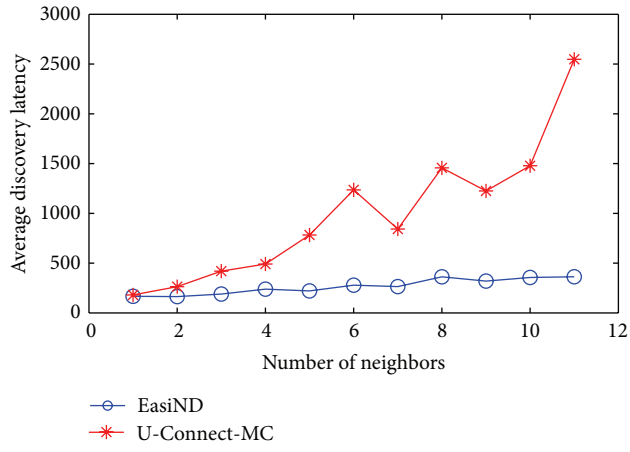


Figure 8: Evolution of preparation time for data transmission. CCA is enabled, and the link-layer retransmission is disabled.



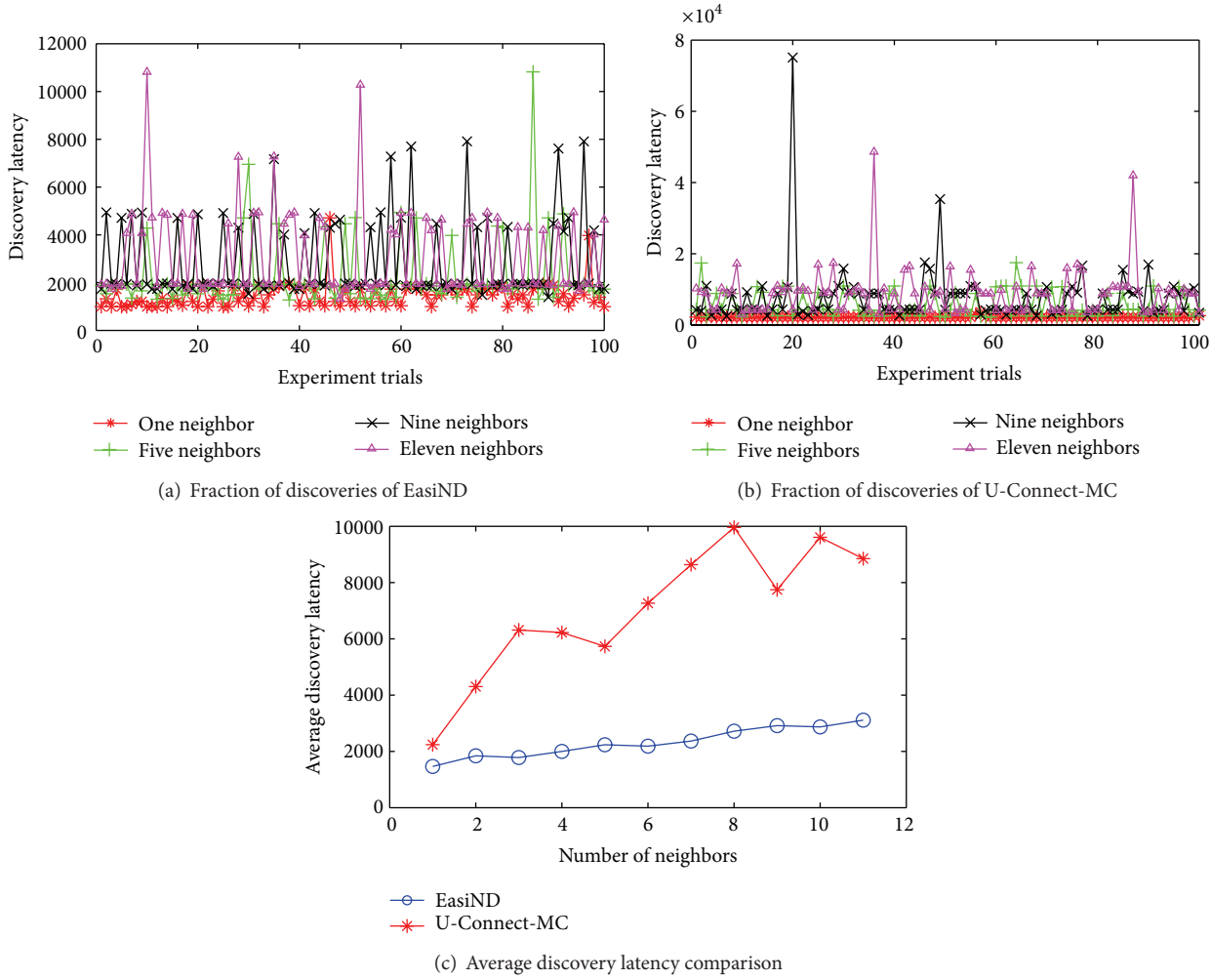
(a) Discovery latency of EasiND

(b) Discovery latency of U-Connect-MC



(c) Average discovery latency comparison

Figure 9: Discovery latency (number of time slots) under different number of neighbors with duty cycle of 12%.



(a) Fraction of discoveries of EasiND

(b) Fraction of discoveries of U-Connect-MC

(c) Average discovery latency comparison

FIGURE 10: Discovery latency (number of time slots) under different number of neighbors with duty cycle of 3%.

where $k-1$ is a prime power and k^2-k+1 is the size of neighbor discovery schedule segment, that is, $m = k^2-k+1$. Then, the theoretical optimal PLP of EasiND is

$$\mathrm{PLP}_e = P_e \times m = k = \sqrt{m - \frac{3}{4}} + \frac{1}{2}. \tag{17}$$

As presented in [3], the PLP of U-Connect is:

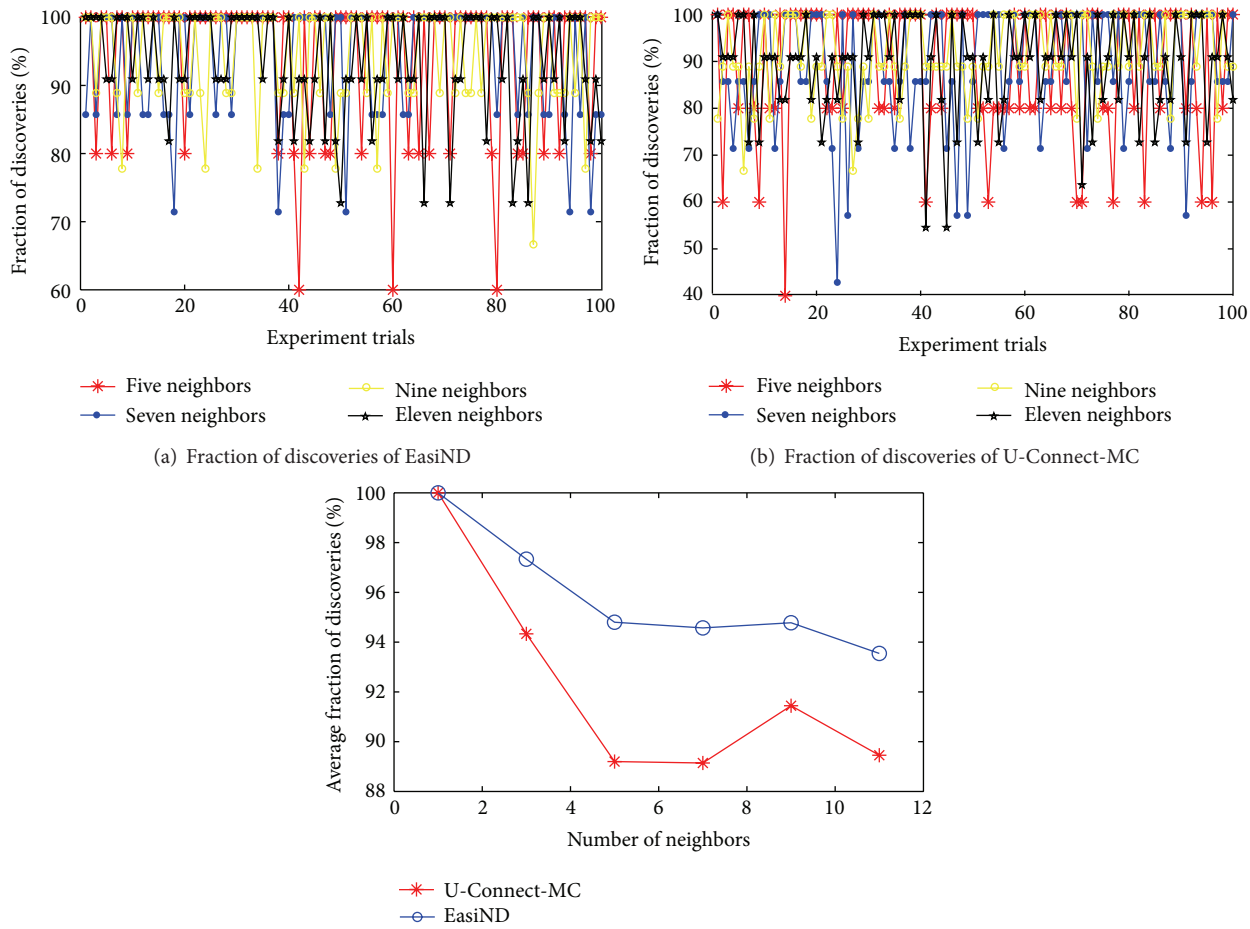
$$\mathrm{PLP}_u = \frac{3p+1}{2}, \tag{18}$$

where p is a prime and $p^2 = m$. So, we have

$$\mathrm{PLP}_u = \frac{3p+1}{2} = \frac{3\sqrt{m}+1}{2}. \tag{19}$$

Therefore, from (17) and (19), we find out that the PLP of EasiND is only two-third of that of U-Connect in theory. Figure 5 shows the power-latency product for EasiND and U-Connect at various discovery latency requirements. Figure 6 illustrates the average energy consumption for EasiND and U-Connect at various discovery latency requirements. The ideal theoretical results demonstrate that EasiND reduces energy consumption and achieves better PLP performance when compared to U-Connect.

5.3. Test-Bed Evaluation. We implement EasiND on our Ez240 wireless sensor node [29], as shown in Figure 7, using TinyOS 2.1 operating system. The Ez240 is equipped with a CC2420 radio [30], which operates on a total of 16 channels in 2.4 GHz ISM band, numbered from 11 to 26. Each of these channels is 2 MHz width with a center frequency separation of 5 MHz for adjacent channels. We also extend and implement the U-Connect on our Ez240 platform for comparison purpose. Here, we refer the extended version of U-Connect for multichannel WSNs as U-Connect-MC. Figure 8 shows the preparation time for data transmission on Ez240 node, which includes the time to turn on the radio, configures transmission frequency, and preprocess data for transmission. We can find that the time needed to prepare for data transmission ranges from 6 to 18 milliseconds under various experiments. We use t_s to denote the length of a time slot. We have made various experiments under different t_s value and found that discovery performance degrades when



(a) Fraction of discoveries of EasiND

(b) Fraction of discoveries of U-Connect-MC

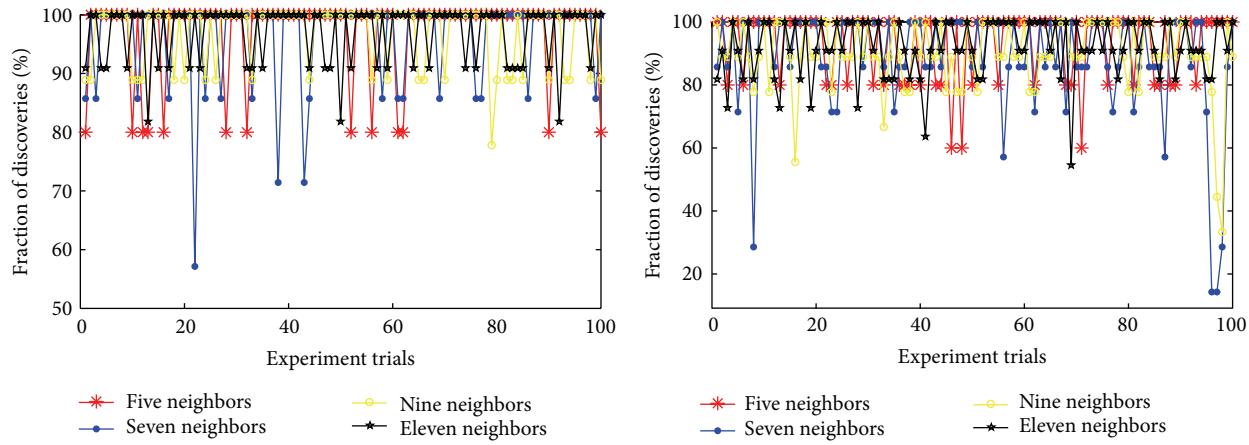
(c) Average fraction of discoveries comparison

FIGURE 11: Fraction of discoveries under different number of neighbors with duty cycle of 12%.

$t_s < 18$ milliseconds. Therefore, we use $t_s = 18$ milliseconds in the rest of the paper unless specified. Moreover, in order to maximize the probability of successful discoveries when time slots are misaligned, we use the same method introduced in [1] that nodes send a discovery message at the beginning and end of the time slot.

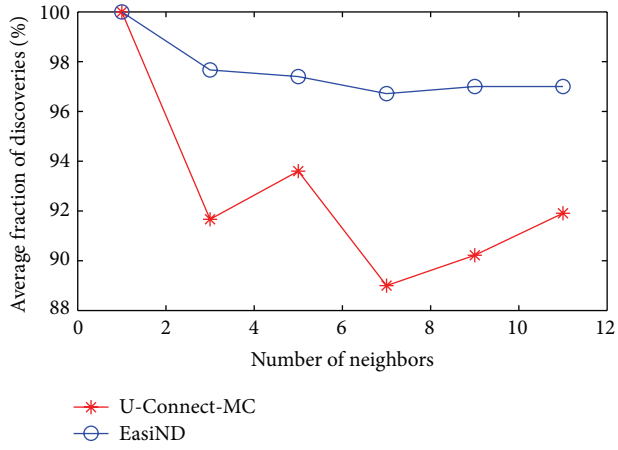
We consider two kinds of duty cycle: 12% and 3%. In case of 12% duty cycle, EasiND uses $m = 95$ and U-Connect uses prime number $p = 13$. In case of 3% duty-cycle, EasiND uses $m = 993$ and U-Connect uses prime number $p = 47$. In both cases, we consider 3 channels, that is, $N = 3$. Specifically, we set $c_{\text{bss}} = 15$ and $c_{\text{css}} \in \{13, 15, 17\}$ in our implementations. We also consider the influence of different number of neighbors on evaluation results.

5.3.1. PLP. Figures 9 and 10 show the discovery latency of EasiND and U-Connect-MC under different number of neighbors with duty cycle of 12% and 3%, respectively. Figures 9(a) and 10(a) describe the discovery latency of EasiND under each experiment trial with different number of neighbors, when nodes operate on 12% and 3% duty cycle, respectively, and Figures 9(b) and 10(b) demonstrate the discovery latency of U-Connect-MC. Each data point in Figures 9(c) and 10(c) is the average measurement from 100 individual experiment trials. As shown in Figures 9 and 10, we can see that EasiND can significantly reduce the discovery latency when compared to U-Connect-MC under all experiment conditions. The reduction ranges from 7% to 86% in case of 12% duty cycle and 35% to 73% in case of 3% duty cycle. The average time slots needed to discover one neighbor for EasiND and U-Connect-MC are about 167 and 180, respectively, with duty cycle of 12%. With the number of neighbors increasing, the time slots used by U-Connect-MC to discover all neighbors increases quickly, but the discovery latency required by EasiND increases much more slowly. For example, in case of 12% duty cycle, the average discovery latency for U-Connect-MC increases to 2547 time slots to discover all 11 neighbors, while EasiND only takes about 363 time slots. With the duty cycle decreasing, the discovery latency of both methods under all network conditions increases. The average discovery latency of U-Connect-MC to discover all 11 neighbors with duty cycle of 3% is 8851 time slots, which is much higher than EasiND's 3112 time slots. According to the definition of PLP, we can easily find that the PLP of EasiND is much lower than that of U-Connect-MC under all experiment scenarios. Therefore, when compared



(a) Fraction of discoveries of EasiND

(b) Fraction of discoveries of U-Connect-MC



(c) Average fraction of discoveries comparison

FIGURE 12: Fraction of discoveries under different number of neighbors with duty cycle of 3%.

to U-Connect-MC, EasiND achieves a much better trade-off between power efficiency and discovery latency.

5.3.2. FD. Figures 11 and 12 show the fraction of discoveries of EasiND and U-Connect-MC under different number of neighbors with duty cycle of 12% and 3%, respectively, including the fraction of discoveries under each experiment trial for both methods in all scenarios as shown in Figures 11(a), 11(b), 12(a), and 12(b). Figures 11(c) and 12(c) show the comparison of average fraction of discoveries measured from 100 independent experiment trials between EasiND and U-Connect-MC. From Figures 11 and 12, we can observe that EasiND can discover more neighbors than U-Connect-MC in a predefined time limitations under all network scenarios. With the number of neighbors increasing, both methods' average fractions of discoveries show an overall decrease trend, but U-Connect-MC is more obvious. For example, U-Connect-MC's average fraction of discoveries decreases to 89.2% from 100% when number of neighbors increase from 1 to 11 in duty cycle of 12%. However, EasiND can discover at least 93.5% neighbors under all network scenarios in duty cycle of 12%. We also observe that both methods achieve better performance in fraction of discoveries when nodes operate on a lower duty cycle. This is because that more collisions would happen due to denser distribution of active time slots when nodes work on a higher duty cycle. Therefore, there is room for further improvement when we take collisions and cooperations between nodes into account, which are left for our future work.

5.3.3. Implications. We have investigated the impact of duty cycle and number of neighbors on neighbor discovery system performances. Both theoretical analysis and experimental results have shown that EasiND consistently outperforms U-Connect under all network scenarios. Because EasiND generally operates in an asynchronous manner, it still works well with misalignment of time slots and clock drift.

6. Conclusions and Future Work

This paper presents a novel asynchronous neighbor discovery method named EasiND for duty-cycled multichannel mobile WSNs. EasiND essentially builds an asynchronous neighbor

discovery system based on cyclic quorum system, which consists of a group of beacon scheduling sequences and channel scanning sequences. EasiND can bound the discovery latency in multichannel communication scenarios and achieves an optimal performance in terms of power-latency product. Both theoretical analysis and test-bed evaluation have shown that EasiND significantly reduces the discovery latency with desired duty cycle by up to 86% and can discover more neighbors in a fixed time limitation when compared to U-Connect. Moreover, EasiND can achieve a better trade-off between power consumption and discovery latency than U-Connect.

As future work, we are pursuing two interesting directions: to (1) extend our study to asymmetric multichannel mobile WSNs and (2) investigate neighbor discovery when considering collisions and cooperations between nodes in multichannel mobile WSNs.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant nos. 61100180, 61202412, and 61272481, the International S&T Cooperation Program of China (ISTCP) under Grant no. 2013DFA10690, and the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant no. XDA06010900. we can construct cyclic quorum system using cyclic difference

References

[1] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys)*, pp. 71–84, 2008.

[2] R. Khalili, D. L. Goeckel, D. Towsley, and A. Swami, "Neighbor discovery with reception status feedback to transmitters," in *Proceedings of the 29th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '10)*, pp. 1–9, March 2010.

[3] A. Kandhalu, K. Lakshmanan, and R. Rajkumar, "U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '10)*, pp. 350–361, April 2010.

[4] W. Zeng, X. Chen, A. Russell, S. Vasudevan, B. Wang, and W. Wei, "Neighbor discovery in wireless networks with multipacket reception," in *Proceedings of the 12th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '11)*, vol. 3, pp. 1–10, May 2011.

[5] A. Purohit, B. Priyantha, and J. Liu, "WiFlock: collaborative group discovery and maintenance in mobile sensor networks," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN '11)*, pp. 37–48, April 2011.

[6] M. Bakht, J. Carlson, A. Loeb, and R. Kravets, "United we find: enabling mobile devices to cooperate for efficient neighbor discovery," in *Proceedings of the 13th Workshop on Mobile Computing Systems and Applications (HotMobile '12)*, pp. 1–6, February 2012.

[7] D. Zhang, T. He, Y. Liu et al., "Acc: generic ondemand accelerations for neighbor discovery in mobile applications," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems (SenSys '11)*, pp. 169–182, 2012.

[8] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: won't you be my neighbor?" in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (MobiCom '12)*, pp. 185–196, 2012.

[9] L. Chen, S. Guo, Y. Shu et al., "Group-based discovery in low-duty-cycle mobile sensor networks," in *Proceedings of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and ad hoc Communications and Networks (SECON '11)*, pp. 542–550, 2012.

[10] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '01)*, pp. 137–145, October 2001.

[11] Y. Tseng, C. Hsu, and T. Hsieh, "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks," in *Proceedings of the 21th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02)*, vol. 1, pp. 200–209, June 2002.

[12] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *Proceedings of the 4th ACM International Symposium on Mobile ad hoc Networking & Computing (MobiHoc '03)*, pp. 35–45, June 2003.

[13] J. Jiang, Y. Tseng, C. Hsu, and T. Lai, "Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks," *Mobile Networks and Applications*, vol. 10, no. 1, pp. 169–181, 2005.

[14] R. H. Kravets, "Enabling social interactions off the grid," *IEEE Pervasive Computing*, vol. 11, no. 2, pp. 8–11, 2012.

[15] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G. Ahn, and A. T. Campbell, "BikeNet: a mobile sensing system for cyclist experience mapping," *ACM Transactions on Sensor Networks*, vol. 6, no. 1, p. 6, 2009.

[16] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[17] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[18] O. D. Incel, "A survey on multi-channel communication in wireless sensor networks," *Computer Networks*, vol. 55, no. 13, pp. 3081–3099, 2011.

[19] O. Akan, O. Karli, and O. Ergul, "Cognitive radio sensor networks," *IEEE Network*, vol. 23, no. 4, pp. 34–40, 2009.

[20] O. D. Incel, L. Van Hoesel, P. Jansen, and P. Havinga, "MC-LMAC: a multi-channel MAC protocol for wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 1, pp. 73–94, 2011.

[21] D. Abdelali, F. Theoleyre, A. Bachir, and A. Duda, "Neighbor discovery with activity monitoring in multichannel wireless mesh networks," in *Proceedings of the IEEE Conference on Wireless communications and networking (WCNC '10)*, pp. 1–6, April 2010.

[22] A. Willig, N. Karowski, and J. Hauer, "Passive discovery of IEEE 802.15.4-based body sensor networks," *Ad Hoc Networks*, vol. 8, no. 7, pp. 742–754, 2010.

[23] N. Karowski, A. C. Viana, and A. Wolisz, "Optimized asynchronous multi-channel neighbor discovery," in *Proceedings of the*

30th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '11), pp. 536–540, April 2011.

[24] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom '09)*, pp. 181–192, September 2009.

[25] K. Bian, J. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom '09)*, pp. 25–36, September 2009.

[26] W.-S. Luk and T.-T. Wong, "Two new quorum based algorithms for distributed mutual exclusion," in *Proceedings of the 17th International Conference on Distributed Computing Systems (ICDCS '97)*, pp. 100–106, May 1997.

[27] C. J. Colbourn and J. H. Dinitz, *Handbook of Combinatorial Designs*, vol. 42, Chapman & Hall/CRC, New York, NY, USA, 2006.

[28] D. R. Stinson, *Combinatorial Designs: Construction and Analysis*, Springer, New York, NY, USA, 2003.

[29] L. Cui, Q. Liu, and D. Li, "The system framework and equipments of the internet of things," *Communications of the China Computer Federation*, vol. 6, no. 4, pp. 18–22, 2010.

[30] Texas Instruments Inc, "2.4 GHz ieee 802.15.4/zigbeeready rf transceiver".

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 123428, 8 pages
http://dx.doi.org/10.1155/2013/123428

Research Article

A Data Transmission Scheme Based on Time-Evolving Meeting Probability for Opportunistic Social Network

Fu Xiao,[1,2,3] Guoxia Sun,[1] Jia Xu,[1,2,3] Lingyun Jiang,[1,2,3] and Ruchuan Wang[1,2,3]

[1] *College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China*
[2] *Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China*
[3] *Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education Jiangsu Province, Nanjing, Jiangsu 210003, China*

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

Received 6 May 2013; Accepted 17 June 2013

Academic Editor: Shukui Zhang

With its widespread application prospects, opportunistic social network attracts more and more attention. Efficient data transmission strategy is one of the most important issues to ensure its applications. As is well known, most of nodes in opportunistic social network are human-carried devices, so encounters between nodes are predictable when considering the law of human activities. To the best of our knowledge, existing data transmission solutions are less accurate in the prediction of node encounters due to their lack of consideration of the dynamism of users' behavior. To address this problem, a novel data transmission solution, based on time-evolving meeting probability for opportunistic social network, called TEMP is introduced, and corresponding copy management strategy is given to reduce the message redundancy. Simulation results based on real human traces show that TEMP achieves a good compromise in terms of delivery probability and overhead ratio.

1. Introduction

Driven by the emergence and application of large number of mobile devices, which are characterized with low-cost, powerful, and short-range communication capabilities, wireless ad hoc network has acquired rapid development. With further research, people started to pay attention to the mobile ad hoc networks, especially those whose communication equipments are deployed on the moving object, such as wildlife tracking network [1], vehicle network [2], and pocket switched network [3]. Traditional communication mode is no longer applicable in these practical application scenarios, due to the regular disruption caused by the sparse deployment, quick movement, and strict constraint both on storage and energy of nodes. Opportunistic network [4], which achieves data transmission via node mobility, appeared in such a situation. As a more natural ad hoc network style, opportunistic network transfers messages through a storage-carry-forward hop-by-hope strategy.

To achieve reliable data transmission for opportunistic network, multicopy technique is usually adopted when the real-time path cannot be guaranteed, for instance, EPIDEMIC [5] and PROPHET [6]; in other words, there are multiple copies of the same message in the network. This will cause data redundancy which affects the network performance. Therefore, efficient data transmission strategy requires effective copy management strategy. In a typical opportunistic network, nodes move randomly and quickly; nevertheless, for opportunistic social network, the mobility of nodes is controlled by human social activities, and the encounters between nodes are more stable and regular, so we can use the history activities of nodes to predict the encounter of nodes in the future in opportunistic network as paper [7] elaborated.

Combined with the characteristic of the opportunistic network as well as the need of copy management, a data transmission based on node time-evolving meet probability which consists of message forwarding and management is proposed in this paper. Message forwarding is divided into three steps: find the periodic neighbor of the destination node, find the appropriate time slot for forwarding, and forward the message to the node which has higher meet probability

with the destination in the right slot. There are two cases for the message management: multicopy strategy is adopted to establish quick contact with the destination in the first step of message forwarding, and for the last two steps message is forwarded in a single copy way to reduce network overhead.

The rest of this paper is structured as follows. Section 1 briefly analyses the related work and our work is elaborated in Section 2. Section 3 presents the simulation and the evaluation and future work of this paper is presented in Section 4.

2. Related Works

A variety of data transmission strategies for opportunistic social network communication is proposed. LABLE [8] proposed by Hui and Crowcroft is the earliest work; the authors think that nodes that belong to the same community have higher encounter opportunity, and they assume that each node has a label to identify their communities; the message is forwarded to the destination directly or by the relay nodes that belong to the same community with the destination. It is inefficient unless the source node can be met directly with the node that is in the same community with destination node. On the basis of LABLE, a community-based data transmission strategy, called Bubble Rap, is proposed in paper [3]. Bubble Rap relied on community and centrality; each node has a local centrality that describes the popularity of the node with its local community and a global centrality across the whole network; it first bubbles the message up based on the global centrality, until the message reaches a node which is in the same local community as destination. The related community detection algorithms, SIMPLE, k-CLIQUE, and MODULARITY are given in paper [9]. Node similarity is defined in [10] to describe the neighborhood relationship between nodes, according to the history of node encounter. Based on the neighborhood relationship, a distributed community detection algorithm is given, as well as a community-based epidemic forwarding. Based on the literature [10], a new social pressure metric (SPM) is introduced in [11] to accurately detect the quality of friendship; this approach considers both direct friendship and indirect friendship to construct its community. It can help to make smarter decisions. Nevertheless, the calculation of metric needs the whole contact information which may be unrealistic for opportunistic network.

All the above work is community-based data transmission strategy. It generates considerable network traffic for community information maintenance overhead. Moreover, most community detection algorithm may lead to the formation of monotonically increasing cluster due to lack of time information; that is, more and more nodes are added to the community with time elapsed, but the outdated nodes cannot be removed from the community timely. On the other hand, community detection takes a long time and brings a "slow start" problem to the network. In summary, community-based data transmission strategy makes good use of community feature of opportunistic social network, but it requires an efficient community detecting algorithm to improve the performance.

At the same time, researchers also proposed a series of data transmission strategy based on node meeting opportunity predicting. PROPHET is a kind of multicopy transmission strategy that can be applied to the opportunistic social network. In the protocol, each node maintains its own transmission probability to the destination and message is forwarded to the node which has a greater meet probability with destination when two nodes meet. There is a "lag" problem when forecasting the encounter probability; furthermore, the excessive copy of the message causes larger overhead. A novel strategy based on node sociability is given in [12]; the key idea is that of assigning to each node a time-varying scalar parameter which captures its social behavior in terms of frequency and types of encounters, and then node forwards message only to the most social nodes. In [13] Mei et al. found that people with similar interests tend to meet more often and then proposed SANE, a social-aware and stateless routing for opportunistic social network; the interest profile of an individual is represented as a k-dimensional vector. The cosine similarity is defined to express the interest similarity between two nodes; a message should be forward to nodes whose interest similar to destination. PeopleRank [14] ranks the node according to the node importance using a similar algorithm as PageRank; node forwards message to destination or a more important node. The author of dLife [15] believes that opportunistic social network should consider the dynamism of users' behavior resulting from their daily routines; each node has two functions: TECD that captures the evolution of social interaction among pairs of users in the same daily period of time over consecutive days; and TECDi that captures the node's importance. The message is forwarded to the encounter if its TECD to destination is bigger than that of the carrier, or its TECDi is higher than that of the carrier when the relationship to destination is unknown. And the literature [16] explored how much delay has to be tolerated for the message delivery from the source to the destination.

3. The TEMP Strategy

3.1. Problem Description. In a typical opportunistic network, node mobiles follow the same pattern (such as random model), so nodes are very similar in terms of data transfer. However, nodes in opportunistic network are mainly human-carried devices whose mobility is controlled by people; therefore, node mobility is distinct, but their encounter is more regular and stable. As described in [15], nodes have different encounter relationship with different nodes at different time periods of the same cycle because of its daily routines. As shown in Figure 1, A has 5 directly meeting nodes, B, C, D, E, and F, if two nodes meet each other in the related time slot, there is a line between them, and the number of encounters is represented by value w on the line. In summary, nodes have similar encounter relation within the same time slot in different cycles but different encounter relation at different time frames of the same cycle; these features should be considered when designing data transmission strategy.

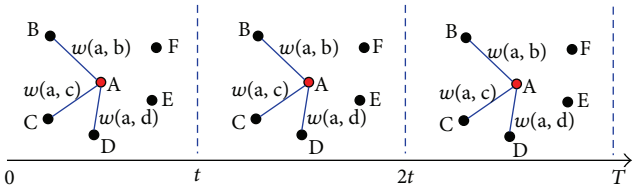


Figure 1: The contact information of A at different time slots within the same cycle.

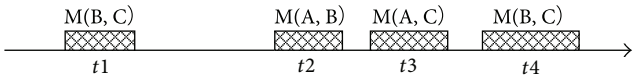


Figure 2: A common scenario of node contact for illustration of the lag problem.

Next, we will first analyze the deficiencies of the existing typical solutions and then present our solution, TEMP forwarding strategy.

3.2. Problem Analysis of Existing Solutions. PROPHET may be first thought of when we try to reflect node time-varying meet probability; in PROPHET the update method mainly includes the increase and decay in the probability. The probability increases according to formula (1) when A meets B and decays in time as formula (2):

$$P_{(A,B)} = P_{(A,B)old} + \left(1 - P_{(A,B)old}\right) \times P_{init}, \quad (1)$$

$$P_{(A,B)} = P_{(A,B)old} \times \gamma^{k}. \quad (2)$$

There is a reaction lag problem in this update method for the prediction of node meeting probability. Consider a common situation shown in Figure 2, at time $t3$; for node B and C, their meet probability decays to a small value after $t3 - t1$, but the probability of A and B increases recently because they meet a while back, then there is $P(A, B) > P(C, B)$ according to PROPHET. Therefore when node A and C encounter at $t3$, C forwards the message, whose destination is B, to A. However we find that it is node C meets node B instead of A in the near future. The message is forwarded toward an incorrect direction just because of the lag problem.

To describe the dynamic probability of contact between nodes, dLife uses TECD and TECDi, two time-varying parameters. First it acquired the average contact length of nodes for each time slot; then for a certain time slot, the meeting probability of two nodes is calculated by the weighted average contact length, with each weighed by a certain coefficient. The main problem of dLife is shown in Figure 3; according to dLife we have TECD(A, D) > TECD(A, C) when C meets D in the first sample slot, so if C has message destination for A, dLife will forward the message to D; however, later we will find that C has more opportunity to meet A in the rest of time slot, so message should be still carried by C. For opportunistic social network, the contact number of nodes is limited, and the more often nodes met before, the less chance they will meet in the future for a certain time slot. Note that, this is different with different cycles; it is universally thought that nodes have chance to meet each other, if they met often in the previous cycles.

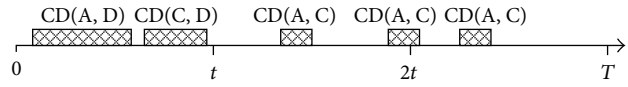


Figure 3: Sketch for illustrating the problem of dLife.

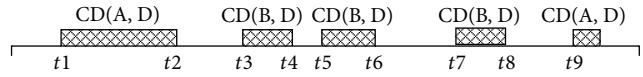


Figure 4: Sketch for elaborating node's contact information.

3.3. Our Solutions. The research results in [7] show that nodes encounter each other with periodic regularity, and node's contact information of last cycle can be used to predict node's meet probability for the next cycle. At the same time dLife pointed out that nodes have different contact relationships for different time slots within the same cycle. Based on the above facts, we make the following assumptions: nodes have varying encounter relationships during different time slots of the same cycle, but for the same time slot in different cycles, node's encounter is relatively stable.

3.3.1. Node's Encounter Probability at Time t within Slot i. First, if the encounter interval of any two nodes is available, the time length between t and the time that node meets destination is acquired; the shorter the length is, the more suitable the node is chosen as the relay node, so we can use formula (3) to represent the meeting probability:

$$P(t)_{(A,B)m} \propto \frac{1}{\left(d_{(A,B)m} - \left(t - t_{(A,B)m-1}\right)\right)} \quad (3)$$

t is the current time, $t_{(A,B)m-1}$ represents the number $m - 1$ meeting time of A and B, and the time interval between number $m - 1$ and number m meeting time of node A and B is $d_{(A,B)m}$. But it is unrealistic to predict the exact time that any two nodes meet at an opportunistic network, even if the information can be obtained; it needs high storage overhead to store this information. Next we will give an approximate solution which is computational and has low storage cost.

We can obtain the contact duration of any two nodes in an arbitrary time slot according to the history information. For a specific time slot i, the contact duration of A and B is relatively stable, which is used to estimate the meet probability by dLife. To make a better prediction, the number and sequence of node's encounter also should be considered. For example, as shown in Figure 4, the shadow rectangles represent the contact duration, the total contact duration of A and B is TCD(A, B), and then we have TCD(A, D) > TCD(B, D). However, for the probability of node's encounter, we should have $P(A, D) > P(B, D)$ before time $t1$, $P(B, D) > P(A, D)$ during $t2$ to $t3$ and $P(A, D) > P(B, D)$ between $t8$ and $t9$.

Based on the above analysis, the method of calculation of node's encounter probability at any time t within the slot i is given as formula (4)

$$P_i(t)_{(\mathrm{A,B})} \propto \left(\mathrm{TCD(A,B)}_i - \mathrm{CD}(t)_{(\mathrm{A,B})}\right) \tag{4}$$

$\mathrm{CD}(t)_{(\mathrm{A,B})}$ is the contact duration of A and B before time t within slot i, and their total contact duration in slot i is $\mathrm{TCD(A,B)}_i$; take the deviation into consideration; if $\mathrm{CD}(t)_{(\mathrm{A,B})} > \mathrm{TCD(A,B)}_i$, $P_i(t)_{(\mathrm{A,B})}$ is set 0. $\mathrm{TCD(A,B)}_i$ is calculated based on the average contact duration in slot i of history cycles, as formula (5), j is the number of historical cycles, and CD(A, B) represents the contact duration of A and B in slot i of cycle k as follows:

$$\mathrm{TCD(A,B)}_i = \frac{\sum_{k=1}^{j} \mathrm{CD(A,B)}_{k,i}}{\sum_{k=1}^{j} k}. \tag{5}$$

3.3.2. Message Forwarding Strategy. TEMP divides a node's active cycle into multiple sampling slots according to node's day-to-day itinerary form. Each node maintains its own contact information of each slot. If two nodes can meet directly, they are neighbor node for each other. A's neighbor node set in slot i is $N_i(\mathrm{A})$, and the total neighbor of A in the entire cycle is $N(A) = \cup N_i(\mathrm{A})$. TEMP forwarding strategy mainly comprises three stages: (1) find the periodic neighbor of the destination node; (2) find the appropriate time slot for forwarding; (3) forward the message to the node which has higher meet probability with the destination in the right slot. Next we will explain the three stages in detail.

(1) Find the periodic neighbor of the destination node.

If the destination node of message does not belong to the neighbor set of the message carrier, the message needs to be forwarded to nodes which have neighbor relationship with the destination as soon as possible. That is, the message carrier node A meets with B, if B satisfies the condition:

$$\frac{(|N(\mathrm{B}) \setminus N(\mathrm{A})|)}{|N(\mathrm{B})|} \geq \lambda. \tag{6}$$

A forwards the message to B, λ is an adjustable parameter, and it is set to 1 in order to reduce the message redundancy in this paper). Messages are forwarded by this way until they reach a node that has neighbor relationship with the destination. Then the message forwarding moves into the second stage.

(2) Find the appropriate time slot for forwarding.

The main purpose of this stage is to find the most recent time slot that the message can be delivered. We define the distance between slot i and j as dis(i, j), which is calculated according to formula (7) as follows:

$$\mathrm{dis}(i, j) = \begin{cases} j - i & j \geq i \\ j + S - i & j < i \end{cases} \tag{7}$$

S is the number of the time slot in a cycle. Note that, dis(i, j) is different from dis(j, i). Assume the destination of message is D. After the previous stage, the message's current carrier A must have neighbor relationship with D; that is, D belongs to A's neighbor collection of a certain time slot i, $\mathrm{D} \in N_i(\mathrm{A})$. when A meets with B at slot k, if there exist $\mathrm{D} \in N_i(\mathrm{B})$ and $\mathrm{dis}(k, i) > \mathrm{dis}(k, j)$, A forwards the message to B. And so it goes on, until the most recent delivery time slot arrives. Then message forwarding goes to step 3.

(3) Forward the message to the node which has higher meet probability with the destination in the right slot.

Message forwarding reaches this stage; it means that the message current carrier can meet with the message's destination D at the current slot i. So if the message's carrier A meets B at time t, if the condition $P_i(t)_{(\mathrm{B,D})} > P_i(t)_{(\mathrm{A,D})}$ is met, A forwards the message to B.

3.3.3. Copy Management. Copy management aims to reduce the message redundancy and the network load. Copy management includes the following two aspects.

(1) In the message diffusion stage, namely, the first step of the forwarding policy, the destination's information is unavailable. In this condition, the message is forwarded based on multicopy strategy; that is, the message carrier still save a copy of the message after it forwards the message to another node. The goal here is to establish contact with destination as soon as possible by the multicopy strategy.

(2) In the last two steps of message forwarding, message carrier has established contact with the destination, so the message is forwarded in a single copy way. The carrier deletes the message once it forwards the message to an appropriate node to reduce the redundancy.

4. Simulation and Evaluation

4.1. Simulation Settings. We develop the TEMP on the DTN simulation platform, ONE1.4.1 [17], developed by the Helsinki University, and also give the performance evolution based on the simulation results. The experimental scenario is based on real human movement trajectory data sets, Cambridge Traces [18]. In order to evaluate the performance of each algorithm, we generate 5000 messages in advance using createCreates.pl, a Perl script of ONE. The source and destination of message are randomly selected, and the message size is evenly distributed between 10 KB and 100 KB. Table 1 shows the main parameters of the simulation.

To evaluate the performance of TEMP, we will compare it to dLife and PROPHET in terms of delivery rate, overhead ratio, and network delay and discuss the results.

4.2. Effect of TTL. In this experiment message buffer size is set to 2 MB. In Figure 5, we can see that the delivery ratio of each protocol develops with the increasing of TTL. dLife and PROPHET achieve high delivery ratio when TTL is less than 1 day, but once the TTL is greater than 1 day, the delivery ratio of TEMP improves significantly and eventually much higher than the other two protocols. In the implementation of TEMP, node's activity cycle is set to 1 day, so when the message TTL is less than 1 day, the performance of TEMP is poor. At the same time, we note that while node's activity cycle in dLife is

Table 1: Default parameter value.

Parameter	Value
Simulation time	990000 s
Update interval	60 s
Number of messages	5000
Message size	10 kB–100 kB
Message TTL	2 day
Buffer size	2 MB
Transmit speed	11 Mbps
ProphetRouter.secondsInTimeUnit	10 s
ProphetRouter.p_init	0.75
ProphetRouter.beta	0.25
ProphetRouter.gamma	0.98
Dlife.numberofslot	24
λ	1

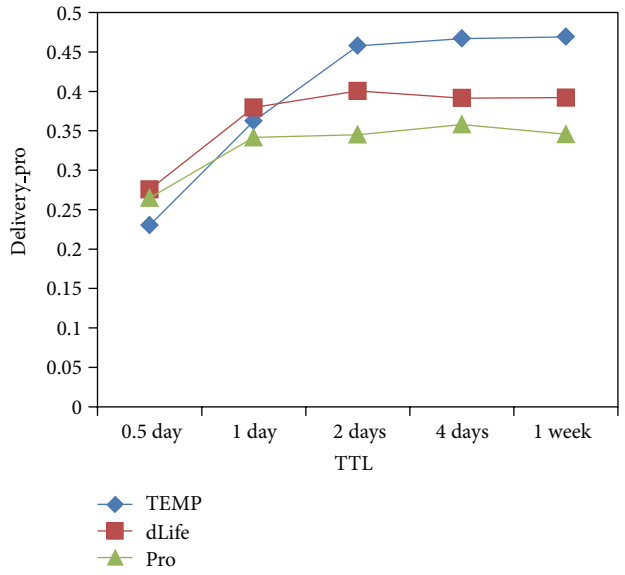


Figure 5: Delivery ratios under varying TTL.

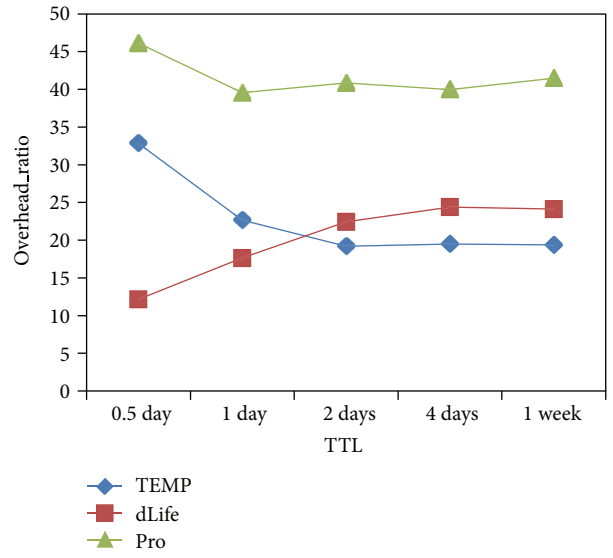


Figure 6: Overhead ratios under varying TTL.

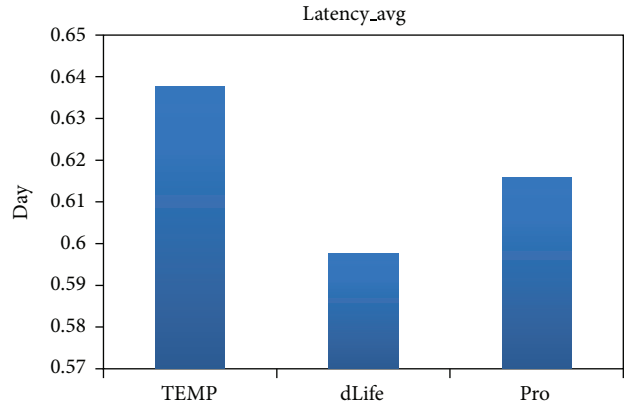


Figure 7: Average network delays.

also set to 1 day, it achieves better performance than TEMP when message TTL is less than 1 day. The reason is that dLife uses the weighted sum of node's contact duration of every slot to calculate TECD and TECDi, and the coefficient of each slot is very close to each other which makes dLife more likely to use the total contact duration to predict node's encounter probability. In a word, it makes little sense to divide the slot for dLife.

The definition of overhead ratio in Figure 6 adopts ONE's default value; that is, overhead_ratio = (relayed-delivered)/delivered. Figure 6 shows that compared with ROPHET TEMP and dLife can reduce the overhead ratio significantly. When message TTL is less than about 1 day, the message forwarding mainly uses multicopy strategy for TEMP, so the overhead ratio of TEMP is much higher than dLife, but it decreases with the addition of message TTL and remains stable finally.

We also observed the delay of each protocol when the TTL changes. In the experiments, we found that when the TTL increases, the delay of each protocol changes slightly. So we use the average of network delay in different TTL settings for evaluation and plot them in Figure 7. We find that even though the average delay of TEMP is higher than that of dLife and PROPHET, the difference among them is less than 0.04 day (about an hour) which is acceptable in opportunistic network.

4.3. Effect of Buffer Size. We studied the effect of buffer size on the performance in this part; TTL value is set to 2 days. The results of the experiment are showed in Figures 8–13.

The plot shows that TEMP outperforms all the other forwarding schemes on delivery ratio. In normal conditions, the usage of multicopy strategy can improve message delivery ratio. However, network resources, such as storage, energy, are limited in opportunistic social network; a large number of message copies bring huge resource consumption which will reduce network performance and bring high overhead

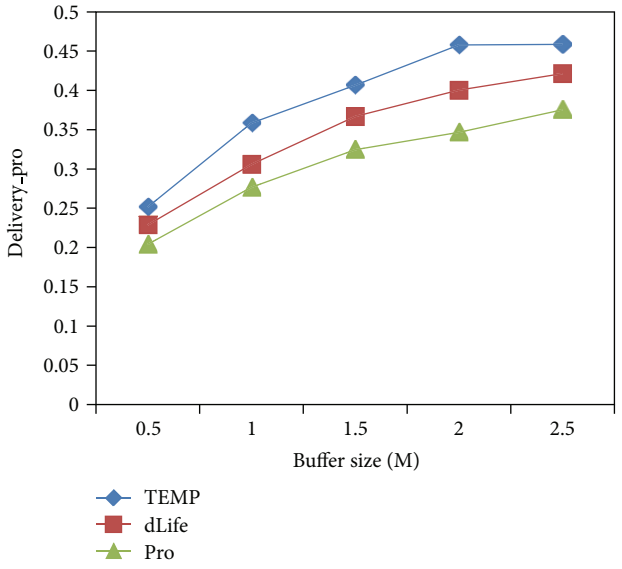


Figure 8: Delivery ratios under varying buffer sizes.

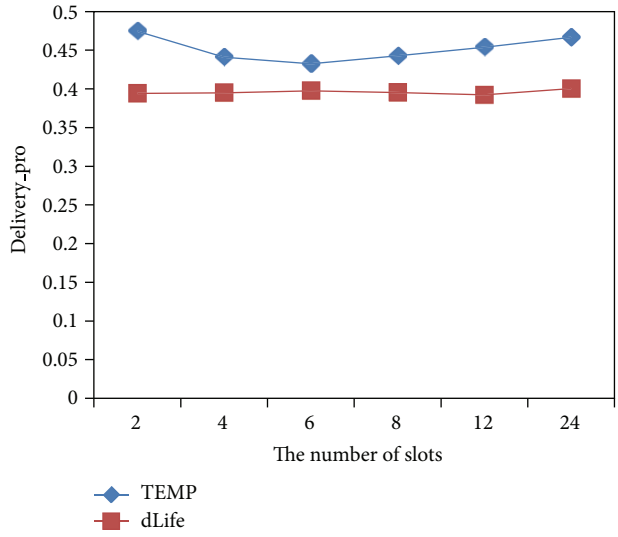


Figure 11: Delivery ratios under varying number of slots.

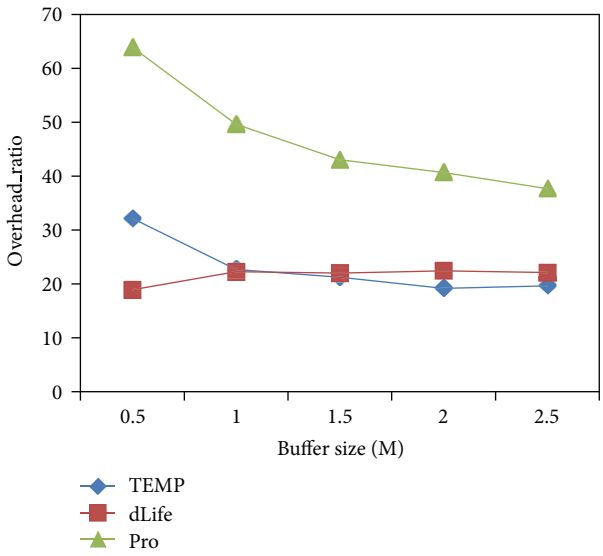


Figure 9: Overhead ratios under varying buffer sizes.

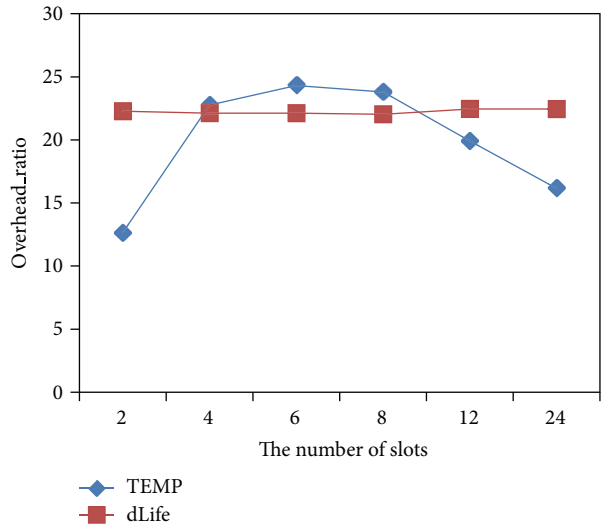


Figure 12: Overhead ratios under varying number of slots.

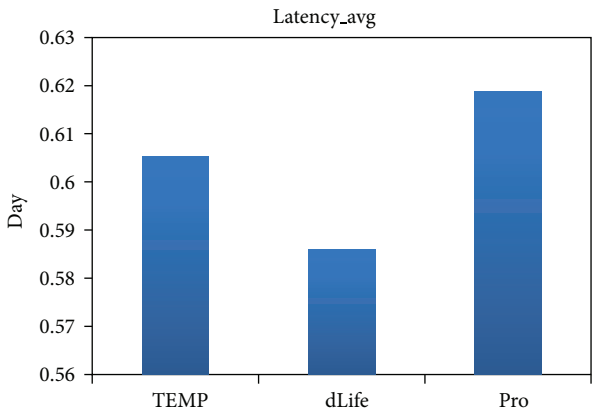


Figure 10: Average network delays.

ratio as is shown in Figure 9; therefore, the delivery ratio of dLife and PROPHET is poor. Moreover, the "lag problem" of PROPHET further decreases the delivery ratio.

Figure 10 shows the average delay of the three forwarding schemes under different buffer sizes. dLife takes node's global importance into consideration and uses TECDi (similar to PageRank algorithm) to make message forwarding decision, so it achieves the best performance on delay. Note again, the calculation of a node's importance metric TECDi relied on the prior obtaining of its neighbor's importance which is very complex. TEMP only uses node's local information and avoids this disadvantage. Overall, the difference of delay, less than 0.03 day (about 0.72 hour), among them is very slightly for opportunistic social network with delay tolerance capacity.

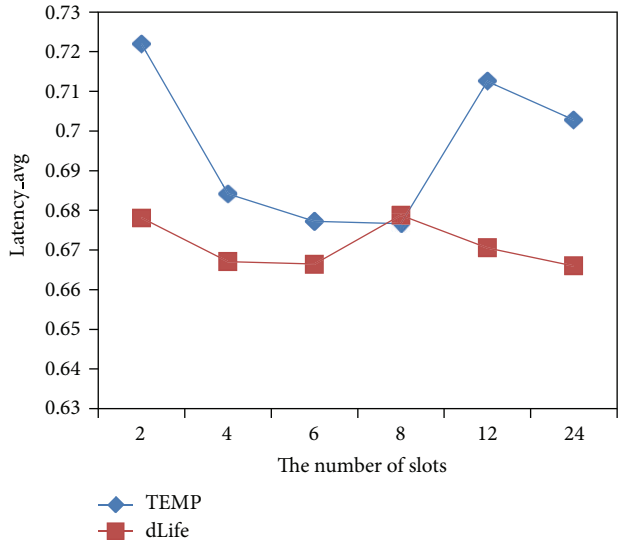


Figure 13: Average delays under varying number of slots.

4.4. Analysis of Time Slot Division. In paper [15], the author pointed out that dLife achieved the best performance when the cycle was divided into 24 time slots. In all experiments above, we used 24 time slots in the implementation of dLife. But for TEMP, we adopt 4 time slots, it should be noted that this division is not optimal for TEMP, and we adopted 4 time slots because we take the two main periods, namely, 6:00 am–12:00am and 12:00 pm–18:00 pm, of human activities into consideration. In fact, the division of time cycle should be based on a large number of observations, but in this paper we divided the time cycle into multiple slots of equal length for convenience. In the following parts, we talk about the effect of time slot division on TEMP and dLife.

The curves in Figure 11 show that the delivery ratio performance of TEMP is always higher than that of dLife under different time cycle divisions. From all the three figures above, that is, Figures 11, 12, and 13, we can find that the performance of dLife changes slightly. This further proves that it makes little sense for dLife to divide time slot although its original intention is to reflect node's different encounter relationship by dividing time slot. Compared to dLife, TEMP is more sensitive to slot division which makes it more suitable for opportunistic social network that node's activities are obviously different during different time slots.

5. Conclusion and Future Work

In this paper, we propose a data transmission strategy, called TEMP, for opportunistic social network. It consists of message forwarding and copy management strategy. Simulation results show that TEMP is more efficient in terms of delivery ratio and overhead ratio. Simultaneously it is more suitable for the scene in which node's activities show a significant difference during different time slots. It is well known that the community is a very important feature of opportunistic social network, but the existing community detection algorithm is nonadaptive and complex. Thus, in the future work, we aim to study a computable and self-adaptive community detection algorithm to assist the message forwarding. Such as, we can use GPS or RFID to locate the community that the node is currently in; of course, the community is a geo-community, but for offline network, node's mobility preference is always associated with the geographic information.

Acknowledgments

This work is sponsored by National Natural Science Foundation of China (61003236, 61170065, 61100199), Scientific & Technological Support Project of Jiangsu (BE2012755), Scientific Research & Industry Promotion Project for Higher Education Institutions (JHB2012-7), and a Project funded by Priority Academic Program Development of Jiangsu Higher Education Institutions (Information and Communication, YX002001).

References

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

[2] U. G. Acer, P. Giaccone, D. Hay, G. Neglia, and S. Tarapiah, "Timely data delivery in a realistic bus network," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1251–1265, 2012.

[3] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 11, pp. 1576–1589, 2011.

[4] Y.-P. Xiong, L.-M. Sun, J.-W. Niu, and Y. Liu, "Opportunistic networks," *Journal of Software*, vol. 20, no. 1, pp. 124–137, 2009.

[5] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Tech. Rep. CS-2000-06, Duke University, 2000.

[6] L. Anders and D. Avri, "Probabilistic routing in intermittently connected networks," *ACM Mobile Computing and Communications Review*, vol. 7, no. 3, pp. 19–20, 2003.

[7] N. Eagle and A. S. Pentland, "Eigenbehaviors: identifying structure in routine," *Behavioral Ecology and Sociobiology*, vol. 63, no. 7, pp. 1057–1066, 2009.

[8] P. Hui and J. Crowcroft, "How small labels create big improvements," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications Workshops*, pp. 65–70, Washington, DC, USA, March 2007.

[9] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proceedings of the 2nd ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch '07)*, Article no. 7, New York, NY, USA, August 2007.

[10] F. Li and J. Wu, "LocalCom: a community-based epidemic forwarding scheme in disruption-tolerant networks," in *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '09)*, Rome, Italy, June 2009.

[11] E. Bulut and B. K. Szymanski, "Friendship based routing in delay tolerant mobile social networks," in *Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM '10)*, Troy, NY, USA, December 2010.

[12] F. Fabbri and R. Verdone, "A sociability-based routing scheme for delay-tolerant networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, Article ID 251408, 13 pages, 2011.

[13] A. Mei, G. Morabito, P. Santi, and J. Stefa, "Social-aware stateless forwarding in pocket switched networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 251–255, Shanghai, China, April 2011.

[14] A. Mtibaa, M. May, C. Diot, and M. Ammar, "PeopleRank: social opportunistic forwarding," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '10)*, San Diego, Calif, USA, March 2010.

[15] W. Moreira, P. Mendes, and S. Sargento, "Opportunistic routing based on daily routines," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 1–6, 2012.

[16] Y. Zhu, H. Zhang, and Q. Ji, "How much delay has to be tolerated in a mobile social network," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 358120, 8 pages, 2013.

[17] The ONE 1.4.1 [EB/OL], Nokia Research Center, Helsinki, Finland, 2010, http://www.netlab.tkk.fi/tutkimus/dtn/theone/.

[18] J. Scott and R. Gass, "Crawdad trace cambridge/haggle (v.2006-09-15)," 2006, http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/content.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 390795, 9 pages
http://dx.doi.org/10.1155/2013/390795

Research Article

A Routing Algorithm Based on Dynamic Forecast of Vehicle Speed and Position in VANET

Haojing Huang[1,2] and Shukui Zhang[2,3]

[1] *Department of Computer Information Engineering, Guangdong Technical College of Water Resource and Electric Engineering, Guangzhou 510635, China*
[2] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*
[3] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

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

Received 12 April 2013; Accepted 29 May 2013

Academic Editor: Yong Sun

Considering city road environment as the background, by researching GPSR greedy algorithm and the movement characteristics of vehicle nodes in VANET, this paper proposes the concept of circle changing trends angle in vehicle speed fluctuation curve and the movement domain and designs an SWF routing algorithm based on the vehicle speed point forecasted and the changing trends time computation. Simulation experiments are carried out through using a combination of NS-2 and VanetMobiSim software. Compared with the performance of the SWF-GPSR protocol with general GPSR, 2-hop C-GEDIR, and the GRA and AODV protocols, we find that the SWF algorithm has a certain degree of improvement in routing hops, the packet delivery ratio, delay performance, and link stability.

1. Introduction

With the development of mobile communication technology and intelligent transportation technology, even if the drivers do not intervene, any vehicle can participate in collecting and reporting useful traffic information such as section travel time, flow rate, and density, as discussed in [1]. Road conditions, traffic congestion and information can be exchanged between vehicles, including speed, direction, acceleration, and position, which can greatly improve the vehicle safety. Business services (such as infotainment, audio, video, and internet applications) also can be applied in this transportation; in this case, even if people leave the computer and mobile phone, they can sit in the car to get the leisure and entertainment. The realization of these ideas comes from VANET (vehicle ad hoc network technology) in [2].

The VANET vehicles must be equipped with a radio transceiver and computer control module, so that they can be used as a network node. The wireless network coverage range of each vehicle may be limited to a few hundred meters; each node can be either a transceiver or a router. Therefore, it is necessary to provide end-to-end communication over long distances by means of intermediate nodes. The VANET does not need the support of wired infrastructure, but some of the permanent network nodes may appear in the form of a roadside unit. Roadside units can provide a wide variety of vehicle network services, such as the placement of information in the driveway side, providing geographic data information, or using buses, taxis, and other vehicles as a gateway to connect to internet as discussed in [3].

Ad hoc network is a kind of distributed wireless multihop network composed of a group of nodes with routing function, and it does not rely on any of the default network infrastructures. In ad hoc network, the transmission range of nodes is limited. When the source node sends data to the target node, it usually requires other auxiliary node, so routing protocol is an indispensable part of the ad hoc network as discussed in [4]. Traditional data aggregation schemes for wireless sensor networks usually rely on a fixed routing structure to ensure that data can be aggregated

at certain sensor nodes. However, they cannot be applied in highly mobile vehicular environments. Unlike the traditional ad hoc networks [5], VANET has some different characteristics such as regularity and predictability [6]. The global positioning system (GPS) can provide precise positioning service for vehicles. However vehicle is moving at high speed, obstruction in existence, and frequent changes in network topology, so VANET routing leads to more complex routing problems. From a functional perspective, the routing protocol is a mechanism by which communication network sets guidelines for the business data from the source node to the destination node. The design objectives of routing protocol are to meet the application requirements while minimizing the network energy consumption, to use resources effectively, and to expand the network throughput. Among them, the application's needs generally include delay, delay jitter, and packet loss rate. And the network capacity can be seen as a function, which is related to each node's available resources, the number of nodes in the network, node density, the end-to-end communication frequency and topology changes, and other factors, as discussed in [7]. VANET routing protocols have three common classifications: greedy perimeter stateless routing (GPSR) routing algorithm is an algorithm using location information routing, which uses a greedy algorithm to establish the routing in [8]. As discussed in [9], resource-constrained mobile sensors require periodic position measurements for navigation around the sensing region.

2. Problem Statement

GPSR protocol is a stateless routing protocol and a typical location-based routing protocol; it requires neither regular exchange of routing information nor broadcast flooding to route requests. And it does not need to store routing information, thus reducing the network bandwidth resource consumption and the processing power of node, which is very effective in a network for high density and node average distribution, such as a highway. But in recent years, many researchers have found that implementation of GPSR in the city road environment leads to some defects as the following aspects.

2.1. The Unstable Link and a Huge Amount of Computation. The GPSR protocol uses greedy forwarding strategy, selecting the nearest neighbor node from the destination node as the next hop node in the network, which means that it is the farthest from neighbor nodes to the local node, so the link is unstable in a large probability. Because of the frequent movement of node in the network, the speed of the vehicle will result in changes of the distance between the nodes. Also the greedy algorithm cannot be effectively computed, so the unstable link will lead to decline in the performance of the protocol. Changing topology makes the computation of the greedy algorithm large, but the computing power and energy of node is limited; in this case, GPSR protocol needs flattening to eliminate the cross-links in the network topology, so when each node adds or deletes any one of its neighbor nodes, it requires to flatten the local topology. The time complexity of this process is (n^2), in which n is the density of the node's neighbors. When the network topology changes frequently, this problem will become more serious.

For example, source node is X, and the destination node is Y. If, at a time T_0, the node X needs to send data packet to the node Y, and node X knows the location of the node Y at this time, then according to the GPSR protocol, the source node and the intermediate node can transmit directly packet forwarding based on position information of the destination node at that moment. However, the destination node is moving in the actual scene, and the intermediate node is likely to transmit packet forwarding to the destination node which has moved to another position based on the position information of the destination node at the time T_0. Then to route addressing continues, an error may occur. It is likely that the data packet arrives at the position of the destination node in the time T_0. If the surrounding node has no position information of the destination node, then the data packet cannot be transmitted to the destination node, which means packet forwarding failure; in other words, packet will be lost.

2.2. Packet Loss and the Increase in the Number of Routing Hops. The greedy forwarding policy and boundary forwarding strategy of the GPSR protocol will lead to redundancy problem. The uneven distribution of the nodes in a city environment will lead to network congestion and packet loss when the load of network is too large, and the wrong greedy forwarding will take up too much resources. The GPSR algorithm uses the right hand to search routes at the boundary in a direction; it may choose relatively long routing paths to reach the destination node, even nearer routing path in the other direction exists. In fact, when the density of network nodes increases, the boundary forwarding policy leading to routing hops increases, resulting in waste of the channel resources as well as the increase in network delay. When the distribution of network nodes is sparse, because of the use of unified right-hand rule, a message in a boundary mode is likely to repeatedly miss the opportunity to change back to the greedy forwarding mode, which needs to go through a lot of mode nodes.

For example, because of the limitations of the cruciform road topology, node distribution is very uneven. The source node X needs to select a node Y as the next hop node; if the node Y is the best host, then it is transferred to the boundary forwarding mode. Then implementing the right-hand rule, there will be two results: one is that packet will eventually arrive at the destination node, but the results of this addressing increased routing hops; another is that there is large number of nodes on the left side of the node Y, then the packet may have been sent along the left hand side of the road to go farther and farther away from the target until life cycle of the last node is reduced to 0, and then discard the packet.

3. The Algorithm Design

3.1. Relative Speed Fluctuation and Distance Change between Nodes. Real-time computing needs to constantly recalculate the route for change of dynamic speed, which brings

about enormous computation. Due to high-speed mobility of node, speed is roughly between 5~40 m/s, resulting in rapid changes in the network topology, so the life of path is short. For example, the average speed is 100 km/h on the road; if the node's coverage radius is 250 m, then the probability of the persistent link exists for 15s is only 57%, as discussed in [10]. In [11], with the knowledge of the access point locations, more accurate localization for the mobile device trajectories and motion sensors can be obtained. In [12], bounding-box mechanism is a well-known low-cost localization approach for wireless sensor networks. However, the bounding-box location information cannot distinguish the relative locations of neighboring sensors.

Therefore, before the route discovery, forecast of the relativly more stable links is important. The vehicle nodes with relativly more stable speed and trajectory can improve the efficiency of the greedy algorithm and reduce the computational energy consumption. The greedy algorithm selects the nearest node from the target node to send forwards. However, because the vehicle is in a state of movement or stoppage at any time, the speed of the node is changing, so the distance between the nodes is also changed. According to the research, for a car traveling in the big city, the average speed is about 30 km/h, depending on traffic conditions. In general, the limit of maximum speed is 60 km/h on urban trunk road, is 40 km/h on urban secondary roads, and is 30 to 40 km/h in ramp and tunnel is. There are two major possibilities about the vehicle speed changes, one is road congestion, and the other is wait for traffic lights or parking in intersection. In either case, we can consider only the relative velocity between the vehicle nodes. If the speed of nodes is the same or vehicles park simultaneously at the same time, the relative speed is 0. In [13], the difference in propagation speed helps to report encounters between nodes. Therefore, if the vehicle is accelerating or decelerating, the relative speed increases or decreases, the distance between nodes changes, then the shortest path between the nodes also changes.

If the relative speed of the vehicles between nodes is greater than the speed threshold value V_z, forwarding node may exceed the signal radiation range of the source node in the time threshold T_z; the packet should be considered to be sent to another one. However, the speed of the vehicle is unpredictable; we will design a heuristic algorithm to forecast the speed fluctuation to predict the movement of the vehicle. Some improved GPSR routing protocol will take into account the direction of movement of the vehicle. Reference [14] proposed a mobile target tracking scheme that takes full advantage of Voronoi diagram boundary to improve detection ability. In fact, no matter vehicles traveling to the same direction or opposite, and the vehicles are in front or behind of each other, they cannot affect the distance between nodes in greedy routing algorithm if we can predict and compute the relative speed of vehicles. The method can also forecast multiple nodes within a topology; if the majority of nodes of the entire topology are in a stable state of relative speed, the problems caused by the network topology changing too quickly can be improved to some extent.

For example, in the time of $t_{(x)}$, the speed points forecasted of source node A were 20, 23, 25, 15, and 10 m/s, the speed points forecasted of forwarding node B car were 23, 21, 20, 18, and 19 m/s, and then the relative speed of the two vehicles were 3, 2, 5, 3, and 9 m/s, whereas the relative distance were $3t_{(1)}$, $2t_{(2)}$, $5t_{(3)}$, $3t_{(4)}$, and $9t_{(5)}$ m. If the relative distance was more than communication circle radius R of A, A cannot communicate with B at this moment. If within the time threshold T_z, A was unable to send data packet to B, the relative speeds was unstable; consider giving up point B from routing. Conversely, within the time threshold value T_z, the forecast result of the relative speed leading to the distances between B and A was less than R; it can be considered that the node B relative to A was stable and can be used as the next hop of the routing. If there were multiple stable nodes or no stable node, implement greedy forwarding directly. The change in distance caused by speed fluctuated as shown in Figure 1.

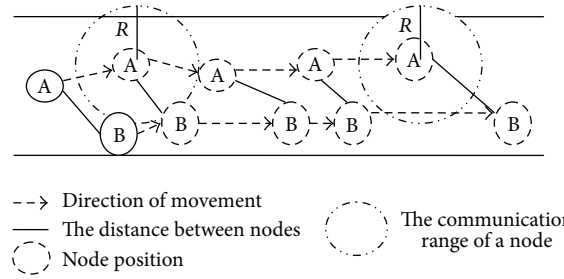


Figure 1: The change in distance caused by speed fluctuation.

3.2. Ideal and Optimal Forwarding Nodes and Movement Domain. In [15], the resultant formulation of the DGPR (distributed Gaussian process regression) approach only requires neighbor-to-neighbor communication, which enables each sensor node within a network to produce the regression result independently. When node A found the next forwarding node B, B node was located the boundary of the circle the radius was R of communication range of A; we considered that B was optimal forwarding node of A. Similarly, if C was the optimal forwarding node of B, D was optimal forwarding node of C; we considered this particular routing to be called continuous ideal and optimal routing, which was the shortest distance routing at this time.

As discussed in [16], localization is one of the most important issues in wireless sensor networks. But no orderliness for locations of nodes changes at any time when vehicles travel on the road. The emergence of continuous ideal and optimal routing was unlikely. But the possibility of the approximate continuous ideal and optimal routing is there. When there was the closest distance between the target node and B, which was infinitely closed to communication circle radius R of A, we can take the point B as the approximate optimal forwarding node. If the infinitely close to ideal and optimal forwarding nodes in the process of forwarding information can be computed, the number of routing hops can be declined. However, since the vehicle is traveling without the law can be followed, we designed a motion range computation method, used to predict the movement trajectory of the vehicles. In the VANET, mobile nodes are subjected to the restrictions

of the road models. Although it is not possible to accurately compute the position of the destination node at a particular moment, it can estimate the range of its movement. Reference [17] presents a novel probabilistic framework for reliable indoor positioning of mobile sensor network devices. As discussed by LAR protocol, the source node X is assumed to know that the position of the destination node D is $L(X_x, Y_y)$ at this moment, and the average moving speed is V_p. Therefore the region where the node D may occur can be estimated. The region is the circular region which used $L(X_x, Y_y)$ as the center, $R = V_p \times (T_1 - T_0)$ as the radius. It is the "desired domain." However, the error may be very serious which uses average speed to compute the mobile range of vehicles.

When implementing speed fluctuation forecasted to predict the curve of the speed, distance between the node A and B can be computed at $t_{(x)}$ time. However, the vehicle traveling on the road does not necessarily move straight. In fact, in most of the cases, the vehicles are doing the curvilinear motion. So directly compute the straight-line distance between the two nodes that will certainly cause serious errors. Therefore, if the position coordinates of the A and B were A (x_A, y_A) and B (x_B, y_B), when $x_A = x_B$, then the positions of A and B were on back and forth of a straight line; the straight line distance can be computed; if $y_A = y_B$, A and B located on the road around the location and straight-line distance can also be computed. Since communication range of a single hop node is only a few hundred meters in [18], the width of the urban road W is typically much less than the signal transmission range of vehicle node. When $y_A = y_B$, it is possible to compute its distance, because the greedy algorithm should select the node from which to the source node is the farthest away. If $x_A = x_B$, the distance K should be computed; K-value is the closer to the optimal node, the possibility of which is the forwarding nodes is great. Therefore, the average speed V_p cannot make the diameter $2R$ of the desired domain exceed the road width W; if the diameter is longer than the width, the maximum speed V_t in the curve should be loop selected to compute until $2R = 2V_p \times (T_1 - T_0) \leq W$; the array of speed will achieve stability and convergence. The packet forwarding of original GPSR protocol is based on the position information of the destination node; the node is no longer forwarding the packet to a fixed position coordinates, but to the region where the destination node may occur, that is, forwarding the packet to a moving area; we call it "movement domain."

The localization problem in mobile sensor networks is a challenging task, especially when measurements exchanged between sensors may contain outliers, that is, data not matching the observation model in [19]. If the continuous ideal and optimal routing exist in the case, the signal packet transmission speed was V_x between the four nodes: A, B, C, and D, the radius of signal transmission circle was R_x, and the time of A sending the data to the D directly was $R_A/V_A + R_B/V_B + R_C/V_C + R_D/V_D$. Based on the speed fluctuations forecasted algorithm, in that period of time, we have to select the speed point forecast: V_A, V_B, V_C, and V_D, the distance is the shortest among the four nodes, the instant speed of four nodes was selected from the speed forecasted queue.

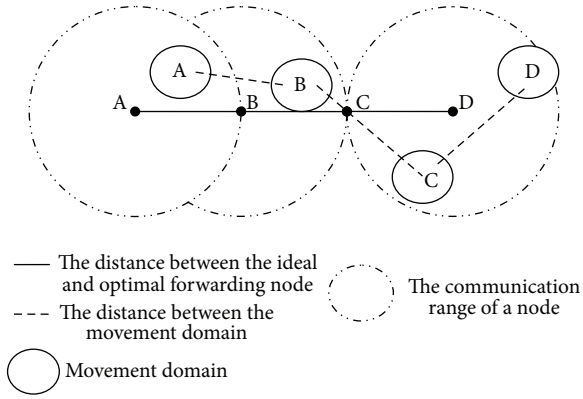


Figure 2: The optimal forwarding nodes and node movement domain.

The optimal forwarding nodes and node movement domain as shown in Figure 2. As shown above, the vehicle is not a linear movement, select the speed forecasted to compute movement domain of nodes, we can predict the range of movement of the vehicles. Therefore, if traffic is heavy, when signal packet transmission range of the vehicle is greater than the distance between nodes, the packet can be directly forwarded to the location of the movement domain. If there is less traffic on the road, the distance between nodes may be greater than the diameter of signal packet transmission circle, which may result in the inability to send the data. At this time, the greedy algorithm can use border forwarding to search forward node.

3.3. SWF Routing Algorithm. The basic principles of GPSR greedy forwarding algorithm are when the node is in greedy forwarding mode, source node select nodes of the farthest distance from themselves within communication range, which is the nearest to the destination node as the next hop node. As discussed in [20], select the node nearest to the destination node as a relay node within the transmission range so that to increase the possibility of a local maximum and link loss because of high mobility and urban road characteristics. So, we design a kind of SWF (Speed Wave Forecasted) routing algorithm combined with speed fluctuation forecasted and computation of the movement domain, as the speed fluctuations forecasted algorithm needs to input the data of speed fluctuations of the vehicle. The vehicle begins to move, to start route discovery, first greedy forwarding. Begin to forecast the speed of the vehicle which is computed at the time threshold value T_Q. Because the possibility of movement changes of the stable relative speed of nodes is small, so put them in a routing node queue and implement greedy forwarding. At the same time, compute distance of approximate ideal and optimal nodes and movement domain. If the distance between movement domains of nodes is smaller than greedy forwarding distance between nodes, then select the approximate ideal and optimal node as forwarding node. Conversely, the greedy algorithm will select the node which is the nearest from target node as forwarding node. When there is a problem of routing void, not implementing

boundary forwarding at once, but waiting storage for the time T, waiting for the forwarding node P reaches the mobile domain; if it arrives, it is forwarded to the P, then continue to greedy forwarding, or into the border mode. When the nodes are into the border forwarding mode, if there are no neighbor nodes which are closer from the movement domain of the destination node than their own, so construct flat graph, using the right-hand rule, and find the closest to the movement domain within the transmission range of the node as the next hop node. Until the border forwarding is finished, return to the greedy forwarding.

3.4. Speed Fluctuations Forecasted Algorithm. Assuming GPS is installed in each vehicle, vehicles can determine their own location information. Electronic map can provide the actual road information in each of the navigation system. Determine the travel path of the vehicle before departure from automatic GPS navigation. For a car traveling in the city roads, the speed change can be seen as a volatile process. Whenever vehicles's continuous acceleration last time n, then the speed is bound to enter a downward spiral. Its decline is due to aforementioned road congestion, traffic lights, speed limits, and other objective factors. Conversely, when the vehicle speed reaches the lowest point, wave of continuous acceleration fluctuations also occur; when the fluctuations reach the steady state, if the continuous distribution time K of the speed can be forecasted, we can know the relative speed between the vehicles; it can be found as a relatively stable routing and associated collection of nodes.

Definition 1. Due to more attributes impacting velocity fluctuations, we set up the N tuple: $\langle V_{\lim}, M, \mathrm{B}\ldots\rangle$ with an uncertain definition, among which $V_{\lim}$ is the maximum speed of the vehicle, M is road congestion, and B is probability of randomly moving. In a curve graph showing speed trend fluctuations, the velocity curve as a circular is divided into aliquots n. According to the fluctuation of curves, the change in angle of the waves can be found and is called circle changing trends angle, in order to compute the time of continuous and stable vehicles speed last out, which is Speed fluctuations changing trends time. Set speed trend wave forecasted as (I, I, II, II, III, IV, V, VI, VII...); time balance point of trend wave as the value of X-axis, its identification symbol is X_T, X belongs to W; speed balance point of trend wave as the value of Y-axis, its identification symbol is s.

3.5. Speed Point Forecasted of the Speed Trend Wave. As discussed in [21], a number of routing protocols have been developed in the last years based on SI principles, and, more specifically, taking inspiration from foraging behaviors of ant and bee colonies. The idea of speed forecasted comes from the wave fluctuations in equilibrium theory in [22]; when the continuous decline of the I wave speed is complete, the number of speed point forecasted of the increase wave II is usually π times of wave I; the very small number of cases may exceed π times. Top speed of vehicle node on road of the first wave change of velocity is 20 m/s; the minimum speed is 5 m/s. When II = III, top speed of wave II is s (II) = 62.8 m/s; minimum speed is about 15.7 m/s. I/2 = $(2/2\pi) \times$ II, use $\pi/2$ as a constant, can compute several other possible speed points of waves II. The $\pi/2$ times of midpoint of wave I is a speed point forecasted X, as $(1/2)\mathrm{I} \times (\pi/2) = (1/2\pi) \times (\pi/2)\mathrm{II} = (1/4)\mathrm{II}$; $\pi/2$ times of X point; is the next speed point forecasted Y; as $(1/4)\mathrm{II} \times (\pi/2) = (\pi/8)\mathrm{II} = 0.393\mathrm{II}$. The $\pi/2$ times of Y point is the next speed point forecasted Z, as $(\pi/8)\mathrm{II} \times (\pi/2) = (\pi^2/16)\mathrm{II} = 0.617\mathrm{II}$, use $\partial(\pi)$ to express the division ratio of the group speed waves. Similarly, use $\pi/2$ as a constant rate to partition wave I, also calculate speed point forecasted of the wave I. According to the speed partition rate calculation method, use $\pi/2$ ratio partition speed point forecasted waves split below the wave I/2; another group speed points forecasted can be obtained before the speed wave trend change. Merge speed points forecasted partitioned by $\pi/2$ and $\pi/2$, and obtain speed points collection coefficient $\partial(5)$ partitioned by pi and the speed points forecasted as shown in Figure 3.

$$\partial(\pi) = \begin{cases} 1 \\ \dfrac{3}{4} \\ \dfrac{1}{2\pi}\left(\dfrac{\pi}{2}\right)^3 \\ \dfrac{1}{2} \\ \dfrac{1}{2\pi}\left(\dfrac{\pi}{2}\right)^2 \\ \dfrac{1}{2\pi}\left(\dfrac{\pi}{2}\right) \\ \dfrac{1}{2\pi} \\ \dfrac{1}{2\pi}\left(\dfrac{2}{\pi}\right) \\ \dfrac{1}{2\pi}\left(\dfrac{2}{\pi}\right)^2 \\ \cdots \end{cases} = \begin{cases} 1 \\ 0.75 \\ 0.5 \\ 0.617 \\ 0.5 \\ 0.393 \\ 0.25 \\ \cdots \end{cases}. \tag{1}$$

The heuristic formula of speed points forecasted of wave II can be summed up:

$$s(t) = \partial(\pi)\,\mathrm{II} = \frac{2}{\pi}\pi\mathrm{I}. \tag{2}$$

So we can obtain a set of speed points forecasted: $A = 5$ m/s, $Y = 7.86$ m/s, and $Z = 10$ m/s.

3.6. Speed Fluctuations Changing Trends Time. The time balance line of the vertical direction of the circle is divided into aliquots n that is changing trends time point for wave fluctuations; when wave goes through time balanced line or into the time zone near the balanced line, the fluctuations will changing trends or increase. Set waves I and II running time as 20 seconds, wave III running time is 25 seconds. The change trends angle of aliquots 5 circle is $\partial(5) = (72^\circ, 144^\circ, 216^\circ, 288^\circ, 360^\circ)$. Let 288° is the closest from

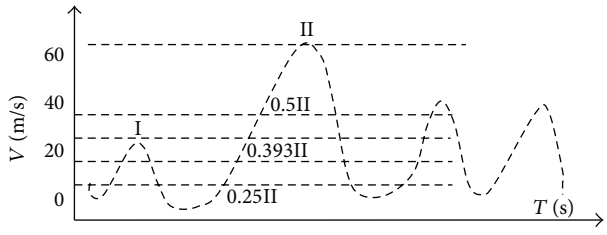


Figure 3: Speed points forecasted.

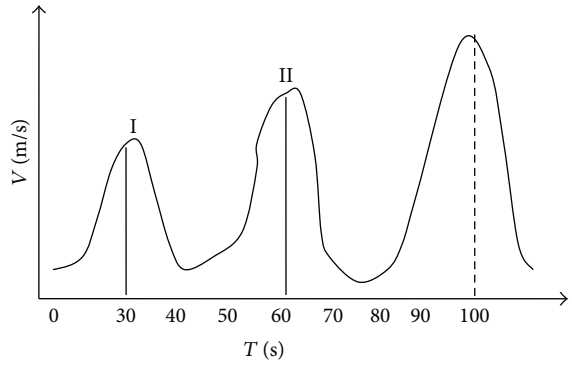


Figure 4: Speed fluctuations changing trends time.

waves III running time denote changing trends time point of waves III, is 4/5 times of circle changing trends time; the changing trends time of wave II arriving in the position of wave III is about 16 seconds, so heuristic formula of changing trends time is

$$\mathrm{III}_T = \left(\mathrm{I}_T + \mathrm{II}_T\right) \partial\,(n)\,. \tag{3}$$

For a wave run 30 seconds for the situation, select the speed of the data transmission node A according to the running speed of the vehicle to input speed values and computed. That means input of the values of the points of speed wave I into speed fluctuations forecasted algorithm after 30 seconds through heuristic formula computing speed point forecasted and speed fluctuations changing trends time after wave I emerges. According to formula (3), wave I running lasts 32 seconds, wave II running lasts 28 seconds, then the changing trends time point when the speed wave reach to waves III is 7/10 times of 60 seconds cycle, is 42 seconds, which means that, after about 42 seconds, the vehicle speed will rise, so $(\mathrm{I}_T+\mathrm{II}_T)\partial(7) = 60\times7/10 = 42$. In summary, the time of the vehicle stable speed is 42 seconds; accelerating wave III appears from 60 seconds to 102 seconds of II wave; vehicles will reach top speed at the time of 102 seconds. Speed fluctuations changing trends time as shown in Figure 4.

4. Simulation Experiments and Evaluation

NS-2 is a discrete event-driven, object-oriented network emulator. Using NS-2 to simulate network transmission through the establishment of a statistical model, for the design and evaluation of network protocols to provide a good experiment and test platform, compare SWF-GPSR with GPSR and other routing protocols. For comparison performance of the forwarding strategies, select three key indicators for evaluating routing protocols for data collection and analysis of results.

Table 1: MAC layer protocol parameters.

MAC layer protocol	IEEE802.11DCF
The number of packet	8192 bits
MAC layer header	224 bits
PHY layer header	192 bits
RTS	160 bits + physical layer header
CTS	112 bits + physical layer header
ACK	112 bits + physical layer header

(1) Average transmission delay: average time of the transmitter sends a complete data packet to the receiving node.

(2) Packet delivery success rate: ratio of the total number of packets successfully received to the total number of packets sent.

(3) Route length: the number of nodes through which data packet from the source node to the destination node successfully posted, that is, the count of hop.

(4) Link stability: the number of routing link changes in simulation time.

In order to improve the shortage of NS-2 for VANET for network simulation, using vehicular ad hoc networks mobility simulator (VanetMobiSim) software for traffic simulation, establish a system simulation model for simulating nodes move, in order to gain traffic reports and other information on traffic engineering analysis. VanetMobiSim arising from CanuMobiSim (communication in ad hoc networks for ubiquitous computing) is a framework for user mobility model which was developed by CANU research group of Stuttgart University based on JAVA language. It not only contains a certain amount of movement model, but also comprises the analyzer of the geographic data of different formats and visualization module, based primarily on the concept of the module that can be inserted, easy to be expanded, and able to support different types of mobile network simulation or simulation tools, such as NS-2, GloMoSim, and QualNet.

The simulation model map is Manhattan model. In Manhattan model, square grid denotes housing and mesh grid denotes street. The simulation experiments parameters are set as shown in Tables 1, 2, and 3.

In the simulation experiments, we compare SWF-GPSR (GPSR protocol based on SWF algorithm) protocol with general GPSR, 2-hop C-GEDIR, GRA and AODV protocols. Among them, 2-hop C-GEDIR (geographic distance routing) is the GPSR protocol variant, as discussed in [20]. Selecting some of the neighbors to forward the packet, neighbors that received the data packet continue to run the greedy algorithm. Node knows the location information of all it's hop and two-hop neighbor nodes. When a data packet is received,

Table 2: Simulation parameters.

Simulation duration	800 s
Interval of HELLO packet	2 s
Each node can produce data	30 sets
The number of nodes	100
Send packets cycle of CBR (constant bit rate) packets	1 s
Scene area	3500 m × 2500 m (7 × 5 grid, street length is 500 m)

Table 3: Vehicles parameters.

The maximum communication range of car node	300 m
The maximum speed of the vehicle	20 m/s
The minimum speed of the vehicle	5 m/s
Average moving speed	10 m/s
Average residence time when vehicles are at the intersection	2 s
Average maximal distance between vehicles	30 m

the node selected the nearest node X from the destination point from it's hop and two-hop neighbor nodes. If X is a hop neighbor, then the packet is forwarded directly to the X; if X is two-hop neighbor, it will have to find out neighbor node y to be able to reach X; data packets are submitted to X through y. We can see from Figure 5 that, in urban environment, due to the changing of the topology, the packet forwarding hops increase. Compared with general GPSR and 2-hop C-GEDIR, the number of hops of SWF-GPSR protocol is fewer in the packet forwarding process.

The GRA (geographical routing algorithm) is another variant of GPSR protocol, which is different from GPSR; when local optimization problems occur, GRA will start the route discovery mechanisms (flooding mechanism) to search route to the destination node from the best host. If a route reply message is received, save the route in the routing table to reduce the number of routing lookup process that may occur in the future. As can be seen from Figure 6, in the urban road environment, with the increase of the number of nodes, the gap of average delay end-to-end between general GPSR and SWF-GPSR is more obvious; SWF-GPSR is faster than the general GPSR and GRA in transmitting the data packet to the destination node.

The curve of Figure 7 shows the curve of packet arrival rate when average speed of the vehicles is between 5 m/s and 19 m/s. In the simulation experiment, randomly select 20 pairs of nodes for communication. As can be seen from Figure 7, when average speed of nodes increases, the performance of AODV, general GPSR, and SWF-GPSR shows a downward trend. But the packet arrival rate of SWF-GPSR protocol is always higher than the others. AODV (ad hoc on-demand distance vector routing) is a hop-by-hop routing that needs to maintain routing tables, though including information of the destination node in the routing table, but due to the changes of the vehicles movement location makes

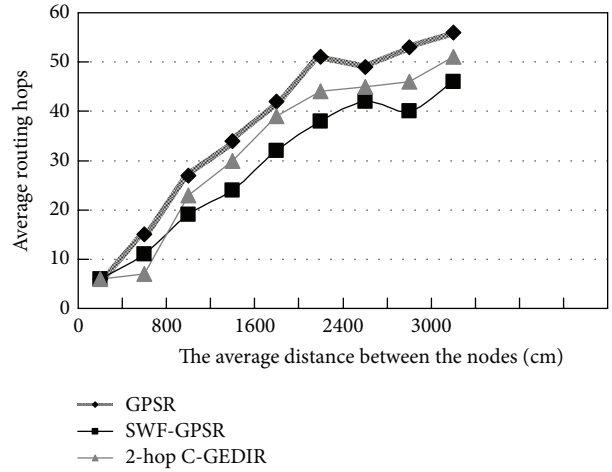


Figure 5: Comparison of average routing hops contrast.

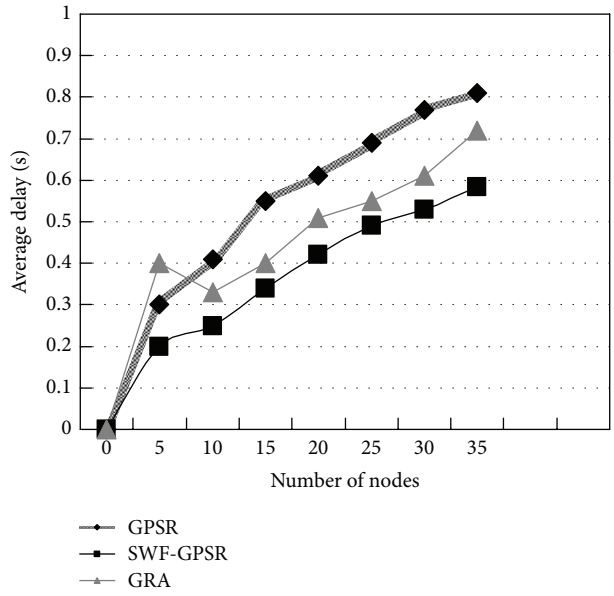


Figure 6: Comparison of average delay.

topology change frequently, resulting in a large number of invalid routing information.

As shown in Figure 8, through speed fluctuations forecasted algorithm and movement domain computation, the stability of the route link of SWF-GPSR is higher and the stability of within 10 nodes is similar to the general GPSR. However, when the number of routing nodes increases, the rate of change of GPSR link increased rapidly; by contrast, SWF-GPSR link is always keeps only about 30% of the rate of change.

The advantage of SWF algorithm is that it can forecast links of which the stability is strong through the speed fluctuations of vehicles in the real time, to meet the needs of VANET routing protocols. The GPSR protocol greedy algorithm in routing void area will use the boundary forwarding strategy, which can to some extent make up for unanticipated random

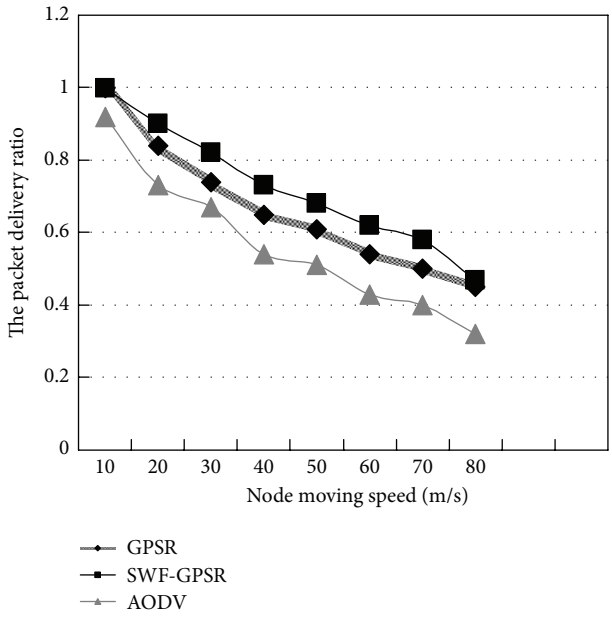


Figure 7: Comparison of packet delivery ratio.

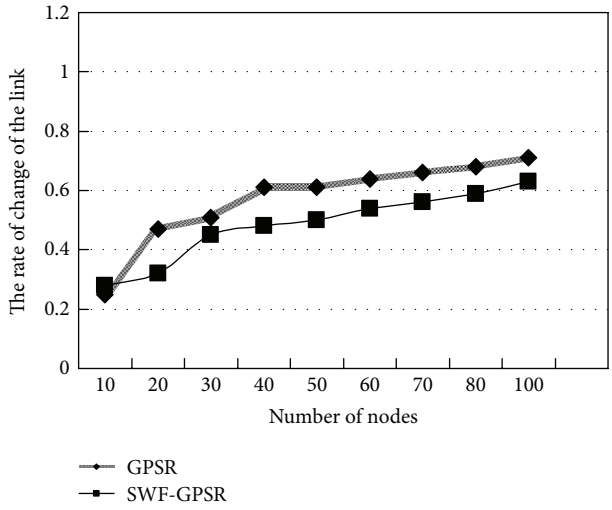


Figure 8: Comparison of change of link.

speed changes, changes in the direction of travel of the vehicle, road congestion, and other problems.

5. Conclusion

In this paper, we propose an SWF algorithm to improve the GPSR greedy algorithm. Unlike other research in greedy algorithm of GPSR protocol, this research begins with changes in the location and distance from the relative speed of vehicles, according to features of vehicles running on urban road, makes heuristic forecast for vehicle speed based on wave fluctuations in equilibrium theory, achieves collection of the nodes of stable relative velocity, then designs the heuristic SWF algorithm to predict continuity and change timing of vehicles speed, and then computes position that vehicles may occur in, finding out the shortest route before boundary forwarding model is activated. Through theory analysis and simulation experiments, we evaluated SWF-GPSR and other protocols in the average transmission delay, routing hops, packet delivery ratio, and link changes. The results show that the SWF-GPSR protocol has better robustness and higher performance.

However, the structure of city road is so complex; set the same route before driving, that is, a certain degree of idealization. SWF-GPSR protocol needs to be thoroughly tested in the scene of more complex and road structure interconnected, such as specific changes in the density of vehicles and road traffic. Furthermore, the error of speed forecasted will result in the error of movement range of the computation, so vehicle may not appear within a predictable range. The biggest problem is that the height ratio between the wave curve is based on the judgment on the speed of change coming from the fluctuations in equilibrium theory. SWF algorithm will have a better accuracy in the speed limit on the road; in opposite, the accuracy will decline, producing larger computing energy. The algorithm of circle changing trends time and angle of the wave, the wave split rate also need more scientific and rational argument and proof of fuller experiments.

The next step of research is to improve the speed forecasted curve segmentation rate and limit speed computation methods, to introduce machine learning algorithm for intelligent judgment in speed change for road features and design more accurate and more efficient VANET routing algorithm for specialized city road.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants No. 61070169, Natural Science Foundation of Jiangsu Province under Grant no. BK2011376, Specialized Research Foundation for the Doctoral Program of Higher Education of China under Grant no. 20103201110018 and Application Foundation Research of Suzhou of China no. SYG201118.

References

[1] S. Panichpapiboon and W. Pattara-atikom, "A review of information dissemination protocols for vehicular Ad Hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 3, pp. 784–798, 2012.

[2] C. Christian and J. Morettil Tian, "Communication architecture of CarTalk2000," in *Proceedings of the l0th World Congress and Exhibition on Intelligent Transport Systems and Services*, pp. 150–176, Madrid, Spain, 2003.

[3] L. Chen, Z.-J. Li, S.-X. Jiang, and C. Feng, "MGF: mobile gateway based forwarding for infrastructure-to-vehicle data delivery in vehicular Ad Hoc Networks," *Chinese Journal of Computers*, vol. 35, no. 3, pp. 454–463, 2012.

[4] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: a survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.

[5] B. Yu, C.-Z. Xu, and M. Guo, "Adaptive forwarding delay control for VANET data aggregation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 1, pp. 11–18, 2012.

[6] R. Schmitz, A. Leiggener, and A. Festag, "Analysis of path characteristics and transport protocol design in vehicular ad hoc networks," in *Proceedings of the 63rd IEEE Vehicular Technology Conference (VTC '06-Spring)*, pp. 528–532, Melbourne, Australia, May 2006.

[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, Boston, Mass, USA, August 2000.

[8] Z. Hengyang, F. Weihong, W. Ling et al., "Greedy geographic routing protocols, a real-time and reliable mobile wireless sensor networks," *Computer Research and Development*, pp. 714–7715, 2009.

[9] I. Amundson, M. Kushwaha, and X. D. Koutsoukos, "A method for estimating angular separation in mobile wireless sensor networks," *Journal of Intelligent and Robotic Systems*, 2012.

[10] M. Rudack, M. Meincke, K. Jobmann, and M. Lott, "On traffic dynamical aspects of Inter Vehicle Communications (IVC)," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC '03-Fall)*, pp. 3368–3372, October 2003.

[11] J. J. Pan, S. J. Pan, J. Yin, L. M. Ni, and Q. Yang, "Tracking mobile users in wireless networks via semi-supervised colocalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 587–600, 2012.

[12] C.-Y. Chang, C.-Y. Lin, and C.-T. Chang, "Tone-based localization for distinguishing relative locations in wireless sensor networks," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1058–1070, 2012.

[13] J. Chen, M. B. Salim, and M. Matsumoto, "A single mobile target tracking in voronoi-based clustered wireless sensor network," *Journal of Information Processing Systems*, vol. 7, no. 1, 2011.

[14] D. Gu and H. Hu, "Spatial Gaussian process regression with mobile sensor networks," *IEEE Transactions on Neural Network and Learning System*, vol. 23, no. 8, 2012.

[15] C.-T. Chang, C.-Y. Chang, and C.-Y. Lin, "Anchor-guiding mechanism for beacon-assisted localization in wireless sensor networks," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1098–1111, 2012.

[16] K. Li, D. Guo, Y. Lin, and I. C. Paschalidis, "Position and movement detection of wireless sensor network devices relative to a Landmark graph," *IEEE Transactions on Mobile Computing*, vol. 11, no. 12, 2012.

[17] C.-Y. Chang, Y. Xiang, and M.-L. Shi, "Development and status of vehicular ad hoc networks," *Journal on Communications*, vol. 28, no. 11, pp. 116–126, 2007.

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

[19] F. Mourad, H. Snoussi, M. Kieffer, and C. Richard, "Robust interval-based localization algorithms for mobile sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 303895, 7 pages, 2012.

[20] S.-H. Cha, K.-W. Lee, and H.-S. Cho, "Grid-based predictive geographical routing for inter-vehicle communication in urban areas," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 819497, 7 pages, 2012.

[21] I. Stojmenovic and X. Lin, "Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1023–1032, 2001.

[22] L. Y. zhong, *Fluctuations Equilibrium Theory*, Southern Press, 2007.